

Mémoire

Rémy Pocquerusse

October 2017

L'utilisation de l'intelligence artificielle  
pour la lutte contre la désinformation

# Sommaire

<b>1</b>	<b>Introduction générale</b>	<b>4</b>
1.1	La désinformation . . . . .	5
1.2	Fake news . . . . .	6
1.3	Intelligence artificielle . . . . .	7
1.4	Fake news et intelligence artificielle . . . . .	7
1.5	Fact-checking . . . . .	7
1.6	Objectifs . . . . .	10
<b>2</b>	<b>Outils et techniques</b>	<b>11</b>
2.1	Web sémantique . . . . .	11
2.1.1	RDF . . . . .	12
2.1.2	Ontologie . . . . .	13
2.1.3	Exemple : Yago . . . . .	14
2.1.4	SPARQL . . . . .	15
2.1.5	Wikidata . . . . .	16
2.2	Claimbuster . . . . .	17
2.3	Knowledge graph . . . . .	22
2.3.1	Wikipédia knowledge graph . . . . .	22
2.3.2	Finding Streams in Knowledge Graphs to Support Fact Checking . . . . .	26
2.4	Fact checking dans un réseau de flot . . . . .	28
2.4.1	Knowledge Stream . . . . .	28
2.4.2	Relational Knowledge Linker . . . . .	31
2.5	Critique . . . . .	32
2.6	Conclusion . . . . .	33
2.6.1	Google knowledge graph . . . . .	33
2.7	Délégation aux apis . . . . .	34
2.8	Falsification d'image . . . . .	34
2.9	Vérification de la source du lien . . . . .	34
2.10	Développement prototype . . . . .	34
2.10.1	Prédire la tromperie . . . . .	34
2.10.2	A faire . . . . .	35
	<b>Bibliographie</b>	<b>36</b>
	<b>Appendices</b>	<b>40</b>

<b>A</b>	<b>Knowledge Graph</b>	<b>40</b>
A.1	Fragment du résultat retourné par le google knowledge graph pour l'entité "Joseph Boyden" . . . . .	40
A.2	Fragment du résultat retourné par le google knowledge graph pour l'entité "Three day road" . . . . .	42

# 1 Introduction générale

Durant mon stage de Master 1, j’ai effectué une mission qui tournait autour d’une problématique simple : donner la possibilité à un utilisateur de décrire une chose et comprendre l’objet de cette chose côté machine. Plus précisément il me fallait catégoriser un texte pour pouvoir ensuite le classifier (parle-t-on de science, d’écologie ?, etc.). Quelques pistes m’ont été données pour commencer mes recherches, notamment le traitement du langage naturelle avec les possibilités liées au web sémantique. C’est en explorant les ambitions du web sémantique que je me suis aperçu des enjeux d’un web ouvert aux données. En formulant des requêtes en langue naturelle ou structurée (type SQL) il est possible de récupérer des données formatées et liées, exploitables par une machine. Avec l’enjouement pour l’intelligence artificielle qui a explosé grâce au big data [12], la donnée est devenu précieuse. Beaucoup de problématiques de calcul et d’apprentissage tournent autour de la donnée, dans le machine learning par exemple il faut des données fiables, structurées et en grande quantité. La création de dataset prend du temps et nécessite souvent une intervention externe du fait de la quantité de données à traiter manuellement. Par exemple ClaimBuster (outil que nous présenterons plus tard), a nécessité 3 mois de travail et 226 participants pour créer un dataset viable [14].

Pourtant ces données sont disponibles sur le web. N’y aurait-il pas un moyen de réduire ce temps de 3 mois au temps d’une simple requête ?

C’est à une de ces problématiques que le web sémantique (dont nous détaillerons le fonctionnement plus tard) tente aujourd’hui de répondre. Le web sémantique avec l’intégration de la sémantique dans le web vise à donner du sens aux données. Le mot Socrate ne réfère plus à un simple mot mais à un nom, à une personne qui possède une date de naissance, une nationalité, etc. Chaque donnée va référer à un objet qui possède des attributs universels qui le décrivent [21].

Le web sémantique a déjà eu un impact important sur le web mais méconnu, ce qui peut s’expliquer par le fait qu’il est souvent défini comme un outil fait par des scientifiques pour des scientifiques [2].

Un des cas les plus important (et visible) est l’intégration de la sémantique dans le système de recherche de Google. Cela se traduit par des informations organisées à l’écran comme les infoboxes (panel situé à droite qui détaille notre recherche) ou encore des réponses directes aux recherches, exemple :

“Quel est le vrai nom de molière ?” va nous afficher directement la réponse.

Le web sémantique a ouvert toutes sortes de données autour de différents thèmes : la médecine, le e-commerce, etc. La plupart des CMS permettent à leur utilisateurs de structurer leur données et de les ouvrir au reste du monde. Par exemple, le CMS Drupal permet de produire des données liées [6]. Ce n’est pas négligeable quand on sait qu’il fait tourner environ 2.1% de tous les sites de type CMS [32] (il est aussi possible de le faire sur la plupart des CMS).

En médecine on peut noter Bio2RDF, projet open source qui regroupe plus de 11 milliards de triples relatifs aux sciences de la vie et à la recherche clinique. On peut aussi citer DBpedia qui est une référence dans le domaine du web sémantique et qui s’est fixé pour but, depuis 2007, de produire des données liées avec des informations extraites de wikipédia.

Toutes ces données sont facilement maniables et interconnectables. Dans le cas présent nous allons voir comment ces données sont utilisées et pourrait être utilisées pour lutter contre la désinformation.

**TODO : A revoir**

## 1.1 La désinformation

“La désinformation est un ensemble de techniques de communication visant à tromper des personnes ou l’opinion publique pour protéger des intérêts (privés ou non) et/ou d’influencer l’opinion publique.” [34]

La désinformation est donc un acte conscient et organisé pour nuire à un groupe de personnes. Elle peut se traduire par de la propagande, du prosélytisme ou de la manipulation sur tout type de sujet. Elle peut prendre tout type de configuration, mais prend souvent la forme d’allégations alarmantes ou révoltantes qui vont frapper le lecteur. Le but est de susciter une émotion forte pour que le lecteur s’empresse de partager l’information. Un mensonge plus frappant qu’une vérité se diffusera plus vite et plus loin avec un impact plus important. On estime par exemple, que sur twitter, une fake news a 70% de chance en plus d’être partagée qu’une autre information [30]. Ainsi la désinformation se traduit plus généralement par une tentative de manipulation de l’opinion publique (exemple ici) en transmettant des informations

partiellement erronée (il est plus facile de croire à un mensonge enrobé de vérité).

Il faut tout de même différencier la désinformation politique dont sont à l'origine des grandes organisations ou des états (on parle souvent de propagande) de la désinformation économique qui touche plutôt au buzz et à toute forme de reconnaissance (réseaux sociaux, etc.).

## 1.2 Fake news

Sur internet quand on parle de désinformation, on parle de fake news ou fausses nouvelles. On définit une fake news comme étant une information dont le but est de tromper consciemment un lecteur. Une fake news peut donc être définie comme étant une tentative de désinformation mais qui pour la plupart visent à induire en erreur et créer du buzz autour d'un fait. Elle sont particulièrement présentes sur les réseaux sociaux et dans les usines à clics. Mais on les retrouve aussi dans la presse spécialisée ou encore dans les médias politisés dont le but est la propagation de fausses informations pour servir leurs intérêts (retouche d'image, etc.). Mais un article faux n'est pas forcément une fake news tant qu'il n'y a pas l'intention de tromper, l'article peut avoir des fins humoristiques ou satiriques par exemple.

Une information se base sur un fait réel et donc vérifiable. Afin d'identifier une information comme étant fausse il faut pouvoir le prouver factuellement. Il est donc important de connaître l'existant (un ensemble d'informations vraies) pour pouvoir identifier une information fausse. Ici le but n'est pas de lutter contre ou éradiquer les fake news (cela reste de la liberté d'expression, dans une certaine mesure). Nous chercherons simplement à les identifier.

TODO : citer l'étude sur l'impact des fake news durant la campagne américaine (voir dans présentation ebs)

**Pourquoi favoriser leur détection ?** A cause de l'impact sur le monde réel que cela peut avoir. Susciter une vive émotion dans l'opinion publique est une des finalités de la fake news. Cela peut avoir des conséquences graves sur les événements qui se déroulent (réponse par la violence, la haine). C'est un constat encore plus vrai en temps de guerre ou durant des périodes sensibles.

### 1.3 Intelligence artificielle

Le but de l'intelligence artificielle (IA) est de créer des machines intelligentes capables de comprendre leur environnement et de performer des actions leur permettant d'atteindre un but précis. Cela se traduit par l'imitation des facultés humaine comme la résolution de problèmes ou l'apprentissage.

**Le Machine learning** fait référence a la capacité d'un système à apprendre de données formatées pour ensuite être capable de prédire quelque chose [5].

TODO : A finir

### 1.4 Fake news et intelligence artificielle

Détailler l'utilité de l'intelligence artificielle pour la détection de fakenews.

Machine learning ?

Vérification de source.

Vérification d'un fait : vérification d'une image ?

### 1.5 Fact-checking

TODO : Définition du fact checking et dérivés (ex: fact checking system)

TODO : Cela revient à demander à une machine de faire du journalisme. Parler des enjeux.

TODO : Comment le fact checking intègre l'intelligence artificielle ?

TODO : A voir, <https://fullfact.org/automated>

TODO : A voir : <http://idir.uta.edu/factwatcher/nba.php>

Le fact-checking ou vérification de faits représente l'acte de vérifier la véracité d'une assertion. Il s'agit de traiter l'information afin de démêler le vrai du faux, du partiellement vrai du partiellement faux.

