

UNIVERSITE DE MONTPELLIER
RAPPORT DE PROJET

Claims checking

Belkassim BOUZIDI
Chakib ELHOUITI
Massili KEZZOUL
Abdelkader NEDJARI
Ramzi ZEROUAL

Encadrant :
M^r Konstantin TODOROV



UNIVERSITÉ
DE MONTPELLIER



10 mai 2020

Remerciements

Tout d'abord nous souhaitons adresser nos remerciements au corps professoral et administratif de la faculté des sciences de Montpellier qui déploient des efforts pour assurer à leurs étudiants une formation actualisée.

En second lieu, nous tenons à remercier notre encadrant M^r Konstantin TODOROV pour ses précieux conseils et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre projet en acceptant d'examiner notre travail.

Nous remercions M^r Yahia Zeroual pour sa relecture attentive de ce rapport.

Table des matières

1	Organisation du projet	3
1.1	Méthodes d'organisation	3
1.2	Découpage du projet	3
1.2.1	Phase de modélisation	3
1.2.2	Phase de développement	3
1.2.3	Finalisation du projet	3
1.3	Outils de collaboration	4
2	Introduction au sujet	5
2.1	Fact-checking	5
2.1.1	Présentation du principe de fact-checking	5
2.1.2	Présentation de ClaimsKG	5
2.1.3	Travail à réaliser	6
2.2	Technologies utilisées	6
3	Conception et implémentation du projet	7
3.1	Conception et Modélisation	7
3.1.1	Analyse de ClaimKG	7
3.1.2	Conception	9
3.2	Implémentation	10
4	Analyse des résultats	12
4.1	Résultats	12
4.2	Problèmes rencontrés	13
5	Bilan et conclusions	14
5.1	Ce qu'on a fait et pas fait	14
5.2	Perspective	14
A	Annexe	15

Chapitre 1

Organisation du projet

1.1 Méthodes d'organisation

Afin de mener à bien le développement du projet, nous avons décidé de travailler un maximum de temps ensemble et de manière très régulière. Nous nous sommes réunis trois à quatre fois par semaine, en vue de faire le point sur l'avancement du projet et de définir les objectifs restants à atteindre.

Ainsi, selon l'état de progression de la conception, nous réalisions les tâches en retard durant le week-end pour ne pas cumuler de retard et respecter l'intégralité du cahier de charges.

Toutes les semaines, nous nous sommes réunis avec notre encadrant, M^r Konstantin TODOROV. Lors de ces réunions de mises au point relatifs au projet, nous furent prodigués, cela nous a permis de bénéficier de précieux conseils.

1.2 Découpage du projet

Nous avons découpé la réalisation du projet en trois grandes phases :

1.2.1 Phase de modélisation

Durant cette étape, nous nous sommes réunis pour définir les fonctionnalités demandées par le projet. Notamment séparer les fonctionnalités importantes de celles moins importantes. Nous avons également choisi les outils de travail collaboratifs et les principales technologies utilisées, ainsi qu'une première modélisation du projet.

1.2.2 Phase de développement

Durant cette phase, nous avons commencé à implémenter les différentes fonctionnalités que nous avons modélisées lors de l'étape précédente, tout en améliorant la modélisation au fur et à mesure de l'avancement de notre projet. Nous avons notamment réalisé des tests pour les différents modules afin de s'assurer de leur bon fonctionnement.

1.2.3 Finalisation du projet

Cette étape a consisté en la réalisation des tests finaux afin de s'assurer que les différents scripts fonctionnent en toute circonstance et éventuellement corriger les bogues qui peuvent apparaître.

1.3 Outils de collaboration

Afin de s'organiser, nous avons décidé d'utiliser Git à travers le serveur GitLab hébergé par le service informatique de la faculté. En effet le logiciel libre Git a facilité grandement la collaboration entre nous. Le serveur GitLab quant à lui est fourni gratuitement par le service informatique de la faculté.

En ce qui concerne la rédaction de ce rapport, nous avons utilisé \LaTeX , système de composition de documents créé par Leslie Lamport, pour faciliter la rédaction à plusieurs.



