Université de Paris

Catégorisation de données textuelles multilingues pour la détection d'opinions

Encadré par Rogovschi Nicoleta

L 3 A A 1

MANUEL D'UTILISATION

Laforge Johan – Tanriverdi Messie – Thay Jacky – Trividic Yann

Référence du document : L3AA1_2021_MU

Version du document : 1.00

Date du document : 18/04/2021

Auteur : Laforge Johan, Tanriverdi Messie, Thay Jacky, Trividic Yann

Validé par : Trividic Yann

Validé le : 18/04/2021

Soumis le : 18/04/2021

Type de diffusion : Document électronique (PDF)

Confidentialité : Standard – Étudiants et corps enseignant de l'UFR Mathématiques et Informatique de l'Université de Paris

Mots-clés : manuel d'utilisation, documentation, descartes, université, paris, math-info, licence, L3AA1, catégorisation, données, textuelles, multilingues, détection opinion, opinion mining, artificial intelligence, sentiment analysis, Python, nlp.

Table des matières

Introduction **4**

Guide de lecture **4**

2.1. Maîtrise d'œuvre et maîtrise d'ouvrage 4

2.1.1. Responsables 4

2.1.2. Personnels administratif et technique 5

Fenêtre principale **6**

3.1. Menu principal 7

3.2. Dashboard 7

3.2.1. Bouton Home 8

3.2.2. Barre de recherche 8

3.2.3. Platform 8

3.2.4. Source language 8

3.2.5. Target language 9

3.2.6. API status 9

3.2.7. Bouton About 10

Résultats **10**

4.1. Onglet Analytics 10

4.1.1. Occurrences des mots les plus communs 11

4.1.2. Frise temporelle 11

4.1.3. Polarité et diagramme circulaire 12

4.1.4. Nombre de données temporelles 12

4.2. Onglet Charts 13

4.2.1. Exemple de résultats 13

4.2.2. Descriptifs des colonnes du tableau généré 14

Formulation de requêtes **15**

5.1. Paramètres communs à toutes les requêtes 16

5.2. Paramètres spécifiques aux recherches locales 17

5.3. Paramètres spécifiques aux recherches Twitter 18

5.4. Paramètres spécifiques aux recherches IMDb 18

5.5. Exemples de requêtes 19

Importation de fichiers CSV **19**

Exportation de fichiers CSV **20**

Messages d'erreur **20**

8.1. Au démarrage 21

8.2. Concernant les requêtes 21
8.3. Importation et exportation de fichiers CSV 22

Limitations des API **22**
9.1. Twitter 23
9.2. Google Translate 23
9.3. detectlanguage 23

Glossaire **25**

1. Introduction

Ce document a pour but de définir les attentes du maître d'ouvrage envers le maître d'œuvre durant la réalisation du projet. Le non-respect de ces attentes pourra donc engendrer des pénalités. Par ailleurs, le cahier des charges constitue une pièce de références du contrat pour les deux parties (client et fournisseur) en levant toutes les ambiguïtés possibles. Cela permet donc de fournir une livraison au plus proche des attentes du client.

Les termes en gras dans ce texte disposent d'une entrée dans le glossaire, section 10 de ce document.

Pour une lecture compréhensive de ce document, il est recommandé au lecteur de se munir des documents *Cahier des charges*, *Cahier de recette*, *Conception détaillée* et *Manuel d'installation*.

2. Guide de lecture

2.1. Maîtrise d'œuvre et maîtrise d'ouvrage

Les différents acteurs de la maîtrise d'œuvre et de la maîtrise d'ouvrage étant identiques, nous nous sommes permis dans ce document, par souci de concision, de fusionner ces différentes catégories.

2.1.1. Responsables

Rogovschi Nicoleta
Contact : nicoleta.rogovschi@u-paris.fr
Mme Rogovschi est enseignante-chercheuse à l'Université de Paris dans l'UFR de Mathématiques et d'Informatique. Elle est la maîtresse d'ouvrage, la maîtresse d'œuvre et l'encadrante principale de ce projet.

Janiszek David
Contact : david.janiszek@u-paris.fr
M. Janiszek est enseignant-chercheur à l'Université de Paris dans l'UFR de Mathématiques et d'Informatique, dont il est aussi le directeur.
Il pilote le cours "Projet tutoré" (IF06M030) dans lequel s'inscrit ce projet (voir section 4).