Seulement les sources d'informations sont innombrables et le flux de données journalistique est tel qu'il est impossible de vérifier chaque information manuellement. Que ce soit sur des blogs, les réseaux sociaux, les sites d'information, etc. filtrer l'information et la vérifier représente un temps de travail important.

**TODO : Parler du fait que le fact-checking est une nouvelle forme de journalisme**

**Initiatives contre les fake news** Que fait-on dans le monde réel pour lutter contre les fakenews ? Survey of initiatives against fake news : [11]

**TODO : parler de facebook/google suite à la campagne américaine**

**Fact-checking : les limites** Le fact-checking est un travail qui est chronophage, en effet il faut tout d'abord trouver et identifier les faits vérifiables et dignes d'intérêt. Il y a ensuite un travail de recherche pour comprendre le fait et le vérifier en faisant attention de ne pas tomber soi-même dans le piège de la désinformation. Cela signifie confronter et vérifier ses sources. Tous ces critères font qu'entre le moment où une déclaration est faite, et celui où elle est vérifiée s'écoule un certain laps de temps, qui peut éventuellement rendre le fact-checking inutile si la déclaration a déjà eu l'impact voulu.

**Quels objectifs ?** L'objectif derrière le fact checking reste actuellement très politique, il s'agit de traquer les mensonges de personnalités influentes.

Mais à terme le fact checking doit permettre de toujours manipuler une information avérée et sûre quelque soit sa source ou sa forme.

Il n'existe actuellement aucun système universel, 100% automatisé et 100% fiable qui permettent de réaliser des opérations de fact-checking sur tout type de support. La finalité d'un système de fact-checking serait tout d'abord d'être capable de caractériser, classifier et comprendre l'information pour ensuite vérifier sa légitimité.

**TODO : Définir les étapes de vérification d'un fait**



**Système entièrement autonome ou du moins le plus possible** Quelle serait les tâches d'un système entièrement autonome ? Quelle est la faisabilité de chacune de ces tâches ? Et enfin quelle marge d'erreur est on prêt à accepter avant de définir notre système comme entièrement autonome ?

Un tel système doit pouvoir extraire une assertion d'une source et la vérifier sans intervention humaine et dans un temps raisonnable. Pour cela il doit pouvoir se "nourir" de différentes sources d'informations qu'il va confronter.

**Challenges** Comme nous l'avons précisé pour qu'un fait soit analysé il faut tout d'abord qu'il soit compris. Il faut donc faire comprendre au système le langage humain, c'est ce qu'on appelle le traitement automatique du langage naturel (TALN). C'est un domaine vaste qui fait appel à de nombreux sous domaines dans le traitement du langage comme l'analyse syntaxique ou l'analyse sémantique.

**TODO : Trouver un papier pour référencer ça : TALN loin d'être parfait**

Mais un système aussi performant soit-il ne fonctionnera pas sans données fiables. Il faut pouvoir trouver et collecter ces données pour alimenter le système. Les sources de données structurées et utilisables sont innombrables, que ce soit avec le développement du web sémantique, du linked-data, des bases de connaissances, des apis de type REST (Wikimedia) et de l'open data on possède déjà des sources de données conséquentes et fiables. On a ensuite des sources de données plus générique, moins ordonnées mais tout aussi exploitable. Je parle ici des sites d'information et plus généralement de toutes les informations disponibles sur internet : réseaux sociaux, etc. Ce sont des sources de données plus ou moins fiables mais la quantité d'information produite par ces sources est considérable. Ces sources peuvent être interrogées via des outils de type web scraping et data mining. Toutes ces sources de données peuvent être confrontées pour permettre de chercher et étoffer un fait pour pouvoir le comprendre et le vérifier. Mais les sources de données ne se limitent pas à ce que l'on trouve sur internet.

On identifie donc 3 sources de données différentes, les données :

- Déjà traitées par des sites spécialisés ou communautaires : beaucoup de sites d'information ont développés leurs propres systèmes de fact-checking (tous manuels).

- Structurées ou semi-structurées : issues de l'open data, ou d'api telles que wikimédia
- Non-structurées : issues de toutes sources potentiellement utilisables

Les données non structurées peuvent se trouver directement dans la vie réelle à travers différents médias : les chaînes de télévision, les émissions de radio (mais aussi les journaux, magazines), etc. Tout signal vidéo ou audio est potentiellement une source intéressante pour le système. Il faut pouvoir interroger et structurer ces sources : savoir qui parle, dans quel contexte.

Une fois ces informations structurées nous devons savoir lesquelles définissent des faits réels et vérifiables. Filtrer les informations en fonction de la vérifiabilité d'une assertion. C'est-à-dire éliminer toute phrase qui définit une opinion, de l'humour, de l'ironie, figures de style, etc. et bien sûr les phrases lambda. De plus comment détecter qu'un fait se construit sur plusieurs phrases ? Comment construire une argumentaire appuyé par des preuves concrètes qui permette de remettre en cause le fait.

## 1.6 Objectifs

**TODO : But du mémoire : voir l'ensemble des techniques viables et proposer un système cohérent reprenant les atouts de chaque méthode/concept + architecture technique**

J'ai choisi ce sujet car il fait appel à des technologies et des domaines de recherches très nombreux que j'ai souvent survolé : web sémantique et linked-data, analyse sémantique, bases de connaissances, traitement automatique du langage, machine learning, knowledge graph, web scraping, web crawling, data mining, etc.

En voulant analyser l'information de façon automatique on cherche ici à cartographier et comprendre les ressources disponibles sur le web. Analyser toutes les données disponibles pour en comprendre le sens, et construire des modèles cohérents permettant de structurer l'information pour élaborer des modules basés sur des algorithmes d'apprentissage efficaces. C'est-à-dire récupérer des informations, les structurer et construire des algorithmes issus de ces données par apprentissage supervisé. La finalité de ces algorithmes est multiple : détection de fake news, détection d'image, etc. Ce que je veux montrer ici c'est l'enjeu derrière un web structuré : je souhaite obtenir un

dataset d'images me permettant d'entraîner mon algorithme de détection de visage ? Je n'ai qu'une simple requête à faire. C'est tout à fait possible sur des bases de données de type Triplestore [36] qui vont organiser la donnée comme un graphe connecté ou chaque noeud est relié par une relation. C'est un objectif très vaste voir impossible (TODO : citation sur les détracteurs du web sémantique). Mais pour lutter contre la désinformation et arriver à un système de fact checking efficace et autonome c'est un prérequis important.

TODO : Faire évoluer le but du mémoire en fonction des résultats Le but de ce mémoire est d'étudier différentes approches utilisées dans le fact checking automatique pour pouvoir proposer un système viable et cohérent. Nous tenterons de construire un système reprenant les atouts de chaque méthodes et concepts afin de proposer une architecture technique qui tente de répondre à plusieurs problématiques rencontrées par les systèmes existants.

TODO : Plutôt que d'essayer de vérifier la véracité d'un fait pourquoi ne pas collecter des infos autour de ce fait, construire un argumentaire et laisser l'utilisateur être seul juge ?

## 2 Outils et techniques

### 2.1 Web sémantique

"To a computer, the Web is a flat, boring world, devoid of meaning. This is a pity, as in fact documents on the Web describe real objects and imaginary concepts, and give particular relationships between them. For example, a document might describe a person. The title document to a house describes a house and also the ownership relation with a person. Adding semantics to the Web involves two things: allowing documents which have information in machine-readable forms, and allowing links to be created with relationship values. Only when we have this extra level of semantics will we be able to use computer power to help us exploit the information to a greater extent than our own reading." - *Tim Berners-Lee "W3 future directions" keynote, 1st World Wide Web Conference Geneva, May 1994* [1]

Revenons en au web sémantique afin de clarifier quelques termes et approfondir son mode de fonctionnement. Cela nous permettra de mieux com-

prendre tous les outils liés à la sémantique comme les knowledge graph. Actuellement la navigation et la recherche d'information sur le web nécessitent une action humaine. Une machine n'est pas encore capable de rechercher et d'analyser efficacement des données sur le web. La collecte de données automatique est possible mais la machine ne peut déterminer de manière fiable le type d'entité, les thèmes, les relations, le contexte, etc... des données qu'elle manipule.

Les aspirations du W3C sont de créer un web exploitable par des machines en le transformant en une base de connaissance géante. La navigation sur le web est possible via des hyperliens, cependant elle doit aussi pouvoir se faire au niveau de données structurées pour que les machines puissent exploiter de façon plus efficiente et précise les données contenues sur le web.

TODO : Parler de la couche Proof ? [http://www.kbs.uni-hannover.de/Lehre/semweb07/10\\_proof\\_trust.pdf](http://www.kbs.uni-hannover.de/Lehre/semweb07/10_proof_trust.pdf) et <https://sites.ualberta.ca/~reformat/ece720w2012/papers/IEEEExplore-8.pdf>

Le web sémantique ou web 3.0 est une extension du web, standardisée par le W3C. Ces standards recommandent l'utilisation de format de données et de protocoles normés, plus généralement il s'agit du RDF (Resource Description Framework).

### 2.1.1 RDF

Le RDF (Ressource Description Framework) est un modèle de description des données permettant l'échange entre différentes applications. Il permet la structuration, l'indexation et la standardisation des données disponibles sur le web. Un schéma simple est utilisé pour structurer les échanges et les relations entre les ressources (documents, personnes, concepts abstraits...). Sachant qu'une ressource est représentée par une IRI (International Resource Identifier), par définition unique, il est possible de lier différentes ressources entre elles en utilisant un triplet : <Subject, Property, Object> ou Subject et Object sont définis par des IRI. La propriété d'une relation spécifie la nature du lien entre les 2 ressources : isChildOf, diedIn... Pour spécifier ces relations, le RDF utilise le langage XML (ou RDF/XML) [31].

Ainsi le web sémantique ressemble à un graphe géant regroupant plusieurs milliards de liens (triplets) permettant une navigation et une interrogation des données plus efficaces. En respectant le RDF, une entreprise, une univer-

sité ou n'importe quelle entité peut ouvrir ses données au web afin qu'elles soient exploitables par n'importe quelle machine.