Schéma 1.1 – Logo du GitLab



Schéma 1.2 – Logo de Latex

Chapitre 2

Introduction au sujet

2.1 Fact-checking

2.1.1 Présentation du principe de fact-checking

Le fact-checking ou la vérification des faits, est une technique consistant à vérifier la véracité des faits et l'exactitude des chiffres présentés dans les médias, les différents réseaux sociaux, les blogs, etc... Cette notion est apparue aux États-Unis dans les années 1990. Elle a été mise en pratique par des journalistes d'investigation dans le cadre de leur profession, la méthode s'est démocratisée grâce à des logiciels aidant les particuliers à vérifier les faits.

L'entrée officielle du fact-checking en France date de 1995, quand est créée l'association Acrimed, qui se présente comme « l'observatoire des médias ».

En vue de la prolifération très rapide des informations, il devient de plus en plus important de s'assurer de la véracité des informations qui se trouvent partout sur Internet et autres médias. Tant du point de vue de la société que de celui de la recherche. De nombreuses approches récentes dans diverses communautés scientifiques portent sur des problèmes tels que la vérification des faits, la détection de la pertinence ou de point de vue des documents par rapport à des revendications particulières.

2.1.2 Présentation de ClaimsKG

Le LIRMM¹ en collaboration avec 2 équipes allemandes (L3S Hannover et l'institut de sciences sociologiques GESIS à Cologne) a construit et mis à disposition la base de connaissance ClaimsKG² qui regroupe les informations et méta-données provenant d'un grand nombre de sites journalistiques internationaux de fact checking, tels que Politifact³ ou Snopes⁴. ClaimsKG est un graphe de connaissances d'assertions annotées et liées qui facilite la création de requêtes structurées sur les assertions, leurs valeurs de vérité (True, Mostly False, etc...), leurs auteurs, date de publication, etc... ClaimsKG est généré par un pipeline entièrement automatisé qui collecte des assertions et des métadonnées à partir des sites de fact-checking, il transforme les données en graphes de connaissances selon un modèle établi, et annote les assertions avec des entités DBpedia (Wikipedia). La base actuelle comprend plus de 32 000 assertions publiées depuis 1996 et est mise à jour régulièrement.

1. Le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

2. <https://github.com/claimskg>

3. <https://www.politifact.com/>

4. <https://www.snopes.com/>

2.1.3 Travail à réaliser

Le sujet du TER consiste en l'enrichissement de cette base de connaissances avec des nouvelles données provenant des sites web suivants :

Fatabyyano ⁵ est un site jordanien en arabe. Fatabyyano (terme en arabe qui veut dire "Alors montrez-le") est la première et la seule plateforme arabe certifiée par l'IFCN ⁶ ;

Vishvas.news ⁷ est un site Internet de vérification des faits multilingue qui s'engage à combattre la désinformation et les informations erronées.

Le but du TER sera d'identifier les assertions individuelles dans chaque histoire, ainsi que leur label de véracité et par la suite identifier les relations entre elles, ainsi que les relations entre ces histoires. Les données produites par ce projet seront intégrées à la base de connaissance ClaimsKG.

La principale difficulté qu'on s'attend à rencontrer est la gestion des différentes langues proposée par ces sites web. Notamment dans l'identification des différentes relations entre les assertions. En effet afin de reconnaître les différents mots-clés du sujet traité par les articles, on utilise habituellement TAGME, un puissant outil de reconnaissance d'entités nommées dans un texte. Les langues utilisées par ces sites web ne sont pas prises en charge par cet outil. Il s'agit donc de trouver une alternative afin de reconnaître ses différentes assertions. Ce problème ainsi que sa résolution seront détaillés plus tard dans ce rapport.

2.2 Technologies utilisées

Nous avons implémenter l'application en Python, choix qui s'impose de lui même car couplé à la bibliothèque BeautifulSoup, il devient très facile de faire du web scraping ⁸.

En ce qui concerne la traduction, nous avons choisi d'utiliser Yandex Translator ⁹. En effet en vue de la taille des données à traduire, nous n'avons pas pu utiliser l'API de traduction de Google puisque cette dernière devient payante à partir d'un certain nombre de caractères.

Enfin, la reconnaissance des différents entités concernées est faite grâce à l'outil TAGME ¹⁰ a travers son API python ¹¹.