2.1.2. Personnels administratif et technique

Laforge Johan
Contact : laforgejohan7@gmail.com
Étudiant en 3e année de licence MIAGE à l'Université de Paris.
A intégré en 2017 l'UFR de Mathématiques et d'Informatique.
Développeur Python dans le cadre de ce projet.

Tanriverdi Messie
Contact : tanriverdi.messie@gmail.com
Étudiante en 3e année de licence Informatique et Applications à l'Université de Paris.
A intégré en 2017 l'UFR de Mathématiques et d'Informatique.
Développeuse Python dans le cadre de ce projet.

Thay Jacky
Contact : jacky.thay@yahoo.com
Apprenant en 3e année de licence Informatique et Applications à l'Université de Paris.
A intégré en 2020 l'UFR de Mathématiques et d'Informatique.
Développeur Python dans le cadre de ce projet.

Trividic Yann
Contact : yann.trividic@hotmail.fr
Étudiant en 3e année de licence Informatique et Applications à l'Université de Paris.
A intégré en 2017 l'UFR de Mathématiques et d'Informatique.
Développeur Python dans le cadre de ce projet.

3. Fenêtre principale

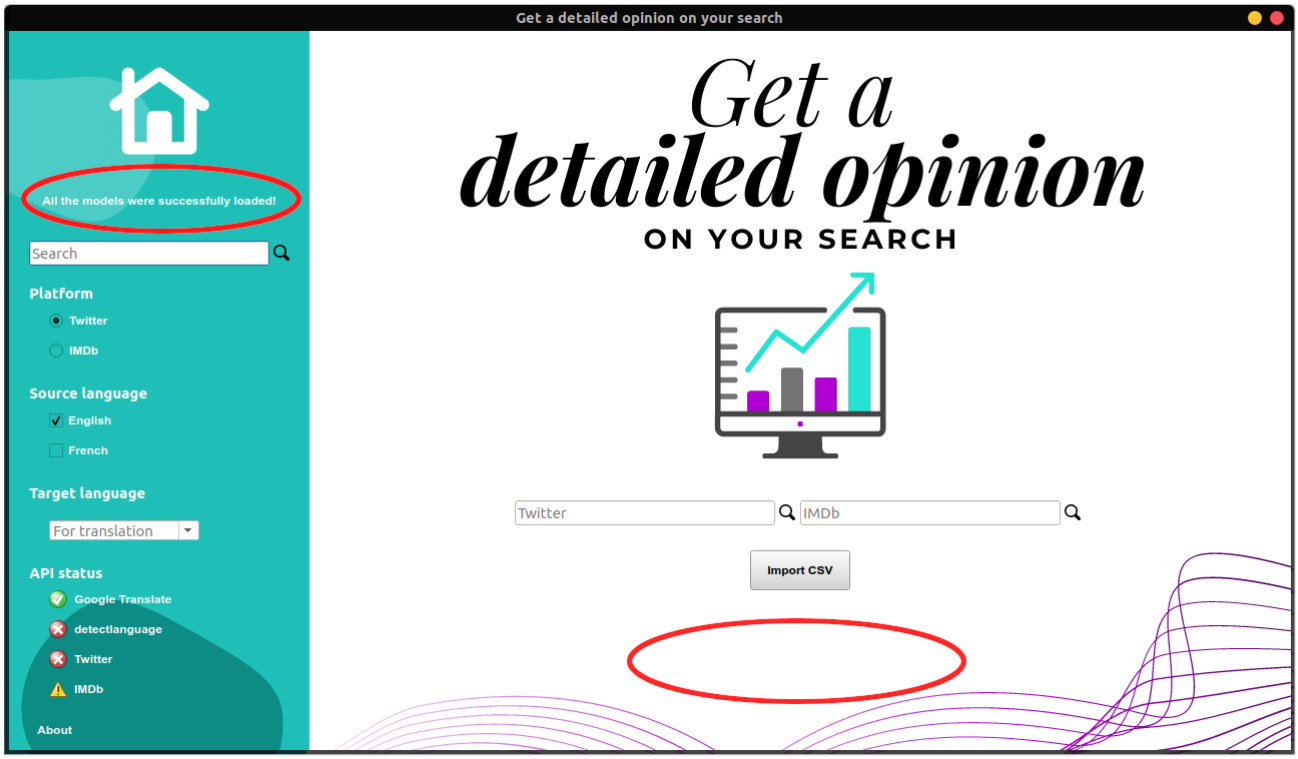


Figure 1 : *Vue du logiciel juste après que toutes les données aient été correctement chargées*