Le RDF permet l'utilisation de différents "vocabulaires" ou ontologie pour traiter les données. Il pourrait être comparé à une grammaire et une ontologie à son vocabulaire. Par exemple FOAF (Friend Of A Friend) est une ontologie permettant la structuration de personnes. [9]

### 2.1.2 Ontologie

Une ontologie permet de représenter les connaissances en les organisant sous forme de classes, de relations, de règles (ou implication) etc... Le but ici est de structurer les informations collectées sur le web.

Une ontologie se structure en essayant de comprendre le monde : comment est structuré notre monde, par quels concepts est-il défini ? Quelles sont les relations entre ces concepts permettant d'expliquer notre monde. Une ontologie ne s'intéresse donc pas à ce qui est possible ou ce qui pourrait être mais s'intéresse à ce qui est. [25] [22]

Toutes ces règles, conventions, standards et vocabulaires ont pour objectif de normaliser l'échange de données liées au web afin de construire un web dans lequel les ressources sont identifiables non par des urls mais par des données. Ainsi deux sources de données différentes qui utilisent les mêmes ontologies peuvent fusionner sans problèmes, et l'interrogation distincte de ces sources se fait avec les mêmes requêtes.

Par exemple, comme dans la figure 5, et via The Open Graph protocol (OGP) il est possible d'intégrer des balises HTML qui vont ajouter de la sémantique à un objet [28]. Cela permet de rendre cet objet interopérable. Il pourra être ensuite traité de façon complexe avec d'autres objets.

```
7 <meta property='og:type' content='website'>
8 <meta property='og:url' content='http://www.knowledgevault.com/'>
9 <meta property='og:image' content='http://www.knowledgevault.com/images/KVault_bug.jpg'>
10 <meta property='og:title' content='Private Knowledge Sharing Platform - Connect Share Learn'>
```

Figure 1: Exemple d'utilisation de l'open graph dans le code source d'une page : <http://www.knowledgevault.com/>

Ces données peuvent ensuite être collectées pour être rassemblées et raffinées dans des bases de connaissance telles que Yago.

### 2.1.3 Exemple : Yago

Une KB (knowledge base ou base de connaissance) est une base de données utilisée pour stocker et structurer du contenu lié à un domaine spécifique afin qu'il soit exploitable par une machine. Une KB lie ses données entre elles en y appliquant des règles, des faits ou des prédicats. A partir de ces constats, la KB peut déduire et faire évoluer la structuration de ses données. Prenons un exemple simple, j'ai une ontologie qui décrit les relations entre les personnes. En particulier cette ontologie me dit que si 2 personnes ont une même mère alors elles ont une relation fraternelle. Ainsi si j'enregistre 1 personne, ici la mère, et que je lui attribue un lien de maternité avec deux autres personnes, ma base sera capable de créer automatiquement un lien de fraternité entre ces 2 personnes.

Certaines KB ont pour but d'exploiter les données du web sémantique, la plupart se cantonne néanmoins, pour le moment, à la structuration des données présentes sur des domaines spécifiques, notamment wikipédia. On peut citer par exemple DBpedia ou Wikidata, ou d'autres encore qui tentent d'étendre leur champ d'action à d'autres sources comme Yago.

Yago est une KB créée par l'institut Max-Planck d'informatique à Sarrebrück en 2008. Contrairement à DBpedia ou Wikidata qui s'appuient sur un effort communautaire, le but de Yago est d'extraire des données de sources différentes (actuellement Wikipedia et Wordnet). Tout cela automatiquement et en plusieurs langues. Beaucoup de KB comme Yago s'organise sous la forme de graphe. L'information dans ces bases est représentées par un graphe géant constitué de millions de triples. Ces graphes sont appelés graphes de connaissance.

Les KB comme Yago permettent d'interroger Wikipedia et différentes sources avec des requêtes précises, par exemple : "Les peintres français du 16ème siècle" ou "Les villes de Chine de plus d'un million d'habitants". Les requêtes après traitement se font en langage naturel et retourne des données structurée et formatée pour être utilisée directement avec le système. Yago regroupe actuellement des informations sur plus de 10 millions d'entité (per-

sonnes, villes, etc.) et plus de 120 millions de faits sur ces entités. Il existe de nombreuses bases de connaissances qui contiennent des données hétérogènes comme Yago mais d'autres se concentrent sur des domaines spécifiques comme la médecine (ex : Precision Medicine Knowledgebase). Ces bases apportent une matière essentielle à la lutte contre la désinformation, ce sont des sources de données fiables et dont le requêtage est facilement automatisable. Le langage d'interrogation de ces données est le SPARQL.

Yago est disponible en open source [ici](#).

#### 2.1.4 SPARQL

TODO : SPARQL et Wikidata : à supprimer une fois intégré

TODO : Exemple complet de fact checking

SPARQL (Protocol And RDF Query Language) est un langage de requête orienté données permettant d'interagir avec des données RDF à travers le web. SPARQL permet de questionner le web sémantique, plus précisément les triplets, qui forme le graphe géant du web sémantique.

Exemples pour Yago (n'est plus accessible sur les SPARQL endpoints donc non testable pour le moment) :

Liste des prédicats associés à l'entité France.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yago: <http://yago-knowledge.org/resource/>
SELECT DISTINCT ?x WHERE {
    yago:France rdf:type ?x.
}
```

Autre exemple sur DBPedia, dans quelles catégories s'inscrit une voiture ?

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dct: <http://purl.org/dc/terms/>

select ?categorie , (COUNT(?result) as ?numberResult)
where {
    ?result dct:subject ?categorie.
    ?search rdfs:label "Voiture"@fr .
}
```

```
?search <http://dbpedia.org/ontology/wikiPageWikiLink>  
?categorie  
} ORDER BY ?numberResult  
LIMIT 1000
```

Tester ici

### 2.1.5 Wikidata

**TODO : A mettre en forme et à intégrer sur un eexemple de fact checking**

Info : dans une entrée, si ns = 14, alors c'est une sous catégorie de l'objet recherché. Ici les requêtes sont limitées à 10 résultats.

Faire une recherche :

```
https://fr.wikipedia.org/w/api.php?action=query&list=search&format=jsonfm&srsearch=Panneau+solaire&utf8=1
```

Rechercher une catégorie :

```
https://fr.wikipedia.org/w/api.php?action=query&titles=Panneau+solaire&prop=categories&clshow=hidden&utf8=1
```

clshow=hidden permet d'afficher les catégories cachées comme les portails qui sont bien plus génériques que les catégories.

clshow=!hidden permet de n'afficher que les catégories spécifiques de l'article, la recherche est bien plus simple.

Catégories liées aux Technologies :

```
https://fr.wikipedia.org/w/api.php?action=query&list=categorymembers&cmtitle=Category:Technologie&utf8=1
```

Catégories principales de wikipédia (portails, en anglais)

```
https://en.wikipedia.org/w/api.php?action=query&list=categorymembers&cmtitle=Category:Main%20topic%20classifications&utf8=1
```

Plus d'infos :

```
https://www.mediawiki.org/wiki/API:Categories
```



<https://m.mediawiki.org/wiki/API:Query#Generators>

## 2.2 Claimbuster

Nous allons maintenant étudier un des outils les plus évolués en matière de fact-checking automatique. Cette étude va nous permettre de mettre en lumière les enjeux et challenges pour aboutir à un système autonome.

### **Claimbuster** [13] [14]

ClaimnBuster est ce qu'on appelle un Fact checking system, c'est-à-dire un système qui va procéder de manière automatique, du moins en partie, à une ou plusieurs étapes de vérification d'un fait. Ce système permet d'extraire des faits d'une source de données et de leur appliquer un score qui détermine si un fait est plus ou moins vérifiable. ClaimBuster est utilisé pour faciliter le processus de vérification de fait, il a pour fonction d'aider le journaliste dans son travail. Cet outil va permettre au journaliste de se concentrer sur la vérification d'un fait plutôt que sur son identification et son extraction. Claimbuster est spécialisé dans la vérification de faits liés à la sphère politique.

Il se construit autour de différentes techniques d'apprentissage et de traitement de données : l'apprentissage automatique (machine learning), le traitement automatique du langage naturel, etc. Il est aussi capable, sur certains faits, d'appliquer automatiquement un état qui définit si un fait est plus ou moins vrai. Dans certains cas, ce système est entièrement autonome. Pour ce type de scénario, ClaimBuster dispose de 2 méthodes pour vérifier un fait.

Soit il va questionner des sources de données de faits déjà vérifiés et dans ce cas il ne se présente que comme une interface entre le fait et sa vérification. Dans le second cas il est capable de produire un verdict en traduisant le fait en question. Il requête ensuite des systèmes de type question-réponses (exemple : Google), compare et confronte les résultats pour établir son verdict.

**ClaimMonitor** C'est le module qui recherche et récupère les données. Au travers de différentes sources de données il va rechercher des textes à analyser. Ce module travaille sur 3 différentes sources de données. Il y a tout

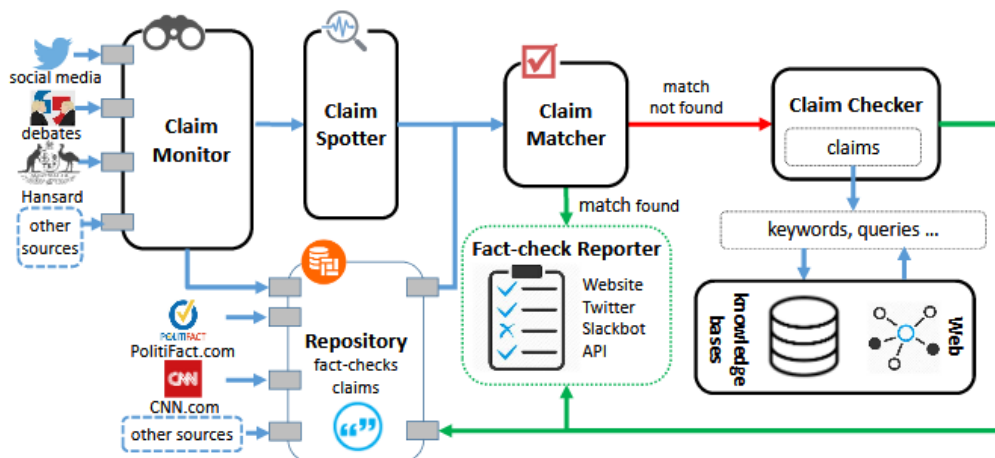


Figure 1: System architecture of ClaimBuster.

