Master 1 IPS 2018-2019
Darlène Bechtel | Encadrants
Hicham Tabet | Konstantin Todorov
Mohamed Salifou-Djitao | Andon Tchechmedjiev

ClaimsKG Statistical Observatory
Conception d'une interface présentant des statistiques associées à l'outil ClaimsKG, graphe de connaissance d'assertions issues du fact checking

Sommaire

1. INTRODUCTION

1.1 Web sémantique

Le terme "Web sémantique" est le terme attribué par le W3C (World Wide Web Consortium) pour désigner le web standardisé des données liées. Ses applications sont d'abord tournées vers les programmes, et non vers le langage naturel. Il est constitué d'un ensemble de technologie visant à indexer et organiser le contenu des ressources d'internet. L'objectif général est de lier de manière logique les données du web. Afin de le rendre plus pertinent, d'extraire le sens de l'information, de décloisonner cette dernière de sa source, son

emplacement ou son format. La notion de métadonnées utilisables par les machines apparaît dès 1994 par Tim Berners-Lee, considéré comme l'un des fondateurs du web. En 1999, la première version de RDF (Resource Description Framework) voit le jour, langage qui définit un cadre général sous forme de modèle de graphe, pour la standardisation des métadonnées des ressources Web. Des vocabulaires spécifiques se sont ensuite développés pour décrire les différentes relations des données. Enfin, les langages comme RDFS (Resource Description Framework Schema) et le langage d'ontologie OWL (Ontology Web Language) destinés à structurer ces vocabulaires, sont publiés en février 2004. En pratique le web sémantique vise à attribuer une URI à chaque ressource, les URI devant permettre aux utilisateurs d'accéder facilement à ces ressources, fournir des informations utiles en utilisant le modèle RDF, et les relier avec ce même modèle à d'autres ressources pertinentes. Il vise donc à optimiser le fonctionnement des machines en organisant et enrichissant d'après un standard, l'information dans le but final d'augmenter l'accessibilité et la pertinence de cette information à l'utilisateur.

1.2 Fake news et fact checking

Les mensonges publics et les fausses informations n'ont rien de nouveau. Ils feraient partie du discours politique depuis l'antiquité (Robert Zaretsky de l'Université de Houston). Le terme fake news serait apparu aux Etats-Unis à la fin du 19ième siècle (Robert Love dans la Columbia Journalism Review). Le domaine politique n'est pas le seul à être concerné. Les entreprises, les lobbys et la publicité peuvent également jouer de la mésinformation ou désinformation à des visées commerciales. Afin d'augmenter leur visibilité, via trafic et abonnements, et d'influencer les personnes, considérés comme autant de consommateurs. Le phénomène actuel des fake news a pris son essor médiatique lors des campagnes présidentielles, en 2016 aux Etats-Unis, puis en France en 2017. Ces temps forts politiques donnent une place prépondérante aux réseaux sociaux, devenus véritables médias politique. En France d'après le baromètre annuel de 2017 du journal la Croix, les 18-24 ans s'informent à hauteur de 41% par les réseaux sociaux, aux USA ce taux dépassait 60% lors de la campagne 2016. Cependant, ces événements ayant mis au jour l'ampleur et les dangers de la désinformation, ont permis une prise de conscience publique. Plaçant au coeur du débat et des préoccupations la nécessité d'une information vérifiée, autrement appelée fact checking. Le nombre de fact checkers a plus que triplé depuis 2014 (poynter). Présent sur tous les continents, ils sont recensés dans 67 pays (liste Poynter). Leur activité devenue cruciale, nécessite un véritable travail journalistique et de la transparence. Les facts checkers peuvent être de diverses nature, organisations indépendantes, groupe de travail au sein de la presse publique, tel que l'AFP factuel ou privé, comme les décodeurs du journal le Monde.

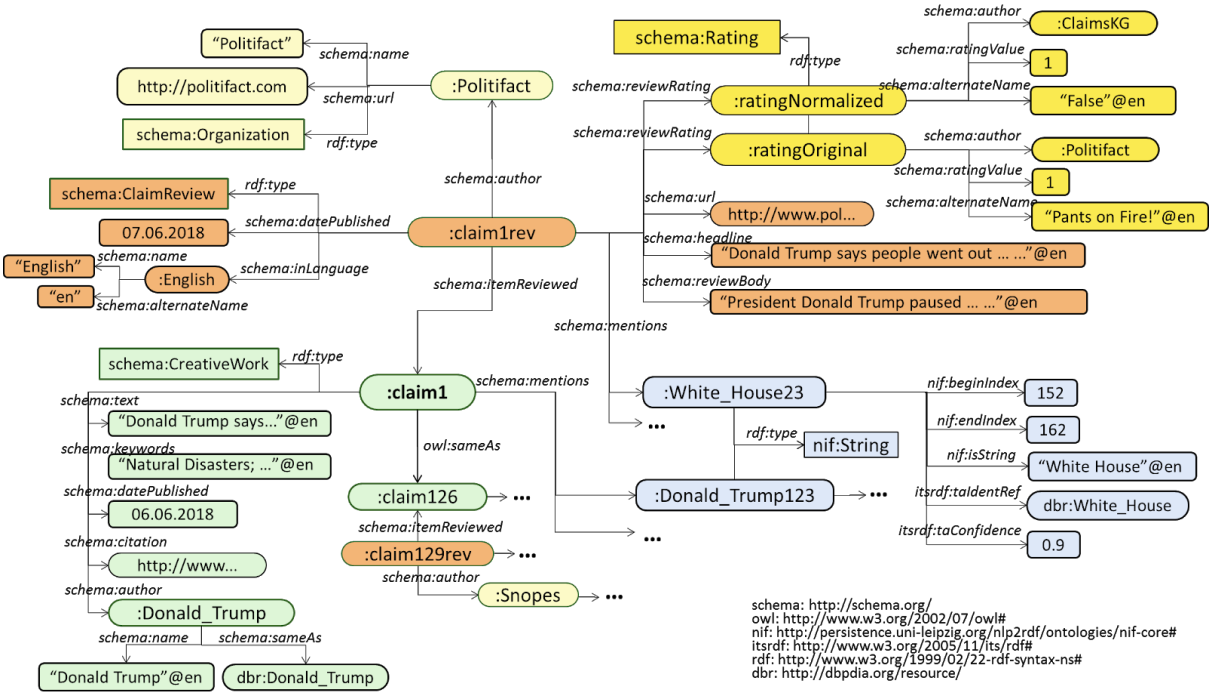
1.3 Détection automatique des fakes news et état de l'art

La fact-checking automatique est donc un sujet d'intérêt croissant pour le domaine de la recherche en Intelligence Artificielle. Il existe plusieurs approches pour extraire des assertions vérifiées et pour tenter de les évaluer automatiquement. Le modèle passé s'appuie sur des représentations structurées permettant des comparaisons des assertions vérifiées dans différentes KB (Knowledge Base, base de connaissance). L'approche la plus récente consiste à mettre des labels, c'est à dire des étiquettes de véracité (vrai, faux, mixte, etc.) sur les assertions pour entraîner et appliquer des modèles de machine learning. Deux

méthodes générales existent. L'Approche de référence, à partir de sites de fact checking: plusieurs projets utilisent cette approche car elle offre l'avantage de la qualité et quantité de la donnée. En effet, les assertions sont analysées et labellisées par des professionnels effectuant un vrai travail journalistique. Les données ont été récoltées manuellement ou de manière automatique depuis des sources comme Channel4, Politifact ou encore la liste Wikipédia des fake news. Cette façon de faire constitue un référentiel. La plus large collection relevant de cette approche est le jeu de données Emergent qui totalise 126 000 stories labellisées, avec cependant un manque de transparence sur le processus. L'autre méthode est l'annotation manuelle des données : elle permet de récolter un bon nombre de données, toutefois la qualité de labellisation est de fait inégale. Le plus large des jeux de données utilisant le crowdsourcing (annotations ouvertes faites par le grand public) prend sa source depuis Wikipédia et comprend près de 200 000 entrées. La structuration des données utilisant la première méthode prend souvent la forme de KG (Knowledge Graph, graphe de connaissance). Ces graphes références sont souvent produits à partir de DBpedia et Wikipedia. Les graphes plus récents représentent donc une assertion comme un triplet vérifiant des liens existant dans les références, afin de faire correspondre ou compléter l'existant. On peut citer Knowledge vault ou KnowMore.