Le logiciel s'ouvre directement dans la fenêtre principale du programme. Un temps de chargement est nécessaire pour charger tous les modèles d'intelligence artificielle pré entraînés permettant la classification des données textuelles selon leur polarité. Ce chargement se fait de manière transparente pour l'utilisateur.

La fenêtre se divise en deux parties : le ***dashboard*** (à gauche) et le menu principal (à droite). Les deux sous-parties suivantes ont pour but de décrire l'utilité et les manières d'utiliser ces différents panneaux.

3.1. Menu principal

Fenêtre d'accueil du logiciel, le menu principal met à disposition de l'utilisateur une série de barres de recherche dédiées aux différentes sources de données possibles. Dans un premier temps, seules deux barres de recherche sont disponibles : celle pour **IMDb**, et celle pour **Twitter**. La troisième barre de recherche, dédiée exclusivement à l'analyse des fichiers chargés localement, est rendue disponible après qu'un fichier **CSV** ait été correctement chargé par l'utilisateur (voir section *6. Importation de fichiers CSV*).

Dans ce menu, l'utilisateur n'a donc pas à spécifier la source des données à extraire : la source est choisie en fonction de la barre de recherche dans laquelle la requête est formulée par l'utilisateur. La syntaxe de chaque requête formulée doit respecter la syntaxe décrite dans la section *5. Formulation de requêtes*.

Si l'utilisateur a formulé une requête valide, alors un **GIF** de chargement vient remplacer la loupe placée à droite de l'espace de saisie choisi. Lorsque tous les calculs sont terminés (extraction, classification et potentiellement traduction), l'utilisateur est redirigé sur la fenêtre de résultats (voir section *4. Résultats*). **Ce temps de chargement peut être plus ou moins long** : il dépend de la rapidité de votre connexion Internet si des données sont à extraire du web, du volume de données à traiter, ainsi que de la puissance de votre processeur. Sur les configurations les plus minimales, certaines requêtes peuvent nécessiter plusieurs dizaines de secondes avant d'être menées à bien.

À l'inverse, si la requête formulée est invalidée par notre interpréteur de requêtes ou n'a pas pu aboutir sur une extraction de données, l'utilisateur en est notifié. Un message en rouge s'affiche alors dans l'ellipse rouge en dessous du bouton `Import CSV`.

3.2. Dashboard

Le **dashboard** permet à l'utilisateur de préciser certains paramètres de sa recherche, de retourner au menu principal, d'obtenir des informations sur l'état de disponibilité des différentes API, et d'obtenir quelques informations sur le contexte du projet. Les sous-sections suivantes sont à propos des différentes composantes du *dashboard.* Leur ordre est le même que celui du *dashboard* de haut en bas.

Les différents paramètres décrits ci-dessous sont tous expliqués plus en détail dans la section *5. Formulation de requêtes*. Veuillez-vous y référer pour une description plus précise.

3.2.1. Bouton *Home*

Le bouton *Home* a deux utilisations principales : retourner au menu principal lorsque les panneaux de résultats sont affichés (voir section *4. Résultats*), et partiellement réinitialiser l'interface. Appuyer sur le bouton *Home* nettoie le contenu des différents espaces de saisie et arrête les requêtes en cours d'exécution pour redonner la main à l'utilisateur.

3.2.2. Barre de recherche

La barre de recherche du *dashboard*, a contrario, permet de formuler des requêtes à la fois concernant IMDb, et des requêtes concernant **Twitter**. La valeur par défaut est Twitter, et peut-être changée en cliquant sur les boutons radio sous *Platform.*

3.2.3. *Platform*

Platform permet de préciser la source des données à extraire : des ***tweets*** extraits directement de Twitter, ou des critiques de films extraites d'IMDb. Ces boutons radio ne permettent pas de sélectionner une recherche locale, qui est accessible uniquement via la barre dédiée dans le menu principal (voir section *6. Importation de données CSV*).