Schéma 2.1 – Logo de Python



Schéma 2.2 – Logo de BeautifulSoup



Schéma 2.3 – Logo de Yandex

5. <https://fatabyyano.net/>

6. International Fact-Checking Network : <https://www.poynter.org/ifcn/>

7. <https://www.vishvasnews.com/>

8. Le web scraping est une technique d'extraction du contenu de sites Web, via un script ou un programme

9. <https://translate.yandex.com/>

10. <https://tagme.d4science.org/tagme/>

11. Lien Github vers l'api Python de TAGME : <https://github.com/marcocor/tagme-python>

Chapitre 3

Conception et implémentation du projet

3.1 Conception et Modélisation

3.1.1 Analyse de ClaimKG

Lors de cette première phase, le plus important a été de comprendre et s’imprégner du code et de la structure déjà mis en place et mis à notre disposition (Claimskg), comprendre les outils utilisés ainsi que la structure déjà établie.

La structure

Comme le projet est d’une envergure immense, bien comprendre la structure était primordiale afin de respecter un maximum les outils utilisés. Notre encadrant nous a mis à disposition un écrit (que vous trouverez dans le dossier autre/ sous le nom de Claimskg.pdf) qui explique la structure globale du projet dont voilà ci-dessous un résumé.



Schéma 3.1 – ClaimKG

Comme vous pouvez le voir sur le schéma, Claimskg est découpé en plusieurs phases. La première consiste à extraire les données brutes à partir de site web de fact-checking et ensuite les structurer afin de produire un fichier CSV. La deuxième phase quant à elle, consiste à faire passer ce fichier CSV dans un générateur de graphes de connaissances¹. Dans notre cas, on s’occupe de la première phase, c’est-à-dire, extraire les données brutes du site web afin de produire le fichier CSV.

Le fichier CSV en question contient l’ensemble des faits traités par un site web. Chaque fait est passé à un script de scraping afin d’extraire chaque information et la structurer. La structure d’un fait est réalisée selon un modèle précis dont voici une partie :

rating value : La valeur de véracité du fait ;

1. Knowledge Graph Generator

3.1.2 Conception

Dans cette phase, nous allons présenter les étapes de conception que nous avons suivies pour bien implémenter notre application. La première étape a été de concevoir quelques diagrammes UML utiles à la bonne compréhension du projet.

Diagrammes UML

Lors de la phase de la modélisation, nous avons vu qu'il était primordial de concevoir un diagramme de classe pour connaître la manière dont les classes interagissent. Comme la figure suivante le montre, des classes ont été rajoutées par rapport aux classes déjà développées.

Dans les sous-classes de FactCheckingSiteExtractor, la méthode "extract_claim_and_review" a été implémentée pour transformer une page donnée en paramètre en une instance de claim. La méthode "get_all_claims" quant à elle, extrait les liens des articles du site web. Les données, une fois extraites seront stockées dans un fichier CSV où chaque ligne représente un article et chaque colonne les informations le concernant.

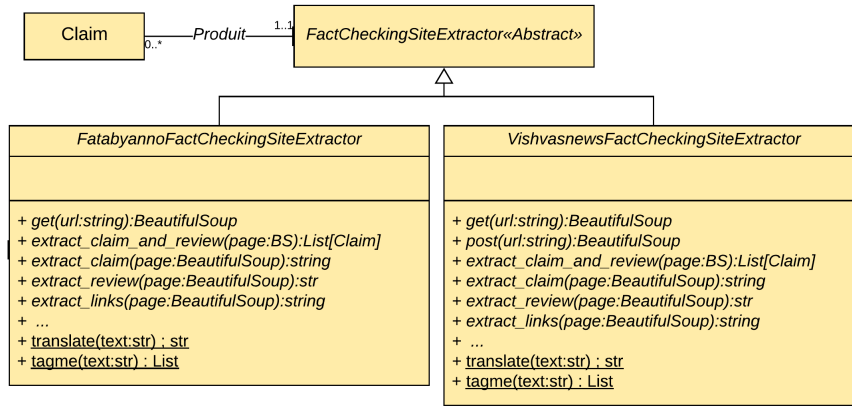


Schéma 3.4 – Structuration de claimsKG-extractor

Le déroulement de l'extraction des données a été quant à elle modélisée par un diagramme de séquence. La figure ci-dessous représente le fonctionnement de l'extraction du site Fatabayano.