1.4 Le projet ClaimsKG

ClaimsKG est un graphe dynamique d'assertions annotées entièrement basé sur des sites de fact-checking reconnus. Il est généré à partir d'une récolte semi-automatique régulière des assertions et métadonnées depuis les sites fact-checking. De cette façon, on peut concentrer l'information autour de la vérification des assertions émergentes, non disponible sur Wikipédia ou les KGs établit. Open source, dans sa catégorie il dispose du plus large spectre de métadonnées et comprend le plus grand nombre d'assertions, soit au moment de la rédaction 28 384. En opposition aux approches existantes, les assertions sont modélisées selon un modèle RDFS spécifique conçu pour cette application, augmentant la réutilisation et l'extensibilité. Le modèle s'appuie sur un vocabulaire établi (des ontologies comme celles de OWL et schema.org) et annotations des assertions avec l'ajout des entités de DBpedia. En terme d'application, la ressource présentée fournira un support de recherche dans le domaine du fact-checking, pourra être utilisée comme entraînement ou évaluation de modèles de machine learning. Et permet de plus à l'utilisateurs d'obtenir un échantillon avec une simple requête en langage SPARQL. Le modèle utilisé est présenté ci-dessous, schema:mentions désigne les entités.

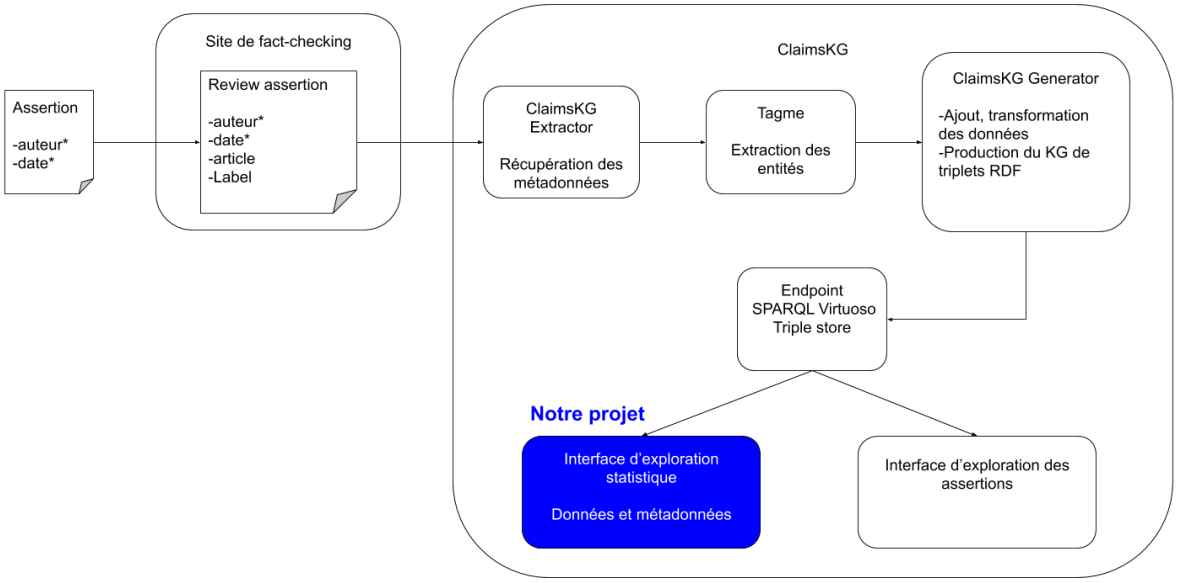


2. DÉVELOPPEMENT

2.1 Recueil des besoins

Ce projet est un projet international regroupant plusieurs laboratoires et entités qui sont les suivantes : Le LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier), le LGI2P (Laboratoire de Génie Informatique et d'Ingénierie de Production) de l'École des Mines d'Alès, le GESIS (Institut des Sciences Sociales de Leibniz), le Centre de Recherche L3S . Ainsi que l'Université de Montpellier.

La problématique du travail présenté ici est de concevoir une interface graphique permettant l'exploration statistique des assertions regroupées par ClaimsKG, mais aussi des données et métadonnées du KG. Ci-dessous un schéma résumé de l'outil ClaimsKG et de la position de notre travail au sein du projet global, les astérisques correspondent à des informations parfois indisponibles, label signifie ici l'indice de véracité des assertions.



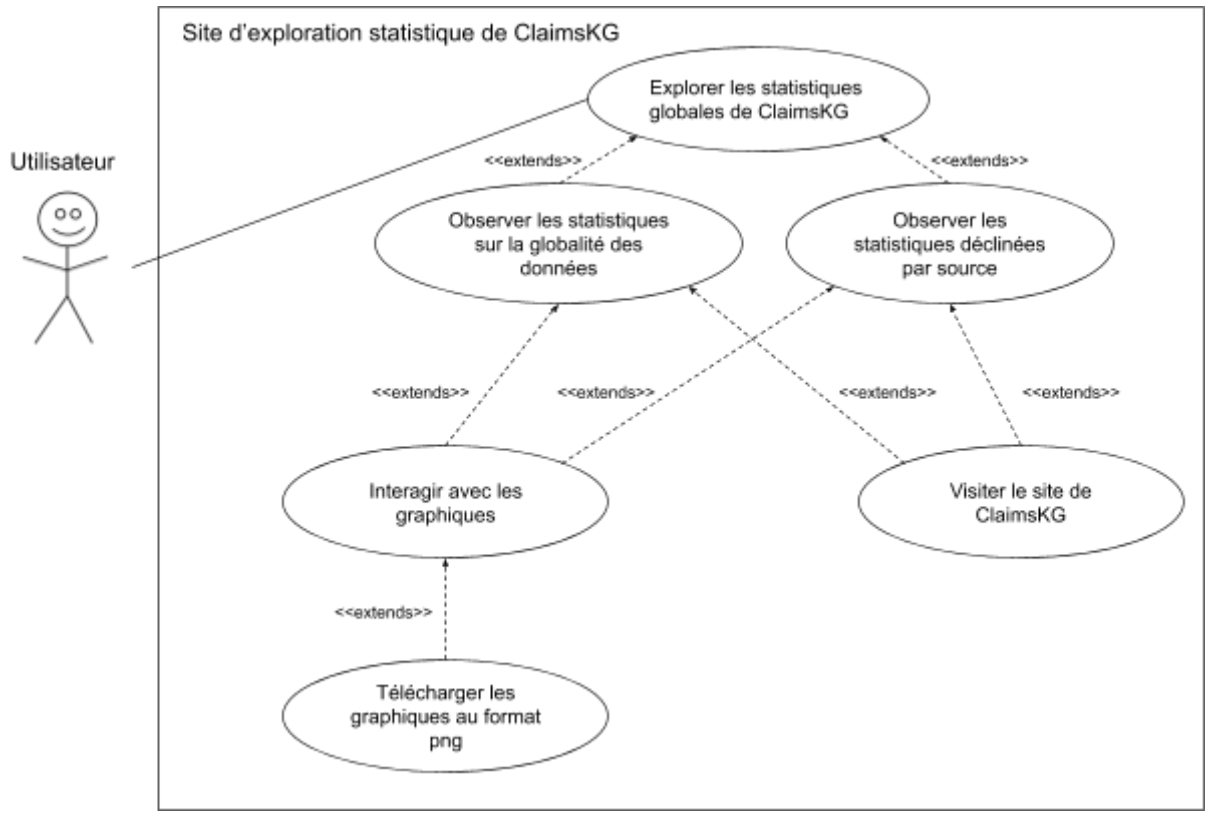
Le livrable doit répondre à plusieurs caractéristiques. L'interface, les données qu'elle expose, ainsi que la manière de les exposer, devront être accessibles à des utilisateurs non spécialisés dans l'informatique. Tels que des journalistes, des professionnels des sciences sociales ou le grand public qui s'intéresse au sujet de fact-checking. L'interface doit présenter graphiquement les informations synthétiques sur le jeux de données de ClaimsKG, afin que les utilisateurs puissent comprendre l'outil et interpréter les autres fonctionnalités. Des fonctionnalités exploitant les données sur les assertions, (autre que la synthèse de métadonnées) visant l'intérêt pour les utilisateurs définitifs devront être proposés. Le livrable devra être déposé sur GitHub en vue de son déploiement.
En terme de conception, l'interface doit pouvoir présenter les données actualisées de manière automatique ou semi-automatique. Le backend doit être réalisé en Python 3 pour des raisons de compatibilités avec ClaimsKG. Les librairies de frontend utilisées pour réaliser les représentations graphiques devront être dynamiques et permettre une exploration interactive. Les parties prenantes auront l'occasion de discuter le livrable et de le valider le cas échéant.