3.2.4. *Source language*

La section *Source language* permet à l'utilisateur de préciser la ou les langues dans lesquelles il veut extraire et analyser ses données. Ce paramètre a plusieurs utilités selon la source des données précisées et selon l'état de disponibilité de l'API detectlanguage.

- Utilisé avec Twitter : l'utilisateur précise la langue dans laquelle seront extraits les tweets.
- Utilisé avec IMDb : seul l'anglais est disponible car IMDb ne contient que des données anglophones.
- Utilisé avec des données locales : en mode hors-ligne, la langue précisée est la langue supposée du jeu de données chargé par l'utilisateur. En mode connecté, si l'utilisateur dispose de crédits d'API detectlanguage, alors les données extraites seront analysées

pour déterminer leurs langues, et seules les phrases dans les langues précisées par l'utilisateur seront retenues.

Pour plus d'informations, se référer au paramètre `lang` de la section *5. Formulation de requêtes*.

3.2.5. *Target language*

Cette section, sous la forme d'un menu déroulant, offre à l'utilisateur la possibilité de traduire les données extraites dans la langue de son choix. Pour préciser plus d'une langue cible, il est possible à l'utilisateur d'ajouter autant de paramètres `target` qu'il le souhaite dans la barre de recherche choisie. Plus plus d'informations, se référer au paramètre `target` de la section *5. Formulation de requêtes*. Les possibilités liées à ce paramètre découlent directement des limitations inhérentes à l'API Google Translate. Pour plus d'informations, voir les sections *3.2.6 API status* et *9. Limitation des API*.

3.2.6. *API status*

L'interface étant très dépendante à l'état de la connexion Internet de la machine et à la disponibilité des différents services d'API, un panneau d'affichage de l'état des différents services se trouve sur le *dashboard*.

Cette section dispose donc de quatre voyants permettant à l'utilisateur de savoir l'état de disponibilité des différents services en ligne utilisés pour le logiciel. Les voici :

- **Google Translate : au vert**, il est possible de lancer des traductions. S'il venait à manquer de crédits pour compléter la traduction de vos données, alors le voyant passera au rouge et les colonnes de traductions n'apparaîtront pas dans l'onglet *Charts* (voir section *4.2. Onglet Charts*). Si le voyant est **au rouge** avant de passer votre requête, alors vous ne pourrez pas préciser de traduction.
- **detectlanguage : au vert**, il vous est possible d'effectuer des détections de langues sources comme indiqué dans la section *3.2.4. Source language*. S'il venait à manquer de crédits pour compléter la détection automatique des langues de vos données, alors le voyant passera au rouge et la colonne `src_lang` n'apparaîtra pas dans l'onglet *Charts* (voir section *4.2. Onglet Charts*). De plus, la langue utilisée pour la classification sera celle précisée au début de votre requête. **Au rouge**, il est impossible d'utiliser les fonctionnalités de détection de langues.

- **Twitter :** L'extraction des données de Twitter par le logiciel repose sur l'utilisation de l'API Twitter. Celle-ci est limitée. Si l'une des limites d'extraction des données est atteinte, alors le voyant apparaîtra **en rouge**. Il vous sera possible d'exécuter certains types de requêtes, mais quelques fonctionnalités risquent de manquer et les données extraites seront alors incomplètes. Dans ce cas-là, il est recommandé à l'utilisateur d'attendre quelques heures puis réessayer. **Au vert**, les requêtes Twitter peuvent être effectuées sans problème.
- **IMDb :** L'extraction des données IMDb repose sur du *scraping* du site www.imdb.com. Pour ce faire, le logiciel utilise des ***webdrivers***. C'est donc l'état de chargement des webdrivers qui est vérifié ici, ainsi que la connexion à internet de la machine. Dans certains environnements, il n'est pas possible de détecter si un *webdriver* a bien été chargé. Dans ce cas-là, le voyant est **au jaune**. **Au rouge**, le webdriver n'a pas pu être chargé et les requêtes IMDb ne pourront pas aboutir. Il vous est alors recommandé d'installer un navigateur Google Chrome ou Firefox. Si vous disposez déjà de ces navigateurs, essayez de les mettre à jour. **Au vert**, le *webdriver* associé à votre navigateur a bien été chargé.