Figure 2: Architecture de ClaimBuster

d’abord la nécessité de travailler sur des informations en live afin de pouvoir instantanément évaluer les propos d’une personne : les débats politiques par exemple. C’est pour cela que les médias audiovisuels sont une importante source de données. ClaimBuster utilise différentes api pour traduire ce flux en données utilisables, notamment la structuration des données au travers de l’identification de l’orateur [16]. Ensuite le Claim Monitor travaille sur les données disponibles via les réseaux sociaux, il suit 2220 compte Twitter appartenant à différents type d’entité (personnalité politique, médias d’information, etc.). Seul les tweets relatifs à la politique sont gardés. Enfin ce module récupère diverses information issues de plusieurs sites web.

**ClaimSpotter** Une fois l’information captée et filtrée, on se retrouve avec des données liées à la politique et potentiellement constatables. Actuellement l’élément de ClaimBuster le plus mature est le Claim Spotter qui va attribuer un score à une assertion en fonction de si le fait définit est plus ou moins vérifiable : proche de 1, le fait peut être vérifié, proche de 0 le fait n’est pas vérifiable. Ce module a été construit autour d’un modèle définit par plusieurs dizaines de milliers de fait vérifiés manuellement. La précision de ce modèle est de plus de 80%. Le processus peut éventuellement s’arrêter à cette étape

si les prochaines étapes n'apportent pas d'informations concluantes. Dans ce cas on va simplement retourner à l'utilisateur une suite de faits définis comme vérifiables et leurs scores.

Le ClaimSpotter et l'identification des faits potentiellement vérifiable s'est fait au travers du machine learning et l'apprentissage supervisé (ici on travail sur des données vérifiées). Le set de données est composé de plusieurs dizaines de milliers de phrases prononcées durant des discours, débats, etc. politiques. Chaque phrase est identifiée comme étant plus ou moins importante et plus ou moins vérifiable. Plus précisément on différencie les :

- NFS (Non Factual Sense) : qui sont des phrases banales décrivant des opinions, des traits d'humour, etc.
- UFS (Unimportant Factual Sentence) : représente des phrase qui décrivent des faits qui n'ont que peu d'importance, ex : "Demain il va pleuvoir"
- CFS (Check-worthy Factual Sentence) : des phrases digne d'intérêts, qui valent la peine d'être vérifiées.

Afin de noter et classifier ces phrases, ClaimBuster va utiliser des Support Vector Machines (SVM), en français, séparateurs à vaste marge. Les SVM sont une classe d'algorithmes d'apprentissage supervisée destinés à la prévision de variable qualitative binaire, c'est-à-dire des problèmes de discrimination. Plus précisément, si l'on prend une phrase de type CFS, on va la traiter comme étant positive (les autres types étant négatifs). La SVM va ensuite trouver la marge optimale et tenter de trouver un classifieur permettant de généraliser et séparer les observations et ainsi leur attribuer un score distinctif.

**ClaimMatcher** A partir de cette étape, on ne travail que sur des faits définis comme vérifiables, on a éliminé les opinions et tout ce qui n'est pas une assertion. Ainsi étant donné une phrase qui énonce un fait, on va essayer de prouver ou non sa véracité. Le ClaimMatcher va tenter de trouver des faits déjà vérifiés (issus de plusieurs site de fact-checking) et en rapport avec l'assertion. Pour cela il utilise 2 méthodes, l'une consiste à simplement établir une concordance entre les mots utilisés pour qualifier le fait et ceux qui définissent les faits vérifiés. L'autre se base sur une l'analyse sémantique

des faits [20].

Si le fait n'est pas présent, alors le système va tenter de vérifier le fait automatiquement.

**ClaimChecker** Le rôle de ce module est de construire un argumentaire autour du fait en collectant des informations additionnelles issues de bases de connaissances. Pour ce faire il va interroger le système de recherche de Wolfram Alpha grâce à des questions générées automatiquement [15].

Par exemple si on a l'assertion suivante : "The largest country in the world is China", et que l'on recherche ceci sur Wolfram Alpha on n'obtient aucune donnée concluante. Mais si on entre ceci : "The largest country in the world", l'assertion qui va être interprétée comme une question par l'absence de cod, le résultat retourné est la Russie. Par confrontation, on peut voir qu'il existe une différence entre les résultats et donc le fait à des chances d'être faux. Faire du fact-checking entièrement automatisé sur des questions simples est donc faisable.

Ainsi ClaimChecker va potentiellement pouvoir arriver à un verdict s'il trouve des incohérences entre les différentes questions posées. Pour assurer une fiabilité plus poussée, on envoie les questions à Google et on analyse les premiers résultats trouvés toujours pour construire un argumentaire.

**Critique** Plusieurs approches ont été utilisées pour attribuer des scores de vérifiabilité aux faits : SVM mais aussi Naive Bayes Classifier (NBC) ou encore Random Forest Classifier (RFC). SVM a été retenu du fait du haut taux de réussite (96%, soit autant que les capacités d'analyse d'un être humain). Aucune ne se base sur le TAL, qui est ici défini comme loin d'être parfait. Mais le fait de construire son algorithme sur de l'apprentissage supervisé va rendre l'évolution de ClaimBuster bien plus compliqué. Si demain ClaimBuster souhaite étendre son activité de fact checking au-delà de la sphère politique vers tout type de domaine, ce sont des dizaines de datasets qu'il faudra construire. Même chose s'il veut s'ouvrir à d'autres langages. Dans cette approche, l'utilisation du machine learning contraint le système à avoir un seul domaine de prédilection : la politique. En outre sans TAL, ClaimBuster ne comprend pas la sémantique du fait qu'il vérifie, cela limite son champ d'action et ses possibilités d'évolution. Sans comprendre à quoi se rapporte le fait, à quoi il fait référence, il est compliqué de construire un

argumentaire fiable. Par exemple pour la phrase suivante : “The American Revolutionary War was a war fought between Great Britain and Russia”, ClaimBuster ne donne qu’un score de 0.32 soit un fait que l’on peut ignorer, qui n’est pas vérifiable ou ne vaut pas la peine d’être vérifié. Autre exemple : “Hayao miyazaki is dead.” ne donne qu’un score de 0.22. Mais ici ce sont des faits vérifiables et qui méritent d’être vérifiés. Pourtant Claimbuster les classe presque sur un pied d’égalité avec l’assertion suivante : “I love war” qui obtient un score de 0.17.

Autre point, certaines étapes nécessaires à la vérification d’un fait semblent peu adéquates ou peu fiables. Le ClaimChecker, afin d’étoffer l’argumentaire, va rechercher des informations sur les premières pages de google. Cette étape est assez floue mais est-ce que simplement sortir des phrases de Google sans vérifier les sources ce n’est pas tomber dans le piège de la désinformation ? Comment construire un argumentaire efficace sans utiliser le TAL ? Cela est d’autant plus vrai lorsque l’on essaye de vérifier une information récente, qui change souvent ou qui est floue. Et ici encore, sans TAL impossible de dire si l’information à un sens et est corrélée au fait.

Ces critiques faites, il faut quand même rappeler que le fact checking automatique n’en reste qu’à ses débuts et ClaimBuster fait parti des pionniers. Trouver un mode viable est plus important que rechercher le graal en proposant directement un système parfait. Ici le but premier est d’aider le journaliste même si la finalité serait d’obtenir un système autonome. ClaimBuster se base sur les assertions de personnalités politiques. En se perfectionnant sur cette base il pourra peut être s’ouvrir à d’autres domaines. Avec le TAL, il est possible d’utiliser une approche qui sera peut être au début moins viable, en effet l’outil évoluera en fonction de l’évolution du TAL. Mais cet outil sera plus simple à ouvrir à tout type de domaines et langues. En outre, ce qui fait défaut à ClaimBuster est le manque de contexte sémantique dans lequel placer un fait. En analysant un fait, ClaimBuster est incapable de dire s’il correspond à une personne ou à tel type de domaine. C’est à ce problème que tente de répondre une autre technique de fact checking qui se base sur les knowledge graph.

TODO : étoffer/finir critique

TODO : Voir les cas d’essais concrets

## 2.3 Knowledge graph

Un knowledge graph (KG ou graphe de connaissance ou réseau de connaissance) est défini comme une base de connaissance organisé sous forme de graphe [8] [27]. Comme pour une base de connaissance, l'information est organisée à l'aide d'ontologies. De part l'intégration d'une sémantique de l'information, il est possible de dériver du savoir de l'information disponible.

Modéliser notre problème de fact checking sous la forme d'un graphe va nous permettre d'utiliser des propriétés et comportement définis dans la théorie des graphes. Les graphes vont nous permettre de comprendre un fait en analysant leur environnement. Cela va nous permettre de fournir ce qui manque à d'autres systèmes de fact checking comme ClaimBuster : un contexte sémantique. Il va complémentariser certaines approches utilisant le machine learning ou encore des approches qui se basent sur le TAL en leur apportant une image de l'existant, un environnement plus vaste. Un KG va permettre d'avoir un aperçu factuel et précis de l'existant. Il va nous apporter un système qui va se baser sur les relations entre les entités définies dans le fait et les confronter avec celles dans l'existant pour établir un verdict. Nous commencerons par voir une application basique, la plus simple possible, d'une approche de fact checking à base de KG. Puis nous verrons des méthodes qui étendent et améliorent cette approche.