2.2 Spécification des besoins

2.2.1 Fonctionnalités et cas d'utilisations

Plusieurs cas d'utilisations ont été identifiés auxquels nous tenterons de répondre tout au long de la réalisation du projet.

Réaliser la page d'accueil :



Description:
L'utilisateur souhaite explorer les statistiques de ClaimsKG et avoir un résumé visuel chiffré des données et métadonnées du graphe. Il souhaite éventuellement découvrir des informations supplémentaire à celle présentée sur la page de ClaimsKG.

Acteurs: Dans ce cas d'utilisation les acteurs peuvent être multiples. Une partie prenante du projet, qui souhaite vérifier visuellement les données après une actualisation du graphe par exemple. Ou qui souhaite simplement observer des métriques synthétiques. Mais aussi un utilisateur non spécialisé, s'intéressant au fact-checking, un futur collaborateur, un étudiant ou un journaliste par exemple.

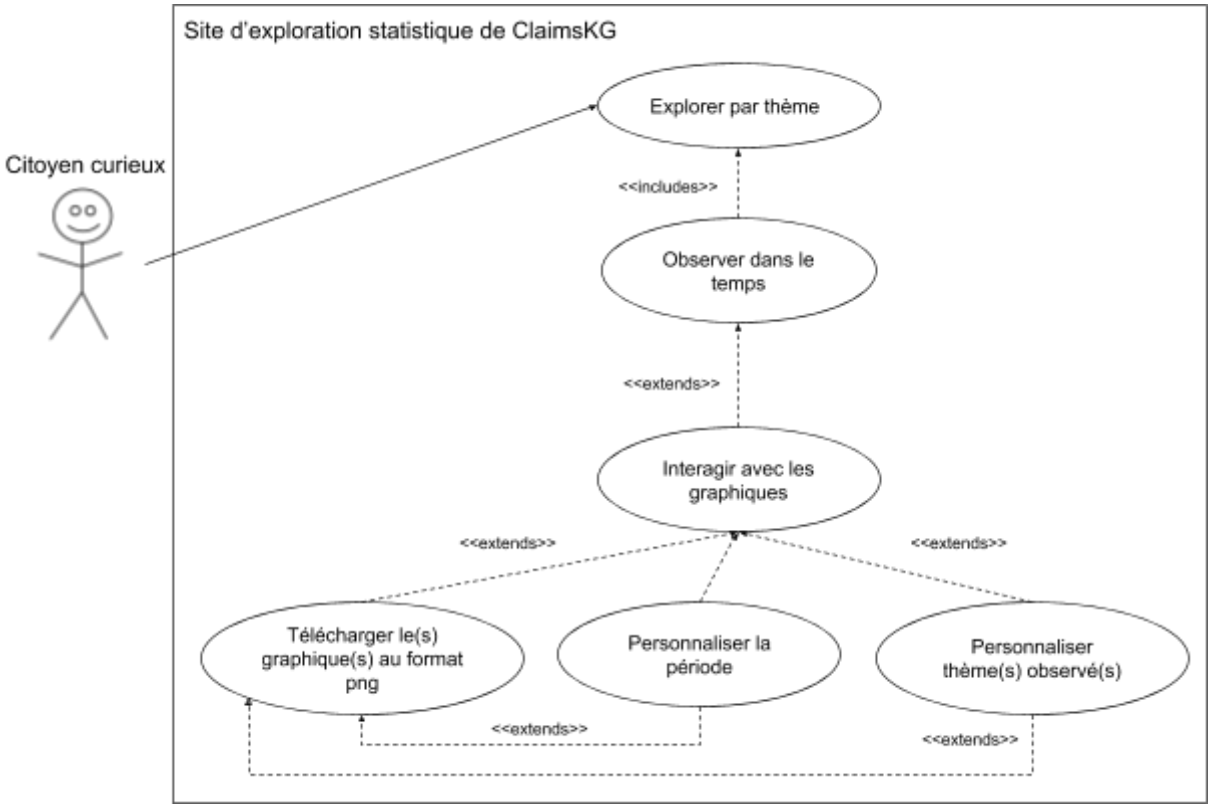
Contexte:
Données en entrée et pré-conditions: les métriques soit les nombre, moyennes, pourcentages etc. (des données et métadonnées) choisies pour résumer et exposer les données du graphe de ClaimsKG. Pour cela, il faut extraire les données du graphes et calculer les métriques.
Données en sortie et post-conditions: les métriques exposées sous la forme de plusieurs graphiques accessibles.

Scénario principal :
1. L'utilisateur arrive sur l'interface.
2. Le système renvoie les graphiques associés.
3. L'utilisateur observe les statistiques globales.

4. L’utilisateur interagit avec les graphiques en explorant les légendes au survol.
5. L’utilisateur observe les statistiques déclinées par sources.
6. Le citoyen interagit avec les graphiques en explorant les légendes au survol.

Variantes:
cas 1 : Le citoyen est satisfait de la simple observation et quitte le système à l’étape 3.
Cas 2 : étape 4,5 ou 6 - Le citoyen peut télécharger le ou les graphiques qu’il désire au format png.

Explorer thématiquement:



Description:
Le citoyen souhaite explorer les données du système autour d’un thème particulier. Il souhaite contextualiser la production et la véracité des informations autour de ce thème. Dans le temps, pour observer les évolutions par exemple en rapport à des moments d’actualités.

Acteur: Citoyen curieux

Contexte:
Données en entrée et pré-conditions: les mots-clés présents dans le graphe de ClaimsKG associés aux claims, les claims doivent être agrégées par thème.
Pour cela, il faut définir des thèmes en se basant sur les mots-clés et choisir des thèmes cohérents avec les données du graphe ClaimsKG, pour regrouper des sous catégorie de sujets et de mots-clés.

Données en sortie et post-conditions: le nombre de claims par thèmes, sans valeur aberrante par exemple dans les dates, regrouper avec différentes granularités dans le temps, exposées sous la forme de plusieurs graphiques représentant l'évolution du nombre de claims associées aux thèmes dans le temps.

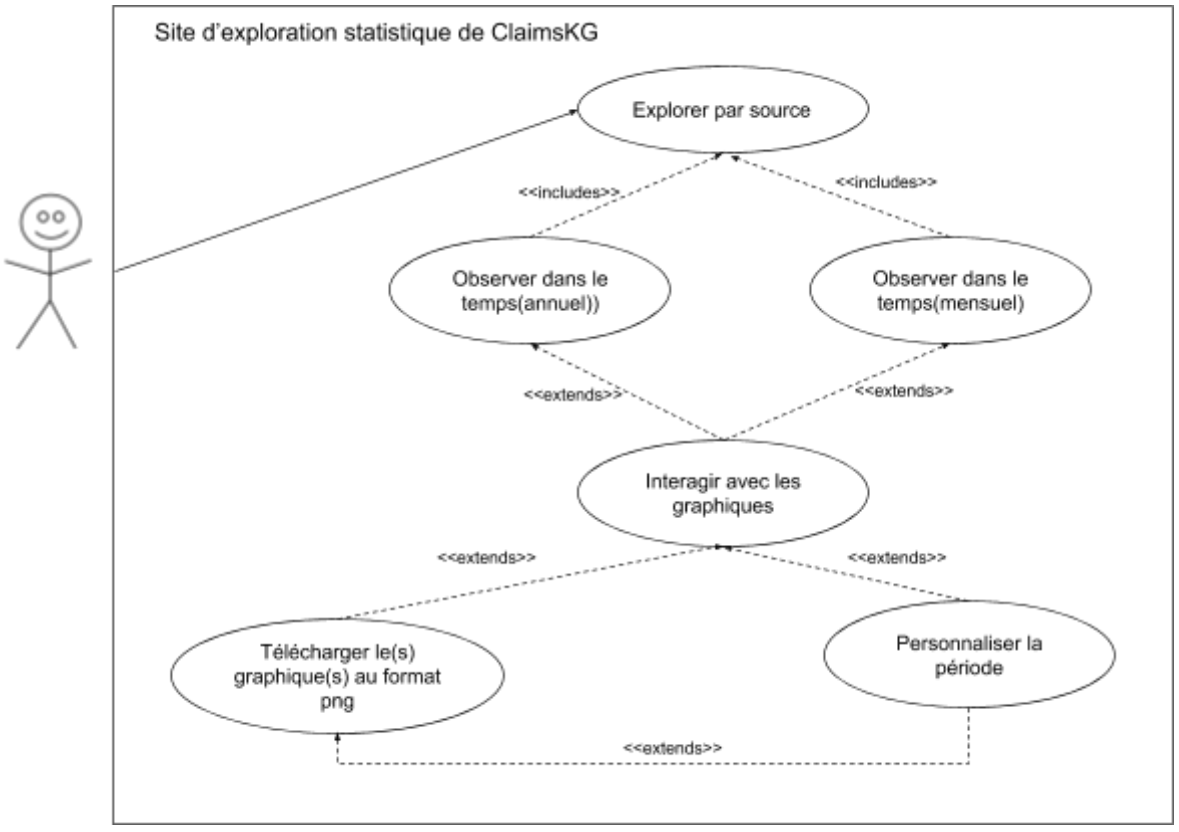
Scénario principal :

1. Le citoyen arrive sur l'interface et choisit la recherche thématique
2. Le système renvoie les graphiques associés.
3. Le citoyen observe l'évolution du nombre de claims associées aux différents thèmes.
4. Le citoyen interagit avec les graphiques en explorant les légendes au survol.
5. Le citoyen focalise le graphique sur l'intervalle de temps souhaité.
6. Le citoyen peut isoler les courbes correspondant aux thèmes qui l'intéressent plus particulièrement.