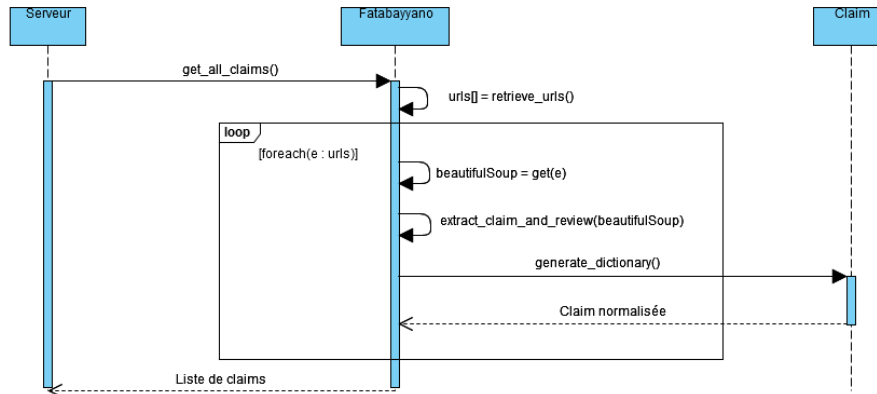


Schéma 3.5 – Structuration de claimsKG-extractor

Représentation des données

La deuxième étape consistait à trouver un site web respectant les critères de fact-checking et qui nous permet d'extraire les données essentielles (la véracité, les claims écrites, un auteur, etc...) au noyau du projet (ClaimsKG). Pour apporter un maximum de diversité ainsi que notre contribution personnelle, nous avons décidé, avec l'accord de notre encadrant, d'extraire les données d'un site d'une langue qui n'est pas encore présente dans le projet. Vu que tous les étudiants de ce projet sont arabophones, la langue arabe était un choix logique. Le seul site officiel de fact-checking en arabe est : <https://fatabyyano.net/>.

Nous avons donc analysé la structure du site web puis commencé son implémentation. L'étape suivante fût de trouver comment faire une "named entity recognition" sur les mêmes principes que celles déjà présentées (TagMe qui renvoie vers des liens wikipedia).

Le second site web sur lequel nous avons travaillé est <https://www.vishvasnews.com/>, il regroupe 11 langues différentes : L'anglais, Hindi, Pendjabi, Ourdou, Bengali, Tamoul, Malayalam, Goudjerati, Télougou, Marathi et L'odia , nous avons analysé la structure du site puis extrait ses données.

3.2 Implémentation

Tout comme la partie conception, nous avons décidé de répartir l'implémentation du projet en X parties

Nous avons commencé par implémenter les classes que nous avons modélisées dans la partie conception, puis nous nous sommes concentrés sur la manière d'extraire les liens de chaque article. Pour cela, nous avons tout d'abord cherché dans le site où et comment les liens sont présentés en inspectant les pages web ainsi que leur code source. Une fois les liens trouvés et après avoir bien cerné la structure du site web, nous avons commencé à écrire la méthode « `get(url:string)` » qui permet, à partir d'un lien URL de retourner une instance de la classe « BeautifulSoup ». Cette classe permet de garder en mémoire le code source d'une page web de façon structurée, et qui permet avec ses méthodes, de recherche et de manipulation. Elle facilite aussi grandement l'extraction des données. Une fois cette méthode achevée, on a pu commencer à écrire la méthode « `retrieve_url(page:BeautifulSoup)` » qui recupere l'ensemble des liens des articles.

L'ensemble des liens maintenant récupéré, nous avons implémenté les méthodes d'extraction - « `extract_claim(page:BeautifulSoup)` », « `extract_date(page:BeautifulSoup)` »...etc - en analysant la position de chaque information une à une dans la structure du code source.