A lire : <https://nordicapis.com/automated-fact-checking-the-holy-grail-of-political->

### 2.3.1 Wikipédia knowledge graph

Le travail suivant repose sur les recherches effectuées ici [3]. Nous tâcherons d'analyser et comprendre le travail effectué, ensuite nous évaluerons sa faisabilité de son couplage avec d'autres modules qui permettront de renforcer et améliorer la fiabilité d'un système de fact checking. Suite aux recherches effectuées dans ce papier il a été prouvé qu'utiliser un graphe non orienté avec l'approche présentée ci-dessous est la plus optimale.

Pour commencer notre démonstration nous allons partir d'une assertion simple : dans un graphe une entité est liée à une autre entité s'il existe un chemin ne dépassant pas  $x$  entités d'écart. Soit un fait en rapport avec deux entités, si ces deux entités ne sont pas liées ou si une des deux entités n'est

pas présente dans le graphe, alors il y a de fortes chances pour que ce fait soit faux. Pour rappel, comme pour une base de connaissance, dans un graphe de connaissance l'information est organisée sous la forme de triplet  $\langle \text{Sujet}, \text{Prédicat}, \text{Objet} \rangle$ . Entre deux entités on peut avoir plusieurs chemins de longueurs variables. Chaque chemin apporte des informations distinctes et plus ou moins précises sur la relation entre ces entités. Rechercher le chemin le plus court possible entre deux entités va nous permettre d'établir un lien de véracité sur lequel nous allons nous baser pour déterminer si un fait est vrai ou faux.

Par exemple pour l'assertion suivante : "Joseph Boyden wrote Three Day Road" on a 2 entités, l'écrivain et l'ouvrage. Pour arriver de l'un à l'autre on va avoir des chemins directs : l'objet livre va avoir une relation de type "auteur" qui va directement lier le livre et son auteur. On va donc attribuer une valeur forte à cette relation. Un autre chemin qui passera par des entités plus génériques se verra attribuer une importance plus faible. La recherche du chemin le plus court se résume à la recherche du chemin qui a le plus de sens dans ce contexte.

Dans cette approche il est important de bien définir l'impact et le rôle de la longueur du chemin entre deux entités. Soit  $G$  un graphe non orienté et  $G = (V, E)$  où  $V$  représente nos sommets, nos entités et  $E$  les relations entre ces entités. Pour déterminer l'existence d'un lien entre deux entités on va utiliser une opération mathématique dite "Fermeture transitive". Le calcul de la fermeture transitive va nous permettre de créer un graphe annexe où toutes les relations sont déjà déterminées (pré-traitement qui va nous permettre de trouver plus facilement s'il existe un lien entre 2 entités) [17].

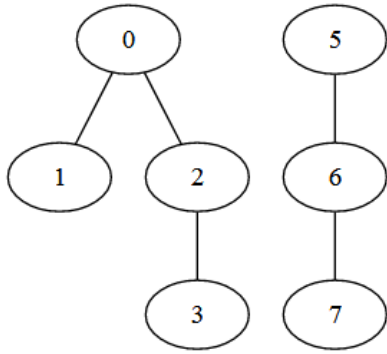


Figure 3: Graphe original, non orienté

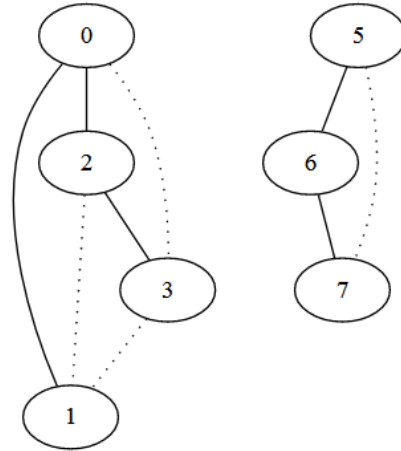


Figure 4: Graphe et sa fermeture transitive

On va attribuer à chaque chemin une valeur de confiance ou score de confiance qui va déterminer si ce chemin peut être utilisé pour évaluer un fait. Ce score va d'abord dépendre du degré (liens avec d'autres entités) des noeuds traversés. Une entité générique aura un degré important ce qui diminuera le score de confiance, ex : un pays, une grande organisation, etc. D'un autre côté ce score sera plus élevé s'il traverse des noeuds moins génériques, ex : personne, livre, etc. En effet un lien entre deux entités sera plus pertinent si le chemin ne traverse que des entités avec peu de relations et le contexte sémantique sera plus précis. En suivant la même logique lorsque deux entités sont directement liées, on va leur accorder un score de confiance maximal : il n'y a pas d'intermédiaires.

Le chemin qui aura le score le plus élevé sera le chemin le plus court et qui traversera les entités les moins générique. C'est ce que l'on va appeler la proximité sémantique. Cette proximité sémantique représente le sens qui relie deux entités. Pour un fait donné, est-ce que c'est deux entités mise bout à bout ont du sens ?

Pour le moment nous avons vu comment choisir un chemin entre deux entités, qui va nous permettre ensuite de déterminer si un fait est vrai ou faux. Pour illustrer ceci, prenons un exemple, soit des faits qui lient des villes avec



des pays. Pour chaque pays et chaque villes on va tester l’assertion : “La capitale du pays x est y”.

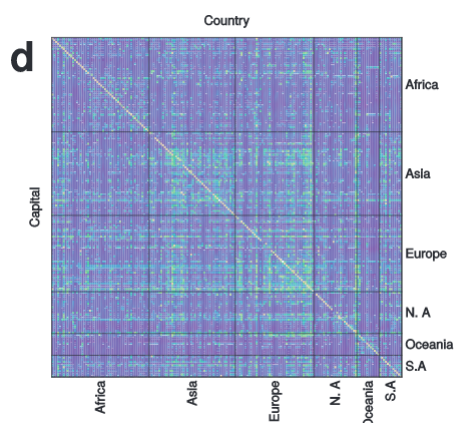


Figure 5: Pays et capitales groupées par continent

TODO : <http://www.kurzweilai.net/a-computational-algorithm-for-fact-checking>

Plus un point est clair et plus le fait a de chance d’être vrai. Le chemin entre un pays et sa capitale possède une proximité sémantique plus forte que le chemin entre un pays et n’importe quelle autre ville. Ainsi on peut voir que cette méthode fonctionne bien, le taux de réussite enregistré est de 95%. Cet exemple est simple mais prouve que le fact checking peut se faire au travers de graphes de connaissances.

**Critique** Cette méthode permet de faire du fact checking sur des faits simples. Elle se fait entre deux entités clairement définies. Le champs des possibles pour les faits à vérifier est donc limité.

Problème de scalabilité, important d’avoir le verdict instantanément.

Ciampaglia et al. [8] propose an approach that relies on a single short, specific path to differentiate a true fact from a false one. Although intuitive, their algorithm fails to account for the semantics of the target predicate

TODO : Ouvrir sur la seconde approche knowledge stream et montrer qu’il améliore cette approche

TODO : A lire : <https://searchengineland.com/google-researchers-introduce-system-rank-web-pages-facts-not-links-215835>

### 2.3.2 Finding Streams in Knowledge Graphs to Support Fact Checking

TODO : Attention les KG c'est pas vraiment de l'ia

Nous avons vu que le fact checking de faits simples était possible via le parcours d'un graphe de connaissance. Seulement plusieurs problèmes peuvent se poser comme la montée en charge d'une telle approche ou sa fiabilité. Dans ce papier, une nouvelle approche est utilisée pour organiser et parcourir un graphe de connaissance. Le code source est disponible [ici](#).

Cette approche consiste à organiser un graphe comme un réseau de flot [24]. Dans un réseau de flot, chaque arête possède une capacité maximale dans laquelle peut passer un flux. Avec cette approche on va chercher à évaluer sémantiquement chaque chemin allant d'une entité A vers une entité B. Plusieurs chemins vont apporter plusieurs contextes qui vont permettre d'évaluer sémantiquement le fait. En outre, la recherche de chemins optimaux est limitée par la capacité maximale de chaque arêtes.

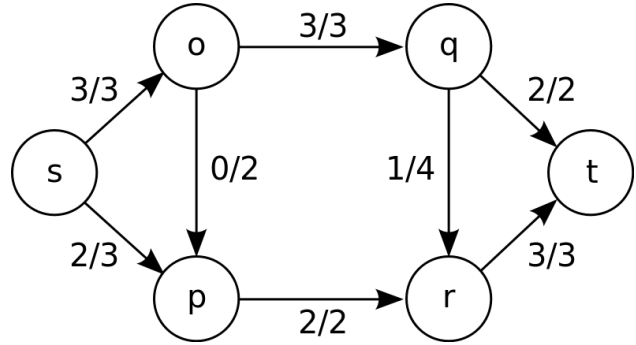


Figure 6: Graphe/Réseau de flot non orienté

En travaillant sur un réseau de flot et en contraignant ainsi chaque flux, on réduit le nombre de chemins possibles entre deux entités. De plus on limite le nombre d'arêtes communes entre chaque chemins. On a ainsi des chemins plus variés en terme de contexte sémantique. Dans la figure 6, pour aller de s vers t il est possible de passer par l'arête  $p \rightarrow r$ , sans contraintes, l'algorithme passera toujours par cette arête. L'espace restreint va l'obliger à passer par  $o \rightarrow q$  et ainsi fournir de nouvelles informations.

Ces chemins vont construire l'argumentaire sur lequel le système va s'appuyer pour rendre un verdict sur le statut d'un fait.