Variantes:

cas 1 : Le citoyen est satisfait de la simple observation et quitte le système à l'étape 3.
Cas 2 : étape 4,5 ou 6 - Le citoyen peut télécharger le ou les graphiques qu'il désire au format png.

Explorer par source:



Description: L'internaute souhaite explorer les données du système autour d'une source (site de fact checking) en particulier. Il souhaite connaître pour Snopes ou Politifact (plus grands contributeurs des données), l'évolution temporelles du nombre d'assertions par label d'une source.

Scénario principal :

1. Le citoyen arrive sur l'interface et choisit la recherche thématique
2. Le système renvoie les graphiques associés.
3. Le citoyen observe pour chaque source, l'évolution du nombre de claims associées à chaque label.
4. Le citoyen interagit avec les graphiques en explorant les légendes au survol.
5. Le citoyen focalise le graphique sur l'intervalle de temps souhaité.
6. Le citoyen peut isoler les courbes correspondant aux labels qui l'intéressent.

Variantes:

cas 1 : Le citoyen est satisfait de la simple observation et quitte le système à l'étape 3.
Cas 2 : étape 4,5 ou 6 - Le citoyen peut télécharger le ou les graphiques qu'il désire au format png.

2.3 Prototype

2.3.1 Architecture

2.3.1.1 Modèle

2.3.1.1.1 Accès, récupération et mise en forme des données

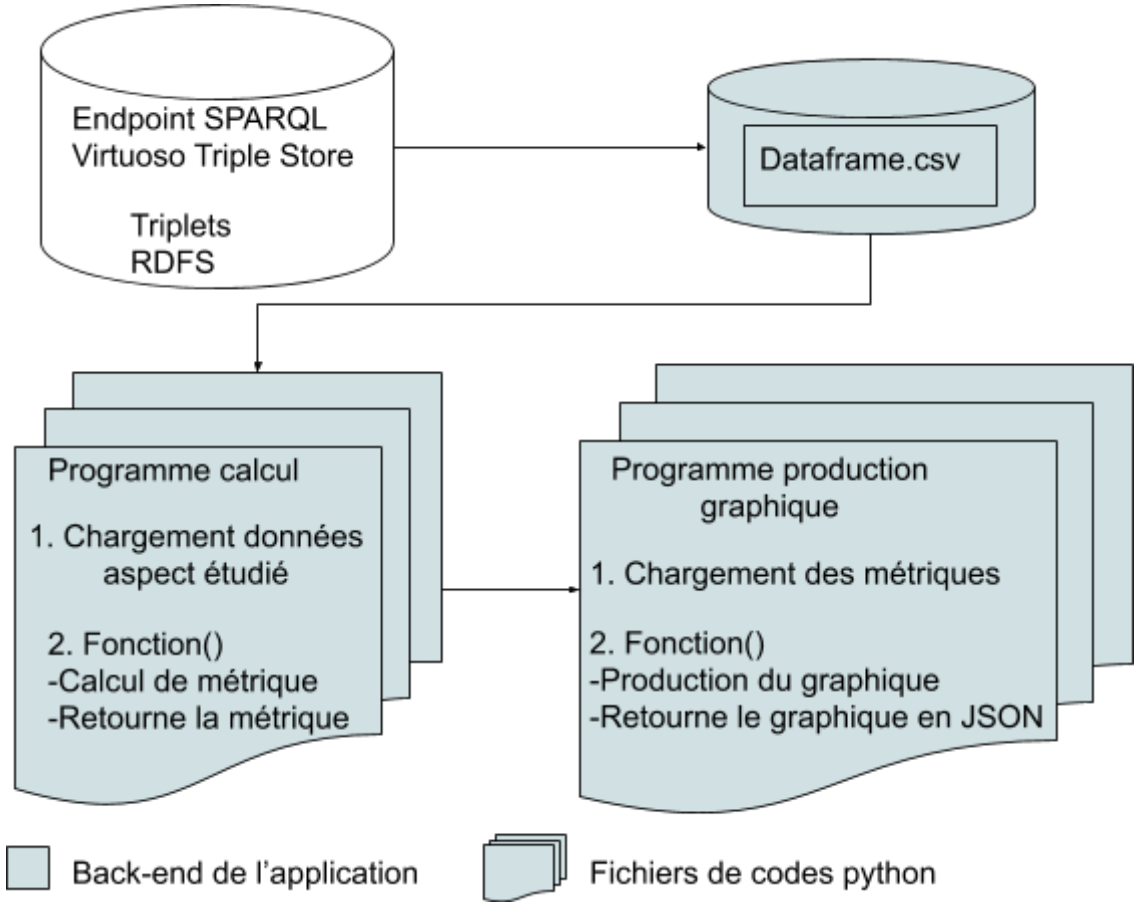
Pour la récupération des données, deux méthodes correspondant à des stratégies et des contraintes imposées différentes, ont été employées.
Premièrement, un accès totalement local, en exploitant un fichier dump du graphe ClaimsKG comportant tous les triplets RDFs. Les programmes utilisant cette méthode ont été découpés selon l'aspect du graphe étudié, par exemple, les entités, les auteurs, les mots-clés, etc. La logique de développement était la suivante:
- Le graphe en .ttl (format du langage RDF) est parsé pour pouvoir exploiter les données.
- Les requêtes SPARQL concernant l'aspect étudié sont effectuées.
- Les résultats sont stockées dans des structures de données, allant de variables, à des dictionnaires, en passant par des csv ou des json.
- Les structures de données sont manipulées pour calculer la ou les métriques choisies.
- Les métriques sont retournées pour être utilisées ensuite pour la création des graphiques.

Ce découpage par aspect étudié a été entraîné par des contraintes de développement. En effet l'étape de parsing (désérialisation) du graphe comprenant plus de trois millions de triplets étant longue. Le fait de faire peu de requêtes à la fois permettait une meilleure vérification du code. Cela a conduit à la nécessité d'avoir un cache local pour accélérer les requêtes et le développement, en limitant le recalcul à chaque exécution d'un programme.
Les principaux désavantages de cette méthode sont donc la durée pour parser le graphe et d'utiliser des données statiques, i.e. de ne pas avoir accès aux dernières mises à jour des données. Néanmoins, les avantages d'un accès et d'une utilisation locale des données peuvent être l'indépendance du fonctionnement du serveur. Cela peut également éviter des problèmes de compatibilité et maintenance d'accès aux données, en cas de changement de leur structuration au niveau du serveur. Une fois les données générées, cela peut aussi augmenter la vitesse d'exécution.
Deuxièmement, un accès au triplets depuis un endpoint SPARQL virtuoso triple store, qui permet de faire directement des requêtes en langage SPARQL et récupérer les résultats

sous différents formats. Pour cette méthode la récupération des données depuis le endpoint peut s'apparenter à la partie extraction du processus l'ETL (Extraction, Transformation, Chargement). Ensuite, les données sont structurées dans une dataframe (un tableau permettant d'avoir des données en colonnes avec différents type, comme des chaînes de caractères, des nombres ou des dates) et stockées dans le dossier de back-end sous la forme d'un csv. Cette manière de faire pourrait correspondre à une philosophie de Datamart, qui est un sous-ensemble du data warehouse, avec la présentation des données projetée vers le domaine métier. Plusieurs avantages sont engendrés par cette méthode. La dataframe qui opérera comme la base de données pourra être générer qu'une seule fois, puis ensuite simplement lue par tous les autres programmes qui l'utilisent. Ceci pourra entraîner un gain de performance, mais aussi un temps de réponse plus rapide. De plus, l'intérêt majeur de l'accès au endpoint et de pouvoir traiter des données synchronisées, provenant du graphe actualisé (au moment de la génération de la dataframe). C'est cette méthode qui a été retenue pour le livrable final, on y fera référence par la suite, sauf mention contraire, le projet a donc été totalement refactoré.