3.2.7. Bouton *About*

Le bouton *About* sert à ouvrir une fenêtre externe contenant des informations contextuelles sur le projet.

4. Résultats

4.1. Onglet *Analytics*

Cet onglet représente le cœur de la visualisation des données développée dans le programme. Après avoir extrait, catégorisé, puis analysé les données obtenues suite à la requête formulée par l'utilisateur, la fenêtre de résultats s'ouvre sur l'onglet *Analytics*. Jusqu'à quatre graphiques peuvent être présentés à l'utilisateur. Les quatre sous-sections suivantes vous décrivent ce qui est affiché *figure 2* en suivant une lecture de gauche à droite et de haut en bas.

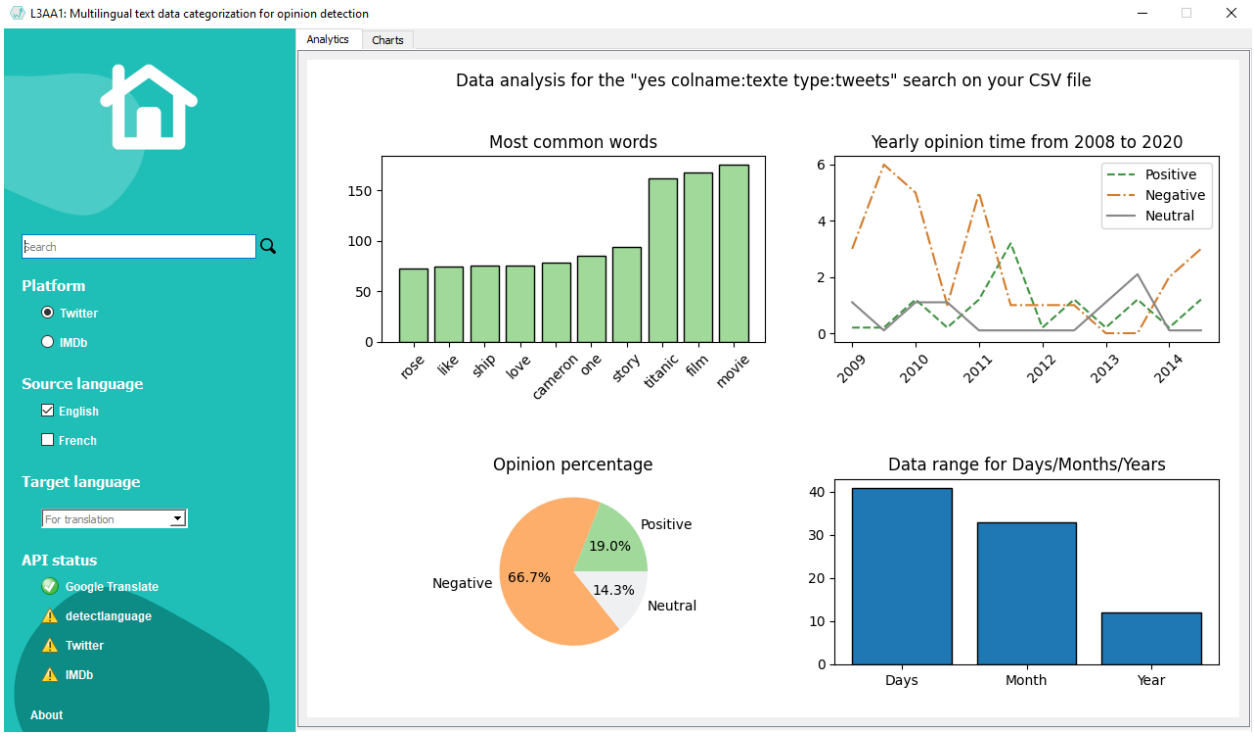


Figure 2 : *Vue de l'onglet* Analytics *après avoir effectué la requête "yes" sur un fichier CSV.*

4.1.1. Occurrences des mots les plus communs

En haut à gauche, un histogramme vous présente les mots ayant le plus grand nombre d'occurrences dans les données extraites. Le nombre d'occurrences est sur l'axe des ordonnées, tandis que les mots sont sur les axes des abscisses.