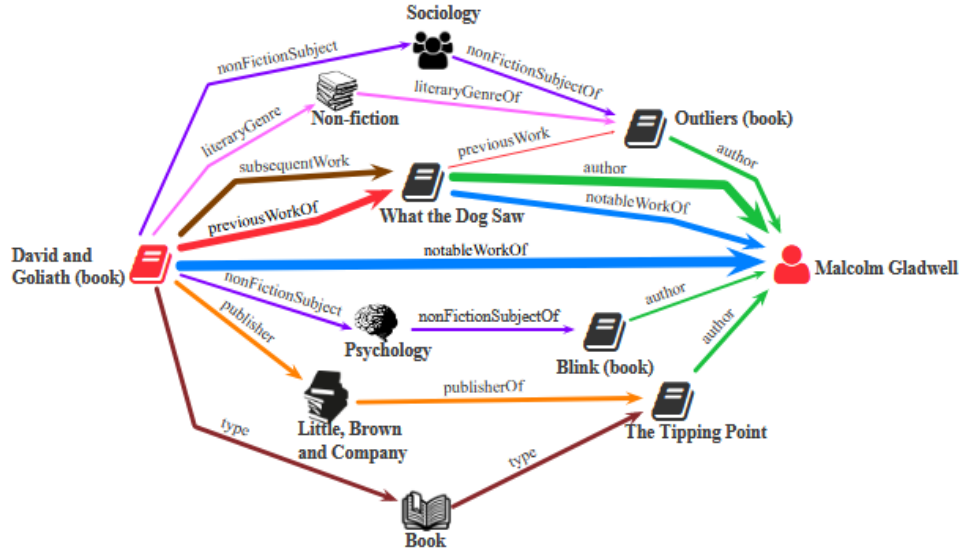


Figure 7: Relations entre un livre et son auteur dans un graphe de connaissance [24]

La figure 7 représente l'argumentaire autour du triplet  $t = (\text{David and Goliath (book)}, \text{author}, \text{Malcolm Gladwell})$ . Plus une arête est directe et plus elle est épaisse. Plus elle est épaisse et plus le flux d'information est important. Cela permet de privilégier les chemins courts dans lesquels les informations contenues sont sémantiquement importantes.

Pour un triple donné  $(s, p, o)$  il faut voir l'information comme un quantité abstraite qui doit voyager du sujet  $s$  vers l'objet  $o$ . Chaque arête possède une capacité permettant de transporter plus ou moins d'informations, mais aussi un coût d'utilisation. Ce coût d'utilisation représente le coût d'envoi d'une unité d'information sur le réseau. Ce coût d'utilisation permet encore de s'assurer que les chemins seront le plus court et spécifique possible.

Le but est donc de trouver les chemins pour lesquelles nous pouvons transporter le plus d'informations entre  $s$  et  $o$  pour un coût minimal.

## 2.4 Fact checking dans un réseau de flot

Nous allons voir à présent deux méthodes de fact checking qui appliquent ce concept de réseau de flot dans un graphe de connaissance : Knowledge Stream et Relational Knowledge Linker. Ces méthodes vont construire l'argumentaire d'un fait autour des relations entre les triples qui composent un chemin.

### 2.4.1 Knowledge Stream

**Analogie entre les relations** Afin de construire un contexte qui va être le plus proche possible de celui du fait énoncé, on va déterminer les relations homologues entre celle du fait et celles existante dans notre KG. Par exemple pour déterminer l'auteur d'un livre on ne va pas aller regarder les relations concernant l'architecture mais bien celles concernant la littérature.

Pour évaluer l'analogie entre les relations on va construire un graphe  $G$  dans lequel les sommets sont des entités et les arêtes des relations. Dans la figure 8, entre les entités  $A$  et  $C$  on a une relation de type  $c$  (qui peut être par exemple une relation de type "est marié à").

La figure 9 représente le graphe adjoint de  $G$ . Par définition, étant donné un graphe  $G$  non orienté, le graphe adjoint de  $G$ , noté  $L(G)$ , est un graphe qui représente la relation d'adjacence entre les arêtes de  $G$ . Ainsi chaque sommet de  $L(G)$  représente une arête de  $G$ . Deux sommets de  $L(G)$  sont adjacents si et seulement si les arêtes correspondantes partagent une extrémité commune dans  $G$  [35].

Dans la figure 9 on peut voir que certains noeuds sont représentés plusieurs fois comme  $a1$  et  $a2$  alors qu'ils définissent la même relation. Pour palier à ce problème il faut contracter le graphe comme dans la figure 10. On fusionne chaque sommet commun et on pondère les arêtes en fonctions des relations précédentes des sommets.

Cette démarche va nous permettre de définir les similarités entre les relations. Les arêtes pondérées dans la figure 10 représentent la fréquence à laquelle chaque relation co-incident avec ses voisins dans  $G$ . Plus cette pondération est importante et plus la similarité entre les relations est forte.

Au final, à partir de la matrice d'adjacence du graphe contracté, on va pouvoir établir des corrélations entre les différentes relations (ici les sommets)

à l'aide du TF-IDF (term frequency-inverse document frequency). Cette méthode va évaluer l'importance d'une relation au sein du graphe. Déterminer l'importance d'une relation va nous permettre d'établir un contexte plus précis pour un fait donné.

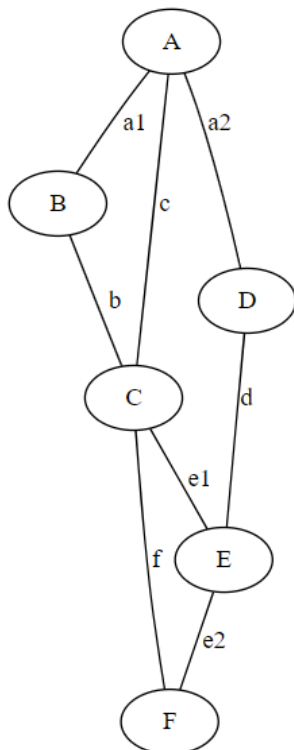


Figure 8: Graphe G original, non orienté

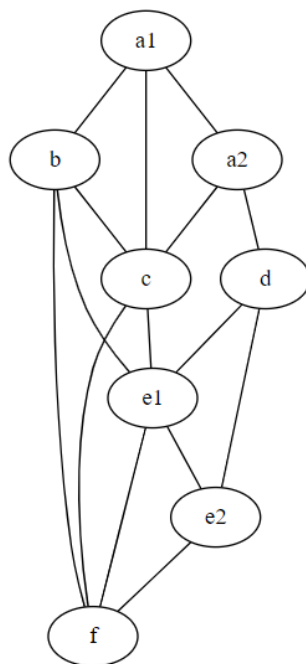


Figure 9: Graphe adjoint  $G'$

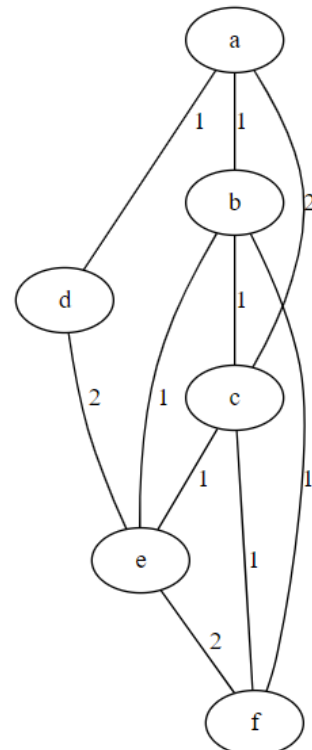


Figure 10: Graphe adjoint contracté  $G^*$

**Problème de minimisation dans un réseau de flot** Pour rappel nous cherchons à trouver l'ensemble des chemins pour lesquels le flux maximum est poussé de  $s$  vers  $t$  pour un coût minimal. Un réseau de flot vient avec différentes contraintes. Il faut tout d'abord pouvoir déterminer la pondération des relations entre les entités, savoir combien d'unités d'information je peux passer d'une entité à une autre. Pour cela on va se baser sur le calcul précédent qui détermine la similarité entre les relations. La quantité disponible

pour chaque flux sera calculé selon la similarité avec le prédicat du triplet cible. Plus deux relations seront similaires et plus le flux alloué sera important. En outre, plus un fait sera général (plus il aura de relations), moins son flux sera important.

Une autre contrainte énoncée plus haut est le coût d'utilisation du réseau, chaque chemin à un coût. On cherche à minimiser ce coût et dans cette approche minimiser le coût signifie traverser le moins d'entités génériques possible. Les entités trop génériques ne permettent pas de récupérer des informations précises dans un contexte donné. Le coût d'utilisation d'une arête sera donc déterminé par les relations de ses sommets avec les sommets adjacents.

La dernière contrainte est que dans un réseau de flot, le flux entrant dans un sommet doit être égal à son flux sortant. Il faut pouvoir déterminer le flot maximum qui peut être envoyé entre  $s$  et  $o$ .

Cette contrainte est représentée ainsi :

$$\sum_{P_{s,p,o} \in \mathcal{P}_{s,p,o}} \beta(P_{s,p,o}) \quad (1)$$

Où  $\beta(P_{s,p,o})$  représente le plus petit flot possible pour un chemin. Il s'agit de l'arête qui possède la capacité minimale sur un chemin. Les unités d'informations que je peux envoyer sur un chemin seront limitées par  $\beta(P_{s,p,o})$ . Dans cette approche, cette arête représente le triple le moins pertinent pour notre chemin.

Si on reprend la figure 6, on ne pourra jamais envoyer plus de 5 unités d'informations de  $s$  vers  $t$ . En effet les chemins qui mènent vers  $t$  sont limités. Si entre  $o$  et  $q$  mon flot maximum était de 2, alors je ne pourrais jamais envoyer plus de 4 unités d'informations de  $s$  vers  $t$ . Donc l'information maximale que je peux envoyer se calcule à partir de la somme des capacités minimales de chaque chemins allant de  $s$  vers  $t$ . Soit la somme des  $\beta(P_{s,p,o})$ .