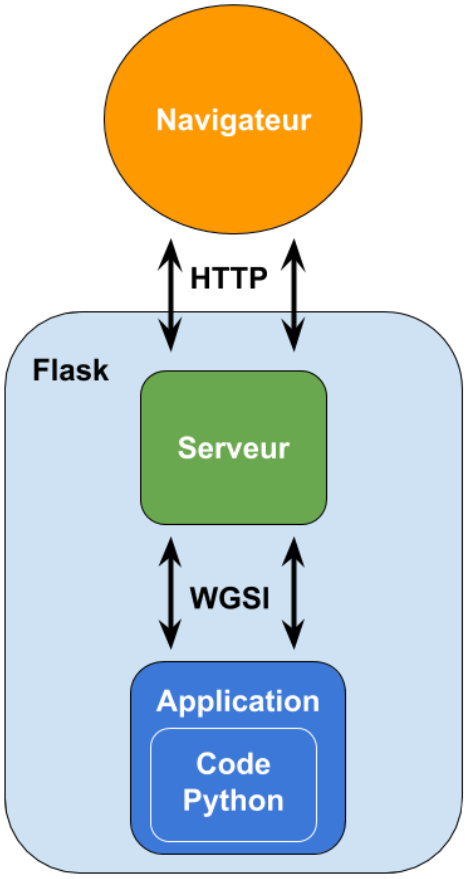
2.3.1.1.2 Structuration du back-end

Dans un objectif de modularité du code les programmes calculant les statistiques des différents aspects étudiés ont été dissociés de ceux dédiés à leurs représentations, c'est à dire la production de graphiques. La structure du back-end est résumée sur le schéma ci-dessous:



2.3.1.1.3 Serveur et application

Le projet étant en codé en Python 3, afin de réaliser notre application web deux outils vont être utilisés : un serveur et pour structurer le développement, un framework.
Un framework ou socle d'applications, est un cadre de travail qui offre un squelette applicatif, simplifiant le développement, et formalisant une architecture répondant à un besoin.
Dans notre projet nous utilisons le microframework Flask. Un microframework est comme son nom l'indique un framework réduit, plus léger, plus simple à utiliser, particulièrement adapté, pour débuter ou créer de petites applications. Par ailleurs, il peut être scalable en ajoutant des composants en fonction de l'évolution des besoins. Flask permet de créer une application en Python, utilise le protocole WGSI (Web Server Gateway Interface), comprend un serveur de développement et fonctionne à travers des requêtes HTTP.



Flask permet dans notre projet de faire un SSR partiel (Server Side Rendering). Une partie du code HTML est composé et rendu, notamment par l'intermédiaire du service de templating Jinja2 qui est un standard pour Python et particulièrement avec Flask. Cela permet d'avoir une API REST, de manipuler des objets python et donc du JSON.

2.3.1.2 Vue

La partie vue de notre projet constitue l'affichage coté client du HTML et CSS.

2.3.1.2 Contrôleurs

Le côté contrôleur est constitué par le Javascript sous plusieurs formes, la partie qui n'est pas rendu par Flask est la partie de production des graphiques avec appel aux librairies externes plotly.js qui utilise aussi d3.js. Les graphiques ainsi produits à partir des JSON vont ensuite pouvoir être dynamiques et permettre différentes interactions au survols et aux clics de l'utilisateur. Il y à également des interactions de navigations de pages sous la forme de JQuery.

2.3.2 Choix des technologies

Le choix des technologies utilisées pour réaliser l'application ont été dictés par plusieurs arguments que nous allons présentés ici.
Premièrement, la compatibilité. Le langage de programmation utilisé est Python 3 car c'est le langage dans lequel est codé l'outil de génération du graphe de ClaimsKG. En effet, dans un désir de scalabilité, si l'outil de génération et l'application d'exploration statistiques devaient être amenés à s'interfacer, l'utilisation d'un langage commun faciliterait l'opération. De plus, cela assure les utilisations et modifications ultérieures du code par les différents membres du projet ClaimsKG. Par ailleurs, Python est un langage largement utilisé, en 2018 il se classe en quatrième position du l'indice TIOBE qui recense les langages les plus populaires et premier selon l'IEEE (Institute of Electrical and Electronics Engineers) qui classe les meilleurs langages de programmation (sources).
Ensuite, un autre aspect important à prendre en compte est d'utiliser des technologies Open Source. cela entraîne un lot d'avantages comme la gratuité, la maintenabilité, et la communauté autour de la technologie. Ce projet étant un projet universitaire réalisé par des débutants en informatique, un argument crucial est l'accessibilité. L'importance de la communauté précédemment évoquée est capitale en terme de documentation, d'exemples et de reproductibilité. Une autre chose à prendre en compte et qui découle des aspects précédents est la richesse de l'environnement (qui comprend également la communauté) et ce qui nous intéresse ici les librairies et frameworks autour du langage qui vont permettre de réaliser les fonctionnalités désirées. Dans notre cas plusieurs librairies de Python et de Javascript ont été utilisées, résumé dans le tableau ci-dessous.

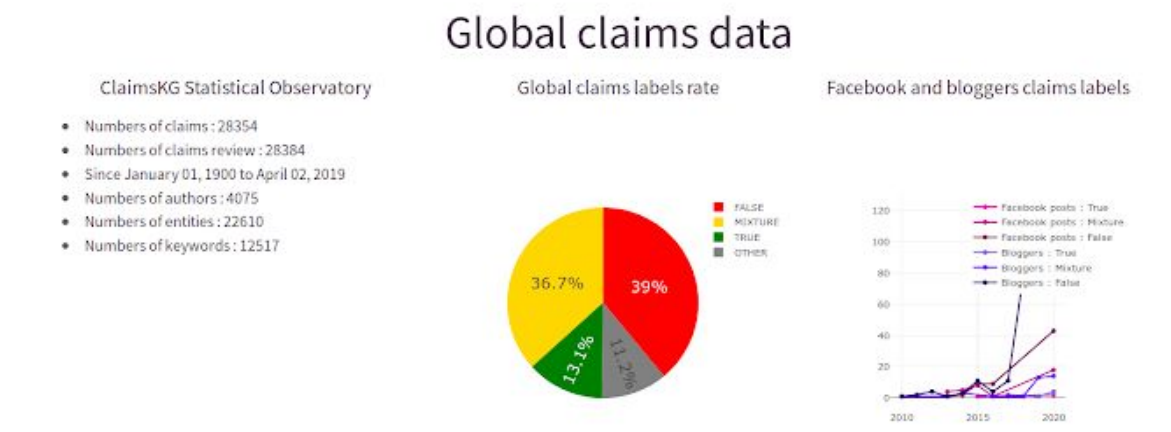
Backend	Graphe RDFs et SPARQL	rdflib, SPARQLWrapper (surcouche de rdflib)
	Dataframe et calculs	pandas, numpy, et statistics
	Graphiques	plotly
	Application, serveur	Flask et son module render-template
Frontend	Graphiques	Plotly.js et sa dépendance d3.js
	Navigation	jQuery

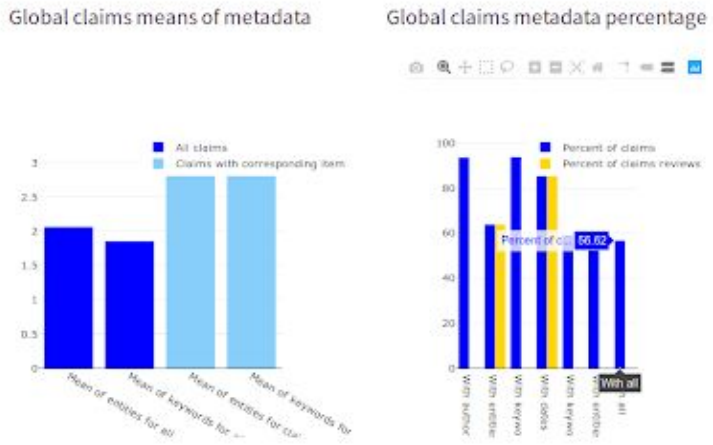
Pour reprendre ce qui est dit plus haut, le choix de flask correspond également à un argument de légèreté et de facilité de développement, de plus il a actuellement 44 153 étoiles sur GitHub contre son principal concurrent le framework Django qui en comptabilise 41,606.