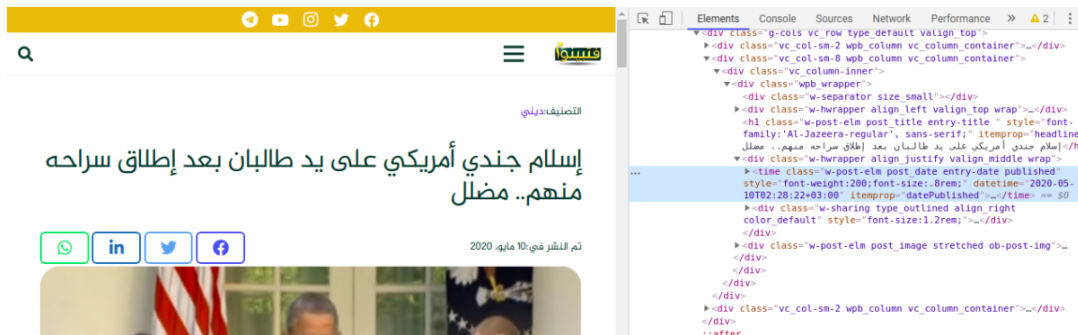


Schéma 3.6 – Extraction de la date du site web Fatabyyano

Ensuite une fois les données extraite du site, nous avons réfléchi à un moyen d'extraire les entités présentes dans la claim et dans la review de l'article. Pour se faire, nous avons cherché une alternative à TAGME pour la langue arabe (pour le site web Fatabyyano), après plusieurs recherches et tests, nous n'avons trouvé aucune solution plausible pour le remplacer. Nous avons opté pour une traduction de l'arabe vers l'anglais en utilisant une API de Yandex, pour ensuite



Schéma 3.7 – Extraction de la véracité du site web Vishvas

pouvoir utiliser TAGME sur l'article traduit.

Pour finir, après avoir implémenté un script en utilisant notre conception, nous avons pu générer une première version du fichier CSV. par la suite, nous nous sommes retrouvés dans l'obligation de le modifier pour qu'il soit en adéquation avec l'architecture du fichier CSV du projet principal (ClaimsKG).

Chapitre 4

Analyse des résultats

4.1 Résultats

Comme premier résultat du site fatabayyano nous avons obtenu un fichier csv contenant l'url de la claim, le rating, la traduction la claim, la date, le titre, tous les liens présents sur la page et les tags ; le tout en langue arabe.

Url	rating	rating_transl	claim	date	tags
https://fatabayyano.com/claim/1	صحيح	TRUE	في ولاية راخين	2019-07-25	"البشرطة,بورما"
https://fatabayyano.com/claim/2	صحيح	TRUE	في رجل مسلم	2019-07-25	"باكات,بورمين"
https://fatabayyano.com/claim/3	صحيح	TRUE	راكان في بورما	2019-07-25	"ن,بورما,تعذيب"
https://fatabayyano.com/claim/4	صحيح	TRUE	تين غرب بورما	2019-07-25	"اتدمير,تكتشف"
https://fatabayyano.com/claim/5	صحيح	TRUE	لهاداً في العالم	2019-07-25	"يادار,الروهنجيا"
https://fatabayyano.com/claim/6	صحيح	TRUE	عنوك عن العمل	2019-07-03	"وم,علوم,طلي"
https://fatabayyano.com/claim/7	صحيح	TRUE	نيقة أم إشاعة؟	2019-05-15	"وم,علوم,طلي"
https://fatabayyano.com/claim/8	صحيح	TRUE	يقفة أم مفيرك؟	2019-05-06	"وم,علوم,طلي"
https://fatabayyano.com/claim/9	صحيح	TRUE	لثانية في 2019	2019-04-14	"وم,علوم,طلي"
https://fatabayyano.com/claim/10	صحيح	TRUE	أبوطلي و دى	2019-04-03	"أخبار,حقيقية"
https://fatabayyano.com/claim/11	صحيح	TRUE	وسط كونهاغن	2019-03-26	"ة,دنماركون#"
https://fatabayyano.com/claim/12	صحيح	TRUE	خدمى فيسبوك	2019-03-23	"وم,علوم,طلي"
https://fatabayyano.com/claim/13	صحيح	TRUE	خلال بث مباشر	2019-03-16	"Fraser_Annii"
https://fatabayyano.com/claim/14	صحيح	TRUE	تقرير فنيبوا	2019-03-15	"newzealand,"
https://fatabayyano.com/claim/15	صحيح	TRUE	مل حول العالم	2019-03-13	"وم,علوم,طلي"
https://fatabayyano.com/claim/16	صحيح	TRUE	لادش المجاورة	2019-02-27	"خلادش,حقيقة"
https://fatabayyano.com/claim/17	صحيح	TRUE	قلب ولا نبض؟	2019-02-23	"وم,علوم,طلي"
https://fatabayyano.com/claim/18	صحيح	TRUE	دين الإسلامى؟	2019-02-15	"مات,Amazon"
https://fatabayyano.com/claim/19	صحيح	TRUE	يقفة أم إشاعة؟	2019-02-06	"هلولندا,Islam,"
https://fatabayyano.com/claim/20	صحيح	TRUE	ر ضد الإمارات؟	2019-01-11	"خطيرة_حيوان"
https://fatabayyano.com/claim/21	صحيح	TRUE	ي سماء كونز؟	2018-12-29	"ن,صوء,#,أزرق"
https://fatabayyano.com/claim/22	صحيح	TRUE	يعيون حمراء؟	2018-09-01	"وم,علوم,طلي"
https://fatabayyano.com/claim/23	صحيح	TRUE	طاولة الجراة؟	2018-08-31	"وم,علوم,طلي"
https://fatabayyano.com/claim/24	صحيح	TRUE	معدنى منصهر؟	2018-08-19	"وم,علوم,طلي"
https://fatabayyano.com/claim/25	صحيح	TRUE	ن الحرم المكى	2018-06-09	"وم,علوم,طلي"
https://fatabayyano.com/claim/26	صحيح	TRUE	بداخل شجرة؟	2018-05-11	"وم,علوم,طلي"
https://fatabayyano.com/claim/27	صحيح	TRUE	نيقة أم إشاعة؟	2018-03-01	"وم,علوم,طلي"