Au final, sachant que je peux envoyer  $x$  unités d'information sur mon réseau, comment maximiser les informations viables que je peux récupérer ? Pour cela il faut pouvoir déterminer des chemins spécifiques pour éviter les chemins trop génériques. Ex : si mon chemin passe par l'entité "France" on va lui attribuer une faible valeur sémantique. On résout ce problème par la contrainte suivante :

$$\mathcal{S}(P_{s,p,o}) = \frac{1}{1 + \sum_{i=2}^{n-1} \log k(v_i)} \quad (2)$$

Où  $k(v_i)$  représente le degré du noeud  $v_i$ . On va établir un score sémantique déterminé par la somme des degrés des noeuds (nombre de relations) en relations avec  $P_{s,p,o}$ . Plus  $P_{s,p,o}$  aura de relations et plus il aura de relations génériques, moins son score sémantique sera élevé.

Ainsi nous allons pouvoir établir un score de fiabilité pour un triple :

$$\beta(P_{s,p,o}) \times \mathcal{S}(P_{s,p,o}) \quad (3)$$

Pour un triple donné, son score de fiabilité sera évalué par le produit entre le flot maximum que peut porter le chemin et le score sémantique du triple. Pour rappel, plus un triple est direct et plus son flot maximum est élevé. Un triple avec un score élevé sera court, le plus direct possible, et aura peu de relations avec des entités génériques. Finalement, pour évaluer le score de fiabilité d'un chemin, il suffit de sommer tous les scores de fiabilité des triples qui le compose.

#### 2.4.2 Relational Knowledge Linker

Cette approche cherche à étendre des méthodes de fact checking sur des graphes de connaissance déjà existantes. Pour l'approche précédente on se concentre sur la sémantique du sujet et de l'objet, on favorise le sens que l'on peut tirer du triple. L'argumentaire se construit autour de la spécificité de chaque triples.

Dans cette approche on va se concentrer sur la similarité entre les relations qui lient l'objet et le sujet pour construire notre argumentaire. On se base sur la sémantique du prédicat. Ainsi on va chercher à maximiser la sémantique autour des relations entre les triples. Cette approche étend ce que nous avons vu avec [3] en y ajoutant la similarité entre les relations que nous avons vu dans la section 2.4.1.

## 2.5 Critique

L'évaluation de cette approche s'est faite sur des datasets déjà formatés, des triples issus notamment du Google Relation Extraction Corpora (disponibles ici). Aucune application dans la vie réelle n'a été conduite. Pour des cas d'application réels il faut pouvoir vérifier un fait en langage naturel. Pour cela il faut pouvoir identifier dans un fait, et avec certitude, les deux entités et la relation qui les unie. On en revient au même problème de TAL. Comment je vais fournir à mon système des données qu'il peut traiter ? On peut envisager plusieurs méthodes pour palier à ce problème, les faits devant être simple, on peut utiliser le TAL ou le machine learning [10].

Pour rappel, cette approche permet de faire du fact checking sur des faits de la forme d'un triplet. Soit des faits simples. Et bien que les faits soient simples, le temps pour vérifier un seul fait se calcul selon une complexité pseudo-polynomiale. Soit  $O(Y|E|\log|V|)$  où  $V$  représente les sommets du graphe,  $E$  les arêtes et  $Y$  le flot maximum d'informations que l'on peut faire passer sur le réseau. Sur des graphes géants tel que DBPedia on obtient un temps moyen (sur laptop) de 356 secondes pour la vérification d'un fait. On peut s'interroger sur l'utilité d'une telle approche pour vérifier des faits en live ou même pour servir d'outil pour aider les journalistes. Avec cette approche la complexité va grandissante en fonction de la taille du graphe. Travailler sur des graphes complets, qui bénéficient de grandes quantités d'informations peut demander beaucoup de temps.

Ce temps de calcul est dû au caractère unique de chaque fait. Pour un fait a analyser il faut reproduire toutes les étapes vue précédemment. Il est impossible de réutiliser des données calculées en amont, sur d'autres faits, pour réduire le temps de calcul. En effet, pour chaque fait le graphe se modélise de façon unique. Pour chaque fait les entités et relations en cause ne seront pas les mêmes. Il est donc nécessaire de reconstruire le graphe.

Une autre limitation est la rareté de la donnée. Bien que les bases de connaissances existantes soient immenses, elles ne sont pas comparables au flux de données quotidien qui transite sur internet. De plus les bases de connaissances comme DBPedia ne contiennent aucune news ou informations récentes. Pour DBPedia on ne pourra vérifier un fait que si l'information est présente dans wikipédia. D'autres part, si ce manque de données venait à être résolu on se retrouverait avec des graphes tellement grands qu'il serait



impossible de les analyser dans des temps raisonnables avec les méthodes présentées.

## 2.6 Conclusion

Nous avons vu que les KG permettent de réaliser des opérations de fact checking sur des faits simples dans des temps plus ou moins raisonnables. Du fait du manque de données et de la complexité des faits à analyser il est difficile de penser à un système de fact checking entièrement autonome à base de KG. En revanche cette approche nous apporte des informations essentielles pour les autres méthodes et systèmes de fact checking : un contexte sémantique. Ce contexte sémantique se matérialise par la construction de l'argumentaire d'un fait qui évalue son environnement et confronte ses liens avec l'existant.

TODO : A lire : <http://blog.dbpedia.org/2018/05/15/the-dbpedia-databus-transforming->  
TODO : A voir : <https://arxiv.org/pdf/1503.00759v3.pdf>

### 2.6.1 Google knowledge graph

TODO : A voir : [https://fr.wikipedia.org/wiki/Knowledge\\_Graph](https://fr.wikipedia.org/wiki/Knowledge_Graph) : world fact book, etc.

TODO : <https://developers.google.com/knowledge-graph/> : permet de rechercher des infos sur des entités

Le Google Knowledge Graph (GKG) est ce qui permet au moteur de recherche de Google d'ajouter de la sémantique aux résultats d'une requête.

Il est possible de résumer la méthode précédente à une simple requête et à l'analyse syntaxique de la réponse. Reprenons l'assertion suivante : “Joseph Boyden wrote Three Day Road”. En annexe sont disponibles les fragments importants retournés par l'api (A.1).

## 2.7 Délégation aux apis

Faire du fact checking via différentes apis. (voir fib) <https://github.com/anandgoel/HackPrinceton>  
: exemple avec pas mal d'api

## 2.8 Falsification d'image

Ex : test sur donald trump qui tient la main du pape

<https://www.stopfake.org/en/13-online-tools-that-help-to-verify-the-authenticity-of-a-photo/>

<https://arxiv.org/pdf/1801.02768.pdf> : Fake Colorized Image Detection

Regarder la date d'existence de l'image sur google et la confronter avec la date de l'article.

TODO : <https://www.letemps.ch/sciences/twitter-mensonge-se-diffuse-plus-vite-plus-loin-verite> : Par exemple, «il y a plusieurs moyens de détecter si une image a été falsifiée», indique Ewa Kijak.

## 2.9 Vérification de la source du lien

Baser la réputation d'un site sur son contenu au fur et à mesure des vérifications (rapport entre fake news/vraies news/news non vérifiées)

## 2.10 Développement prototype

Prototype sur l'analyse de fausses images ? Construire un score de confiance.

<https://wit.ai/> : NLP (pas forcément utile mais intéressant) <https://github.com/anandgoel/HackP>  
: exemple avec pas mal d'api

### 2.10.1 Prédire la tromperie

[4] voir les citations de ce papier

### 2.10.2 A faire

TODO : Lire : <http://nlp.fast.ai/classification/2018/05/15/introducing-ulmfit.html>

TODO : Lire : <https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relati.html> et [https://github.com/google-research-datasets/relation-extraction-corpus/blob/master/20131104-date\\_of\\_birth.json](https://github.com/google-research-datasets/relation-extraction-corpus/blob/master/20131104-date_of_birth.json) TODO : credeye : [19]

- Parler de mon expérience dans l'analyse sémantique de texte

A rechercher :

- Fact checking api - Computational fact checking

Prendre le cas de FiB et expliquer pourquoi il est mauvais ?

## Bibliographie

- [1] Tim Berners-Lee. *Plenary at WWW Geneva 94*. URL: <https://www.w3.org/Talks/WWW94Tim/>.
- [2] Kurt Cagle. *Why the Semantic Web Has Failed*. URL: <https://www.linkedin.com/pulse/why-semantic-web-has-failed-kurt-cagle>.
- [3] Giovanni Luca Ciampaglia et al. “Computational fact checking from knowledge networks”. In: *PloS one* 10.6 (2015), e0128193. URL: <http://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0128193&type=printable>.
- [4] Niall J Conroy, Victoria L Rubin, and Yimin Chen. “Automatic deception detection: Methods for finding fake news”. In: *Proceedings of the Association for Information Science and Technology* 52.1 (2015), pp. 1–4. URL: <https://onlinelibrary.wiley.com/doi/epdf/10.1002/pra2.2015.145052010082>.
- [5] Michael Copeland. *What’s the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning ?* URL: <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>.
- [6] Stéphane Corlosquet et al. “Produce and Consume Linked Data with Drupal!” In: *International Semantic Web Conference*. Springer. 2009, pp. 763–778. URL: [https://link.springer.com/content/pdf/10.1007/978-3-642-04930-9\\_48.pdf](https://link.springer.com/content/pdf/10.1007/978-3-642-04930-9_48.pdf).
- [7] Xin Dong et al. “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*. ACM. 2014, pp. 601–610. URL: <https://www.cs.ubc.ca/~murphyk/Papers/kv-kdd14.pdf>.
- [8] Lisa Ehrlinger and Wolfram Wöß. “Towards a Definition of Knowledge Graphs.” In: *SEMANTiCS (Posters, Demos, SuCCESS)*. 2016. URL: <http://ceur-ws.org/Vol-1695/paper4.pdf>.
- [9] FOAF. *FOAF Vocabulary Specification 0.99*. URL: <http://xmlns.com/foaf/spec/>.