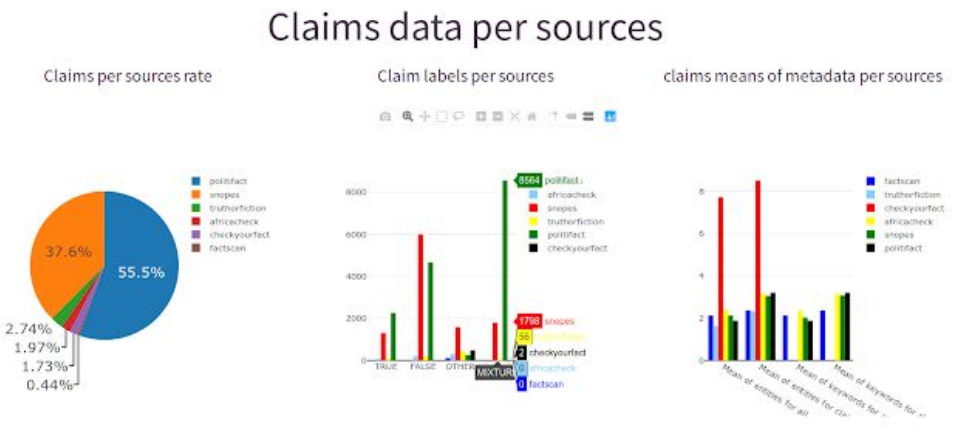
2.4 Mise en oeuvre

2.4.1 Réalisation de la page d'accueil

Les métadonnées que l'on va représenter particulièrement sont les auteurs des assertions, les dates de publications des assertions ou de leur reviews, les mots-clés et les entités. En terme d'UI (User Interface) un template responsive (qui s'adapte en partie à la taille de l'écran) HTML5,CSS3 et Javascript provenant de HTML5 Up Open Source à servi de base au design. Les points d'attention de réalisation ont été la gestion des valeurs manquantes et des doublons avec pandas. Mais également de varier les représentations graphiques. Les métriques choisies sont visibles sur les captures d'écrans ci-après.







2.4.2 Exploration thématique

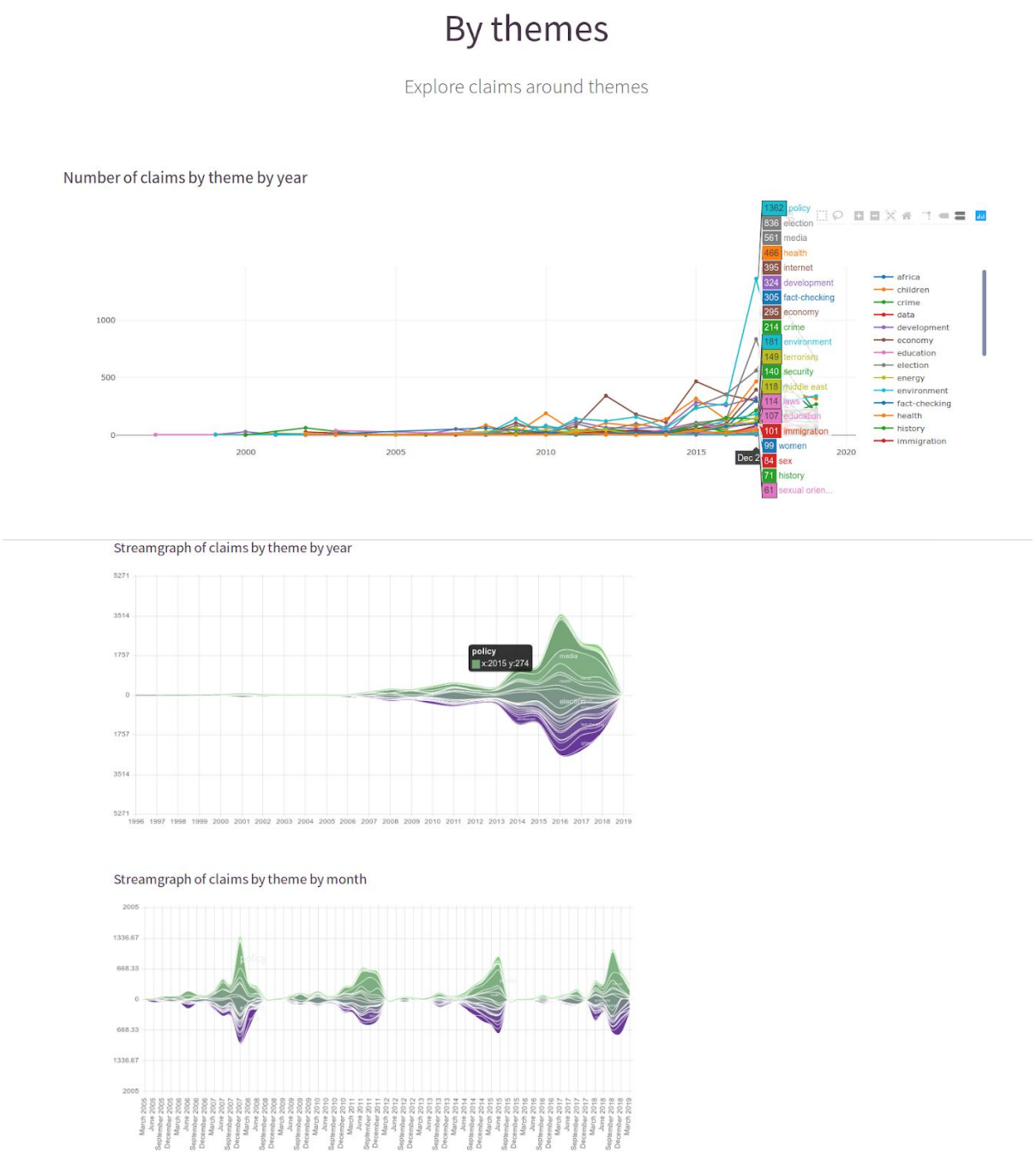
Afin de pouvoir répondre au use-case d'exploration par thème, l'objectif est de regrouper les assertions par thèmes, sujets, pour pouvoir les observer dans le temps. Le but de ce travail est d'être représentatif et non complet, on souhaite que les associations des thèmes aux assertions soient corrects, et la liste de thèmes créée pourra servir de base à un travail plus exhaustif. Plusieurs étapes ont été nécessaires. Premièrement, une phase d'analyse des données qui consiste à récupérer et traiter les mots-clés. Afin d'avoir un regroupement, ou clustering, représentatif, il faut prendre en compte les volumes relatifs des différents mots-clés, certains étant plus présents que d'autres. Ci-dessous un extrait des mots-clés les plus fréquents et le nombre d'assertions auxquels ils sont associés.

health care,1723	candidate biography,910	terrorism,447
economy,1717	crime,893	abortion,410
taxes,1487	fake news,853	criminal justice,406

education,1217	foreign policy,730	workers,391
immigration,1086	history,605	job accomplishments,386
jobs,1050	guns,590	transportation,374
federal budget,1032	legal issues,578	public health,368
state budget,983	environment,535	state finances,359
elections,983	energy,525	women,354
donald trump,925	military,459	children,334