Schéma 4.1 – premier fichier csv obtenu

Puis après avoir appliqué le processus de traduction sur la claim et la review et après obtention des entités présentes dedans, deux autres colonnes sont ajoutées contenant les résultats du tagme du projet initial, ces résultats sont les noms attribués par la base de données wikipedia de cette entité.

Les résultats du second site web "vishvasnews" sont aussi regroupés dans un fichier CSV, qui est généré par le programme du projet initial compte tenu du fait qu'une des langues extraites est l'anglais. Les colonnes contiennent les mêmes informations que le site précédent. Le tout est en anglais, Hindi, Pendjabi, Ourdou, Bengali, Tamoul, Malayalam, Goudjerati, Télougou, Marathi puis L'odia puis par catégorie.

claim_url	Claim	title	claim_authn	links	date	tags	authors	rating_value	claim_entity	review_entity
https://fataby.com/claim/149109	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/149109	https://www.2020-03-25	فاتبينو	صحيح	[{"id": "783925", "id": "589774"}]			
https://fataby.com/claim/103410	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/103410	https://www.2020-03-18	فاتبينو	صحيح	[{"id": "149109", "id": "82138"}]			
https://fataby.com/claim/645068	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/645068	https://www.2020-03-17	فاتبينو	صحيح	[{"id": "103410", "id": "322141"}]			
https://fataby.com/claim/165938	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/165938	https://www.2020-03-06	فاتبينو	صحيح	[{"id": "645068", "id": "645068"}]			
https://fataby.com/claim/215107	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/215107	https://www.2020-02-24	فاتبينو	صحيح	[{"id": "165938", "id": "201483"}]			
https://fataby.com/claim/371324	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/371324	https://www.2020-02-01	فاتبينو	صحيح	[{"id": "215107", "id": "247154"}]			
https://fataby.com/claim/228525	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/228525	https://www.2020-01-24	فاتبينو	صحيح	[{"id": "371324", "id": "371324"}]			
https://fataby.com/claim/208277	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/208277	https://www.2020-01-10	فاتبينو	صحيح	[{"id": "228525", "id": "190758"}]			
https://fataby.com/claim/396275	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/396275	https://www.2020-01-08	فاتبينو	صحيح	[{"id": "208277", "id": "561065"}]			
https://fataby.com/claim/783925	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/783925	https://www.2020-03-25	فاتبينو	صحيح	[{"id": "396275", "id": "468926"}]			
https://fataby.com/claim/149109	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/149109	https://www.2020-03-18	فاتبينو	صحيح	[{"id": "783925", "id": "589774"}]			
https://fataby.com/claim/103410	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/103410	https://www.2020-03-17	فاتبينو	صحيح	[{"id": "149109", "id": "82138"}]			
https://fataby.com/claim/645068	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/645068	https://www.2020-03-06	فاتبينو	صحيح	[{"id": "103410", "id": "322141"}]			
https://fataby.com/claim/165938	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/165938	https://www.2020-02-24	فاتبينو	صحيح	[{"id": "645068", "id": "645068"}]			
https://fataby.com/claim/215107	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/215107	https://www.2020-02-01	فاتبينو	صحيح	[{"id": "165938", "id": "201483"}]			
https://fataby.com/claim/371324	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/371324	https://www.2020-01-24	فاتبينو	صحيح	[{"id": "215107", "id": "247154"}]			
https://fataby.com/claim/228525	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/228525	https://www.2020-01-10	فاتبينو	صحيح	[{"id": "371324", "id": "371324"}]			
https://fataby.com/claim/208277	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/208277	https://www.2020-01-08	فاتبينو	صحيح	[{"id": "228525", "id": "190758"}]			
https://fataby.com/claim/396275	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/396275	https://www.2020-01-08	فاتبينو	صحيح	[{"id": "208277", "id": "561065"}]			
https://fataby.com/claim/216958	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/216958	https://www.2019-12-26	فاتبينو	صحيح	[{"id": "396275", "id": "468926"}]			
https://fataby.com/claim/8221	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/8221	https://www.2019-11-27	فاتبينو	صحيح	[{"id": "216958", "id": "502488"}]			
https://fataby.com/claim/271588	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/271588	https://www.2019-10-12	فاتبينو	صحيح	[{"id": "8221", "id": "8221"}]			
https://fataby.com/claim/21201	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/21201	https://www.2019-07-25	فاتبينو	صحيح	[{"id": "271588", "id": "729317"}]			
https://fataby.com/claim/19457	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/19457	https://www.2019-07-25	فاتبينو	صحيح	[{"id": "21201", "id": "603791"}]			
https://fataby.com/claim/361379	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/361379	https://www.2019-07-25	فاتبينو	صحيح	[{"id": "19457", "id": "887716"}]			
https://fataby.com/claim/502488	ما الحقيقة	ما الحقيقة	https://fataby.com/claim/502488	https://www.2019-07-25	فاتبينو	صحيح	[{"id": "361379", "id": "337973"}]			