- [10] Google. *A Multilingual Corpus of Automatically Extracted Relations from Wikipedia*. URL: <https://ai.googleblog.com/2015/06/a-multilingual-corpus-of-automatically.html>.
- [11] Gulizar Hacıyakupoglu et al. “Countering Fake News: A Survey of Recent Global Initiatives”. In: (2018). URL: [https://think-asia.org/bitstream/handle/11540/8063/PR180307\\_Countering-Fake-News.pdf](https://think-asia.org/bitstream/handle/11540/8063/PR180307_Countering-Fake-News.pdf).
- [12] Steven Hansen. *How Big Data Is Empowering AI and Machine Learning ?* URL: <https://hackernoon.com/how-big-data-is-empowering-ai-and-machine-learning-4e93a1004c8f>.
- [13] Naeemul Hassan et al. “ClaimBuster: The First-ever End-to-end Fact-checking System”. In: *Proceedings of the VLDB Endowment* 10.7 (2017). URL: <https://john.cs.olemiss.edu/~nhassan/file/claimbuster-vldb17-hassan-demo.pdf>.
- [14] Naeemul Hassan et al. “The quest to automate fact-checking”. In: *world* (2015). URL: <http://ranger.uta.edu/~cli/pubs/2015/claimbuster-cj15-hassan.pdf>.
- [15] Michael Heilman and Noah A Smith. *Question generation via overgenerating transformations and ranking*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA LANGUAGE TECHNOLOGIES INST, 2009. URL: <http://www.cs.cmu.edu/~mheilman/papers/heilman-smith-qg-tech-report.pdf>.
- [16] Minumol Joseph et al. “Speaker identification in live events using Twitter”. PhD thesis. 2015. URL: <https://uta-ir.tdl.org/uta-ir/bitstream/handle/10106/25460/JOSEPH-THESIS-2015.pdf?sequence=1&isAllowed=y>.
- [17] Jean-Jacques Lévy. *Graphes*. URL: <http://pauillac.inria.fr/~levy/x/tc/polycopie-1.6/main009.html>.
- [18] Lucas J Myslinski. *Social media fact checking method and system*. US Patent 8,458,046. Apr. 2013. URL: <https://patentimages.storage.googleapis.com/ed/0b/b1/077ee47771206a/US8458046.pdf>.
- [19] Kashyap Popat et al. “CredEye: A Credibility Lens for Analyzing and Explaining Misinformation”. In: *CRF* 71.88.74 (2018), pp. 80–00. URL: <http://people.mpi-inf.mpg.de/~kpopat/publications/www18-demo.pdf>.

- [20] Vasile Rus et al. “Semilar: The semantic similarity toolkit”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2013, pp. 163–168. URL: <http://www.aclweb.org/anthology/P13-4028>.
- [21] schema.org. *Person*. URL: <http://schema.org/Person>.
- [22] schema.org. *schame.org*. URL: <http://schema.org>.
- [23] Baoxu Shi and Tim Weninger. “Discriminative predicate path mining for fact checking in knowledge graphs”. In: *Knowledge-Based Systems* 104 (2016), pp. 123–133. URL: <https://arxiv.org/pdf/1510.05911.pdf>.
- [24] Prashant Shiralkar et al. “Finding streams in knowledge graphs to support fact checking”. In: *arXiv preprint arXiv:1708.07239* (2017). URL: <https://arxiv.org/pdf/1708.07239.pdf>.
- [25] Clay Shirky. *Ontology is Overrated*. URL: [http://www.shirky.com/writings/ontology\\_overrated.html?goback=.gde\\_1838701\\_member\\_179729766](http://www.shirky.com/writings/ontology_overrated.html?goback=.gde_1838701_member_179729766).
- [26] Kai Shu et al. “Fake news detection on social media: A data mining perspective”. In: *ACM SIGKDD Explorations Newsletter* 19.1 (2017), pp. 22–36. URL: <https://arxiv.org/pdf/1708.01967.pdf>.
- [27] Jo Stichbury. *WTF is a knowledge graph?* URL: <https://hackernoon.com/wtf-is-a-knowledge-graph-a16603a1a25f>.
- [28] *The Open Graph protocol*. URL: <http://ogp.me/>.
- [29] *The State of Automated Factchecking - Full Fact*. [https://fullfact.org/media/uploads/full\\_fact-the\\_state\\_of\\_automated\\_factchecking\\_aug\\_2016.pdf](https://fullfact.org/media/uploads/full_fact-the_state_of_automated_factchecking_aug_2016.pdf). Accessed: 2018-04-18.
- [30] Soroush Vosoughi, Mostafa‘Neo’ Mohsenvand, and Deb Roy. “Rumor gauge: predicting the veracity of rumors on twitter”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11.4 (2017), p. 50. URL: <http://soroush.mit.edu/publications/a50-vosoughi.pdf>.
- [31] w3c. *RDF*. URL: <https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/>.
- [32] w3techs. *w3techs*. URL: <https://w3techs.com/>.

- [33] William Yang Wang. “" Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection”. In: *arXiv preprint arXiv:1705.00648* (2017). URL: <https://arxiv.org/pdf/1705.00648.pdf>.
- [34] Wikipédia. *Désinformation* — *Wikipédia, l'encyclopédie libre*. [En ligne; Page disponible le 18-mars-2018]. 2018. URL: <http://fr.wikipedia.org/w/index.php?title=D%C3%A9sinformation&oldid=146557137>.
- [35] Wikipédia. *Line graph* — *Wikipédia, l'encyclopédie libre*. [En ligne; Page disponible le 13-juillet-2016]. 2016. URL: [http://fr.wikipedia.org/w/index.php?title=Line\\_graph&oldid=127817300](http://fr.wikipedia.org/w/index.php?title=Line_graph&oldid=127817300).
- [36] Wikipédia. *Triplestore* — *Wikipédia, l'encyclopédie libre*. [En ligne; Page disponible le 20-décembre-2017]. 2017. URL: <http://fr.wikipedia.org/w/index.php?title=Triplestore&oldid=143705559>.

# Appendices

## A Knowledge Graph

### A.1 Fragment du résultat retourné par le google knowledge graph pour l'entité "Joseph Boyden"

```
{
  "@context": {
    "@vocab": "http://schema.org/",
    "goog": "http://schema.googleapis.com/",
    "EntitySearchResult": "goog:EntitySearchResult",
    "detailedDescription": "goog:detailedDescription",
    "resultScore": "goog:resultScore",
    "kg": "http://g.co/kg"
  },
  "@type": "ItemList",
  "itemListElement": [
    {
      "@type": "EntitySearchResult",
      "result": {
        "@id": "kg:/m/08gyww",
        "name": "Joseph Boyden",
        "@type": [
          "Thing",
          "Person"
        ],
        "description": "Canadian novelist",
        "image": {
          "contentUrl": "http://t2.gstatic.com/images?q=tbn:ANd9GcRXyYM8YrpcumM2NqJZIL5WINncJYOgUIO-w93ztUOdOJiPK0rV",
          "url": "https://en.wikipedia.org/wiki/Joseph_Boyden"
        },
        "detailedDescription": {
          "articleBody": "Joseph Boyden CM is a Canadian novelist and short
```



```

    story writer. His first novel, Three Day Road, won the
    Amazon/Books in Canada First Novel Award and the Rogers Writers'
    Trust Fiction Prize. ",
    "url": "https://en.wikipedia.org/wiki/Joseph_Boyden",
    "license": "https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License"
  }
},
"resultScore": 428.041321
},
{
  "@type": "EntitySearchResult",
  "result": {
    "@id": "kg:/m/0cwbmj",
    "name": "Three Day Road",
    "@type": [
      "Book",
      "Thing"
    ],
    "description": "Novel by Joseph Boyden",
    "detailedDescription": {
      "articleBody": "Three Day Road is the first novel from
      Canadian writer Joseph Boyden.",
      "url": "https://en.wikipedia.org/wiki/Three_Day_Road",
      "license": "https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License"
    }
  },
  "resultScore": 13.805716
},
{
  "@type": "EntitySearchResult",
  "result": {
    "@id": "kg:/g/11bw4txn3g",
    "name": "Al Purdy Was Here",
    "@type": [
      "Movie",
      "Thing"
    ]
  }
}

```

```

    ],
    "description": "2015 film",
    "url": "http://alpurdywashere.com/"
  },
  "resultScore": 13.146592
}

```

## A.2 Fragment du résultat retourné par le google knowledge graph pour l'entité “Three day road”

```

{
  "@context": {
    "@vocab": "http://schema.org/",
    "goog": "http://schema.googleapis.com/",
    "EntitySearchResult": "goog:EntitySearchResult",
    "detailedDescription": "goog:detailedDescription",
    "resultScore": "goog:resultScore",
    "kg": "http://g.co/kg"
  },
  "@type": "ItemList",
  "itemListElement": [
    {
      "@type": "EntitySearchResult",
      "result": {
        "@id": "kg:/m/0cwbmj",
        "name": "Three Day Road",
        "@type": [
          "Book",
          "Thing"
        ],
        "description": "Novel by Joseph Boyden",
        "detailedDescription": {
          "articleBody": "Three Day Road is the first novel from Canadian writer Joseph Boyden. Joseph's maternal grandfather, as well as an uncle on his father's side, served as soldiers during the First World War, and Boyden draws upon a wealth of family narratives.",

```

```
    "url": "https://en.wikipedia.org/wiki/Three_Day_Road",
    "license": "https://en.wikipedia.org/wiki/Wikipedia:
Text_of_Creative_Commons_Attribution-ShareAlike_3.0
_Unported_License"
  }
},
"resultScore": 486.23822
}
```