Ensuite, une partie manuelle intervient afin de réduire les mots-clés en une liste de thèmes qui regroupera le plus possible des autres mots-clés. Ceci en se focalisant sur des thèmes plus globaux, en terme de fréquence d'occurrence mais aussi de sémantique. Par exemple ci-dessus on peut voir ‘health care’ et ‘public health’ on va ainsi créer un thème ‘health’ qui regroupe les deux entrées précédentes. En procédant de cette façon un dictionnaire est réalisé ayant pour clés les thèmes finaux (health) les plus évocateurs possibles pour les utilisateurs, et en valeurs des listes qui contiennent les autres mots-clés, ou mots atomiques faisant partie des mots-clés. Ce dictionnaire servira ensuite de filtre auquel on passera tous les mots-clés. Les listes contiennent en outre certaines racines de mots pour avoir un mécanisme de dérivation des chaînes de caractères. Par exemple, pour traiter ‘criminal justice’ et ‘legal issues’ la clé sera ‘laws’ (présent aussi dans les mots-clés) et les valeurs seront ‘justice’ et ‘legal issues’, afin de récupérer tous les mots-clés contenant ‘justice’. Un exemple pour les racines: le thème ‘economy’ comprendra la valeur ‘financ’. Cette étape réalisée un algorithme est utilisé pour filtrer les mots-clé qui va itérer à la fois sur la liste sans doublons des mots-clés extraits de la dataframe globale et sur le dictionnaire de thèmes en regardant si la chaîne de caractère d’un mot-clé est présent dans celui de la clé du dictionnaire ou de sa liste. Si la chaîne est présente alors on pourra lui affecter le thème correspondant à la clé du dictionnaire dans une nouvelle colonne de la dataframe. Plusieurs thèmes peuvent être affectés à une assertion (ce qui augmente la complexité), tout comme elle peut avoir plusieurs mots-clés. Dans le dictionnaire, les mots des listes ne s’excluent pas non plus le but étant d’avoir le maximum d’informations. Cette approche peut s’apparenter à un clustering manuel par curation utilisant en partie la cooccurrence de mots clés et leur fréquence. 34 thèmes au total sont définis. Il est à noter que la première version du dictionnaire était basée sur le fichier dump qui contenait 154 mots-clés contre 12517 actuellement, et qui s’appuyait également sur une prise en compte succincte des entités, la version actuelle comprend plus d’entrées, et une gestion sommaire des exceptions, pour éviter les problèmes posés par les mots de 3 à 4 lettres qui se peuvent se retrouver dans un grand nombre de chaînes de caractères. Cependant, ne sont pas traités ici les mauvais classements engendrés par des noms propres (personnes, villes etc.) qui devront dans un travail futur avoir été pré-traités et ne pas passer dans le filtre qui fonctionne donc sur la comparaison des chaînes. Le dictionnaire est disponible en annexe. Finalement, les assertions sont groupées par leurs thèmes affectés et par le temps désiré, mois, trimestres ou années, et utilisées pour créer les graphiques de variation en fonction du temps. Deux

représentations sur ces différentes granularités ont été réalisées et sont exposées sur les figures ci-dessous:



2.4.3 Exploration par source

Afin de pouvoir répondre au use-case d'exploration par source, l'objectif est de regrouper pour chaque source et chaque année ou chaque mois le nombre d'assertions par label, pour pouvoir les observer dans le temps. Après la construction de la dataframe, on extrait toutes les sources sans doublons, et on regroupe les labels par année.
Ci-dessous un extrait de la dataframe résultant pour la source politifact:

label	date	nombre

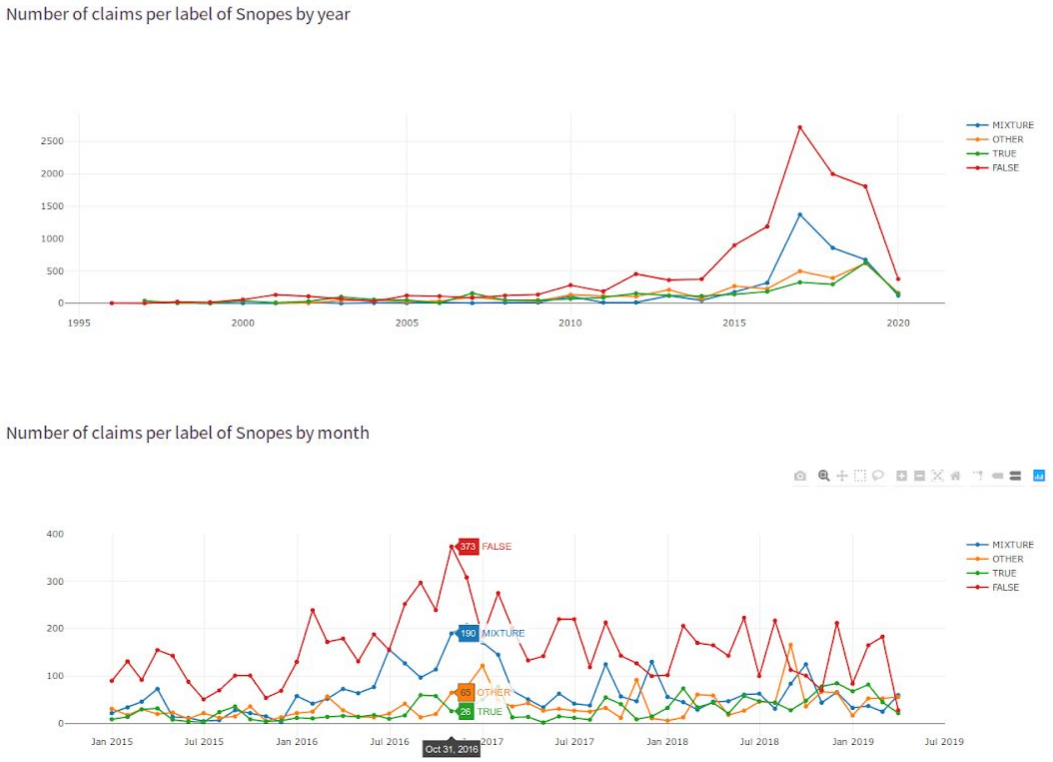
FALSE	2007	21
FALSE	2008	78
FALSE	2009	28
FALSE	2010	158
FALSE	2011	125
FALSE	2012	66
FALSE	2013	66
FALSE	2014	245
FALSE	2015	125
FALSE	2016	142
FALSE	2017	216
FALSE	2018	90
FALSE	2019	8003

Ci-dessous un extrait de la dataframe résultant par mois pour l'année 2015:

label	date	nombre
FALSE	2015-01	1
FALSE	2015-02	1
FALSE	2015-03	7
FALSE	2015-04	5
FALSE	2015-05	0
FALSE	2015-06	10
FALSE	2015-07	2
FALSE	2015-08	3
FALSE	2015-09	11
FALSE	2015-10	39

FALSE	2015-11	37
FALSE	2015-12	9

Ci-dessous une capture d'écran pour Snopes:



Afin d'analyser plus en détail les données un test de corrélation des labels par rapport à leur source a été effectué mais n'est pas encore exposé sur l'interface. Voyons maintenant comment analyser la relation statistique entre deux variables qualitatives grâce au test de khi-deux de contingence que l'on appelle aussi le test de khi-deux d'indépendance. Il mesure des écarts à l'indépendance, c'est à dire la distance numérique entre des effectifs observés et des effectifs théoriques. Il teste l'indépendance entre les lignes et les colonnes d'un tableau croisé. Autrement dit, il permet de se prononcer sur la « significativité » du lien entre deux variables qualitatives.

Une distribution constituée par deux variables dans un tableau de contingence est dite indépendante si la répartition des effectifs est équiprobable, c'est-à-dire que la répartition des effectifs est similaire à celle produite par le hasard.

Le test de Khi-deux se base sur les deux hypothèses suivantes :

H0 : les deux variables *X* et *Y* sont indépendantes

H1 : les deux variables *X* et *Y* sont dépendantes

Plus l'indicateur de khi deux est proche de zéro, plus le tableau des effectifs théoriques et celui des effectifs observées se confondent.

Tableau de contingence

	Label

source	faux	vrai	mixte	autre
politifact	11035	4860	19589	519
snopes	13928	3380	4560	3844
truthorfiction	302	190	110	904

Dans notre cas la valeur calculée de χ 2 est 16111.34 est largement supérieure à la valeur de χ2 lue dans la table de khi-deux avec 6 degrés de liberté, en même temps la probabilité critique du test est inférieure à 5%. Cela nous permet de rejeter l'hypothèse H0 et conclure que les labels des assertions sont dépendantes à leur source.

3. CONCLUSION

Ce travail nous a permis de contribuer à un projet Open Source, le code étant disponible sur GitHub et prochainement déployé (disponible dans le dossier GitHub de ClaimsKG sur le lien suivant: https://github.com/claimskg/claimskg-statistical-observatory). Cela nous a offert la possibilité de participer à notre échelle à la recherche sur le sujet du fact checking qui est un enjeu crucial dans notre société connectée. L'objectif principal a été réalisé en respectant les contraintes définies lors du recueil des besoins. L'interface à destination du grand public est créée, présente des statistiques synthétiques interactives. Elle donne un aperçu de données supplémentaires avec par exemple la recherche thématique, par source et le graphique 'Facebook and bloggers claims label'. Les figures sont légendées afin d'être le plus accessible possible. L'interface a également été validée par des parties prenantes comme le LIRMM, le LGI2P, le L3S et le GESIS. Enfin, les données sont synchronisées de manière semi-automatique avec le graphe ClaimsKG. Ce projet nous a permis d'apprendre, de mettre en pratique et de consolider nos connaissances en informatique et développement.

Plusieurs perspectives découlent de la réalisation de ce projet, décrites ci-après.

Réaliser une analyse poussée de besoins métiers des acteurs qui pourraient utiliser l'interface afin d'enrichir les fonctionnalités.