Schéma 4.2 – fichier csv final de fatabayano

claim_url	Rating_value	Rating_auth	Rating_auth	Fake	Claimer	Entity	Created	Url	Created	Url	Claim	Review	Claim	Review	Claim	Review	Claim	Review	Claim	Review
								</												

Schéma 4.3 – fichier csv de vishvasnews

4.2 Problèmes rencontrés

Le tout premier problème que nous avons rencontré a été lors de l'exécution des scripts d'extraction du projet ClaimsKg. En effet, la majorité des scripts que nous avons essayés ne marchaient plus, certains étaient anciens et n'étaient plus en adéquation avec les mises à jour, soit du projet en lui-même, soit du site web et cause de cela, les bases sur lesquelles nous devions prendre nos marques étaient à prendre avec des pincettes.

Le problème majeur suivant a été de trouver un moyen de faire une « named entity recognition » en langue arabe, car le programme "tagme" ne peut être utilisé sur l'arabe. Nous avons donc cherché des logiciels en open source qui pouvait nous aider à résoudre ce problème, malheureusement, aucun logiciel n'a été satisfaisant ; certains nous ont permis comme "Arabic NER" ou "NERAr" de faire des entity recognition, mais sans retourner les liens Wikipedia de ces derniers, d'autres comme "AiDA" qui semblaient être la solution ne fonctionnaient tout simplement pas. Nous avons donc décidé de traduire les claims et leurs reviews afin d'appliquer le programme "tagme", mais là aussi, les API de traduction ne marchaient pas correctement comme googletans (googletans est différente de google translate qui n'est pas gratuite) ou étaient limitées par jour comme celle que nous avons utilisé "Yandex". Le souci avec cette api était que le nombre de caractère traduit quotidiennement était limité.

Le dernier gros problème rencontré quant à lui est le fait que trois des étudiants de notre groupe ont eu des soucis avec leurs machines - deux pc tombés en panne et un chargeur perdu - quelques jours après le début du confinement. Cette contrainte nous a empêché d'aller aussi loin dans la réalisation du projet et nous a retardé sur l'avancement global.

Chapitre 5

Bilan et conclusions

5.1 Ce qu'on a fait et pas fait

5.2 Perspective

Annexe A

Annexe