En terme d'architecture plusieurs possibilités de raffinements sont envisageables. On pourrait adopter une architecture se rapprochant des microservices. En ayant par exemple deux applications, une gérant le calcul des statistiques et la production de graphiques et l'autre l'affichage de ces derniers. On pourrait également ajouter un autre framework comme Vue.js et faire quelque chose de plus moderne avec une SPA (Single Page Application) où flask servirait d'API renvoyant du JSON à Vue.js qui calculerait les composants et les rendrait. Cela pour permettre de mieux découpler affichage et rendu, et pourrait être plus simple pour le développement que l'utilisation de Jinja2, c'est à dire de mettre du python dans du html.

Pour la partie d'exploration thématique une bonne approche serait de réaliser un thésaurus complet des thèmes s'appuyant sur des ontologies par domaine sémantique, déjà établies ou validées par des experts, afin de trier les assertions. En effet, le thésaurus est souvent une étape préliminaire dans la conception d'une interface exposant des ressources organisées et liées entre elles afin d'organiser la navigation, c'est le cas des sites de

commerces en ligne, les sites regroupant des articles d'actualités ou encore des moteurs de recherche. Un thésaurus est un ensemble de termes normalisés et organisés servant à l'indexation documentaire et informatique. Que cela soit fait avec des outils spécifiques ou manuellement, il nous permettrait également d'avoir une hiérarchie entres les thèmes par exemple pour le thème de la santé on pourrait avoir un section santé publique. Cette hiérarchie serait intéressante pour observer les thématiques à différents niveaux de granularités, par exemple, du plus global au plus précis.
Développer les fonctionnalités proposées, par sources et par entités. En terme de statistique, on pourrait construire des variables quantitatives à partir des variables qualitatives du graphe, par exemple le nombre d'entités et mots-clés par mois, puis calculer le coefficient de corrélation de Spearman ainsi que le tau de Kendall qui mesure la corrélation de rang entre deux variables. D'autres fonctionnalités comme la réalisation de classement des entités en terme de nombre d'assertions et labels auxquels elles sont associées, pourrait être intéressant. Pour les entités et les auteurs il serait pertinent de produire une analyse des personnes physiques, des thèmes auxquels ils sont associés, dans le temps. Notamment afin de dégager des historiques, et des comportements de médiatisation aux personnalités politiques et publiques. Mais aussi d'avoir une indication sur la confiance que l'on peut placer dans les propos de ces mêmes auteurs et entités. Ces analyses pourrait également se décliner par source afin de voir si certains sites de fact checking ont des cibles préférentielles. On pourrait également regrouper les entités par groupe politique ou idéologique en ajoutant de nouvelles données avec DBpedia par exemple. Enfin pour enrichir les analyses temporelles, il serait intéressant de croiser les données de ClaimsKG avec d'autres graphes comme EventKG pour associer des variations avec des événements d'actualités, et tester des corrélations.

4. BIBLIOGRAPHIE

Sites internet
Web sémantique:
https://fr.wikipedia.org/wiki/Web_s%C3%A9mantique
Fake news et fact checking:
http://www.apar.tv/web/liberte-du-net-en-2017-la-manipulation-des-reseaux-sociaux-pour-affaiblir-la-democratie/
https://www.cairn.info/revue-des-sciences-de-gestion-2010-2-page-17.htm#
https://www.franceinter.fr/politique/les-reseaux-en-campagne-vaste-audience-et-petites-manipulations
https://www.la-croix.com/Economie/Medias/Barometre-medias-Francais-veulent-information-verifiee-2017-02-02-1200821914
https://reporterslab.org/fact-checking-triples-over-four-years/
https://www.lesoleil.com/actualite/monde/fake-news-une-expression-nouvelle-pourde-vieilles-histoires-dba52376b2d3e604c8f53f2f7901150a
https://www.futura-sciences.com/tech/definitions/informatique-fake-news-17092/
Etat de l'art:
http://www.emergent.info
Choix des technologies:
https://www.tiobe.com/tiobe-index/

https://www.developpez.com/actu/217533/Meilleurs-langages-en-2018-selon-l-IEEE-Python-conforte-sa-place-de-leader-grace-a-son-ascension-dans-le-machine-learning-et-l-embarque/
Test d'indépendance :
http://mehdikhaneboubi.free.fr/stat/co/khi_deux.html

Article
ClaimsKG et schéma du modèle:
ClaimsKG: A Model and a Live Knowledge Graph of Annotated Claims

Etat de l'art:
Knowledge Vault - X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 601–610. ACM, 2014
KnowMore - R. Yua, U. Gadirajua, B. Fetahua, O. Lehmbergb, D. Ritzeb, and S. Dietzea. Knowmore-knowledge base augmentation with structured web markup. Semantic Web Journal, IOS Press, 2017.

ANNEXE

1. Contenu du dictionnaire des thèmes

Thème	Liste de mots-clés et radicaux associés
Development	agriculture, population, transport, city planning, marriage, construction, jobs, suicide, suicide rate, reading, sanitary products, human rights, food, hunger, infrastructure, sanitation, housing, literacy rate, tourism, science, transformation, research, freedom of speech, workers, ethics, religion, public safety, social security, polls and public opinion, families, diversity, museum, social justice warriors
History	museum
Statistics	rate, ranking, stats
Media	news, journal, magazine
Energy	electricity, eskom, oil sands, solar power, solar panel, gasoline, gas company
Health	hiv/aids, world health organisation, cancer, alcohol, malaria, drugs, ebola, malnutrition, vaccination, traditional medicine, mental health, zika, hypertension, smoking, tobacco, tuberculosis, hiv, diabetes, disease, infection, contraception, care, abortion, drug, diarrhea, vaccin, reboiled water

Women	sexual violence, income inequality, rape, teen pregnancy, girls, sex trade stats, sex work, inequality, femicide, female genital mutilation, gender inequality, gender-based violence, gender equality, abortion, woman, sexism
Reader suggestion	suggested by reader
Economy	employment, poverty, income inequality, unemployment, debt, welfare, business, budget, economic freedom fighters, gdp, income tax, tea trade, civil servant spend, revenue spend, inflation, trade, wealth, budget resources, revenue, aid, tax, jobs, financ, economi, taxes, workers, deficit
Violence	sexual violence, gender-based violence, abuse, harassment, assault
Crime	human trafficking, rape, genocide, murder, slavery, sexual violence, gender-based violence, abuse, harassment, assault, theft, robbery, pedophilia, pedosexual, sex trafficking, sex with animals, watergate
Laws	law, prisons, courts, justice, judiciary, human rights, legal issues, supreme court, civil rights
Africa	jacob zuma, sona, nigeria, south african police service, buhari, kenya, south africa, eskom, cyril ramaphosa, zuma, western cape, raila odinga, julius malema
Education	children, teachers, school
Children	sexual violence, child healthcare, child mortality rate, teen suicide, teachers, teen pregnancy, pedosexual
Fact-checking	fakes, hoax, fake news, fake quotes, fact check, fake campaign, fake muslim crime, fake crime
Youth	children, school, teen suicide, teen pregnancy, child healthcare, child mortality rate
Policy	government, sona, election, elections, democratic alliance, corruption, unemployment, land reform, state of the nation, counties, literacy rate, security, civil servant spend, democrat, campaign, republica, candidate biography, watergate, midterms
Election	elections, candidate biography, voting record, midterms
Security	police, fbi, guns, weapon, federal bureau of investigation, military, nuclear, navy seals
Environment	water, pollution, wildlife, drought, sea, ocean, carbon, climate,

	fisheries, fishery, dolphins, kyoto protocol, animals, oil spill, seal
Immigration	migration, xenophobia, refugees, human rights, diaspora, immigrants
Terrorism	islamic state, boko haram, 9/11, isis
Middle east	egypt, iran, iraq, turkey, saudi arabia, yemen, syria, jordan, united arab emirates, israel, lebanon, palestine, oman, kuwait, qatar, bahrain
Industry	mining, informal mining, aviation, eskom, aluminum foil, oil sands, palm oil, gas pipeline
Data	open data, data mining, internet
sex trade stats	_
Sex work	_
Sexual orientation	homosexual, bisexual, transexual, transsexual, heterosexual, asexual, pansexual, transgender, gays and lesbians, equality nc, gay, same sex, same-sex
Sex	sexual abstinence, sexual stimulation, sexual intercourse, contraception
Internet	facebook, twitter, instagram, snapchat, bloggers, internet rumors, memes, viral video, 4chan
Sciences	science, research, thermodynamics