4.1.2. Frise temporelle

En haut à droite, une frise temporelle vous représente l'évolution de la polarité des textes extraits à partir de votre requête en fonction du temps. Trois courbes apparaissent : le nombre d'occurrences de phrases (tweets, critiques, etc.) catégorisées comme **positives**, **négatives** et **neutres** sont respectivement colorées en **vert**, en **orange** et en **gris**.

La nature des graphiques peut évoluer en fonction de leur pertinence à être visualisés. Ainsi, la frise pourra montrer une évolution de la polarité annuelle, mensuelle ou quotidienne en fonction du le niveau de granularité des données temporelles extraites.

4.1.3. Polarité et diagramme circulaire

En bas à gauche, vous pouvez retrouver la part des différentes polarités extraites sous la forme d'un diagramme circulaire, le même code couleur est suivi que dans la section *4.1.2. Frise temporelle*.

4.1.4. Nombre de données temporelles

En bas à droite, en appoint de la frise temporelle est ajouté le nombre de données temporelles. La couleur n'a pas d'importance ici.

Attention : suite à des problèmes de compatibilité, le nuage de mots initialement prévu (voir le graphique bas à droite de la figure 3) a dû être retiré de la version publique du code source.

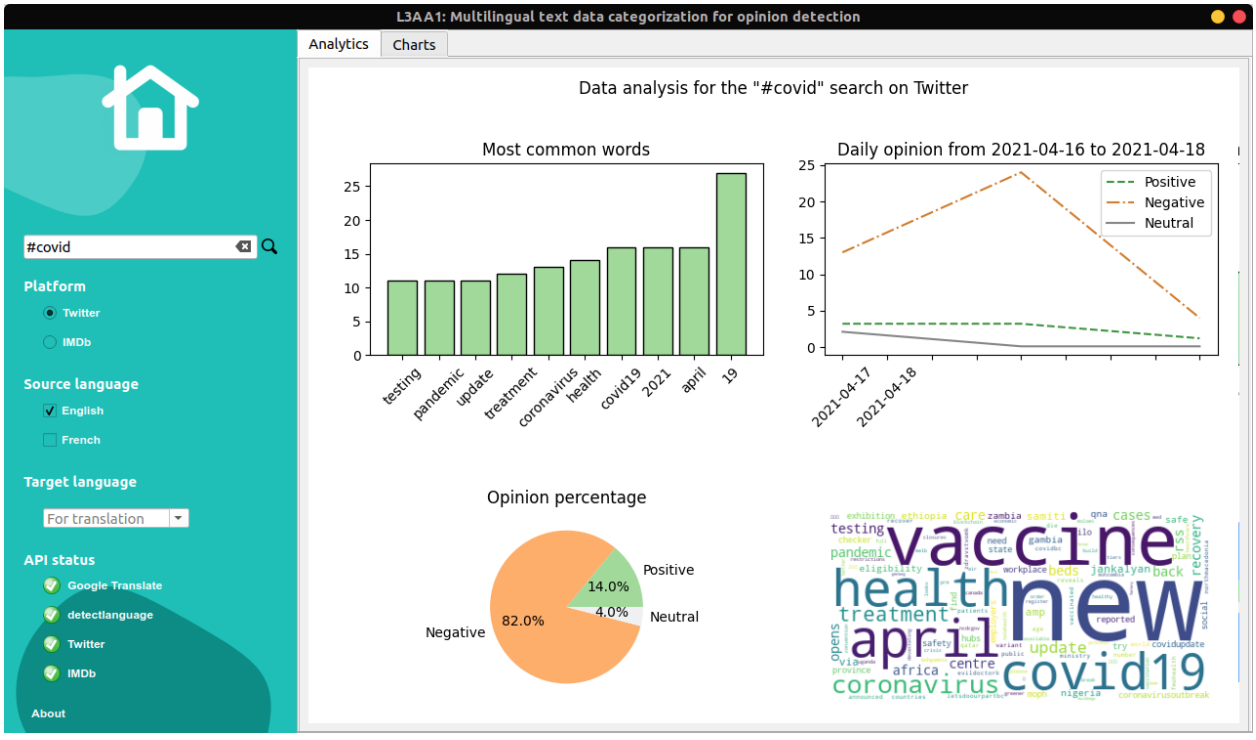
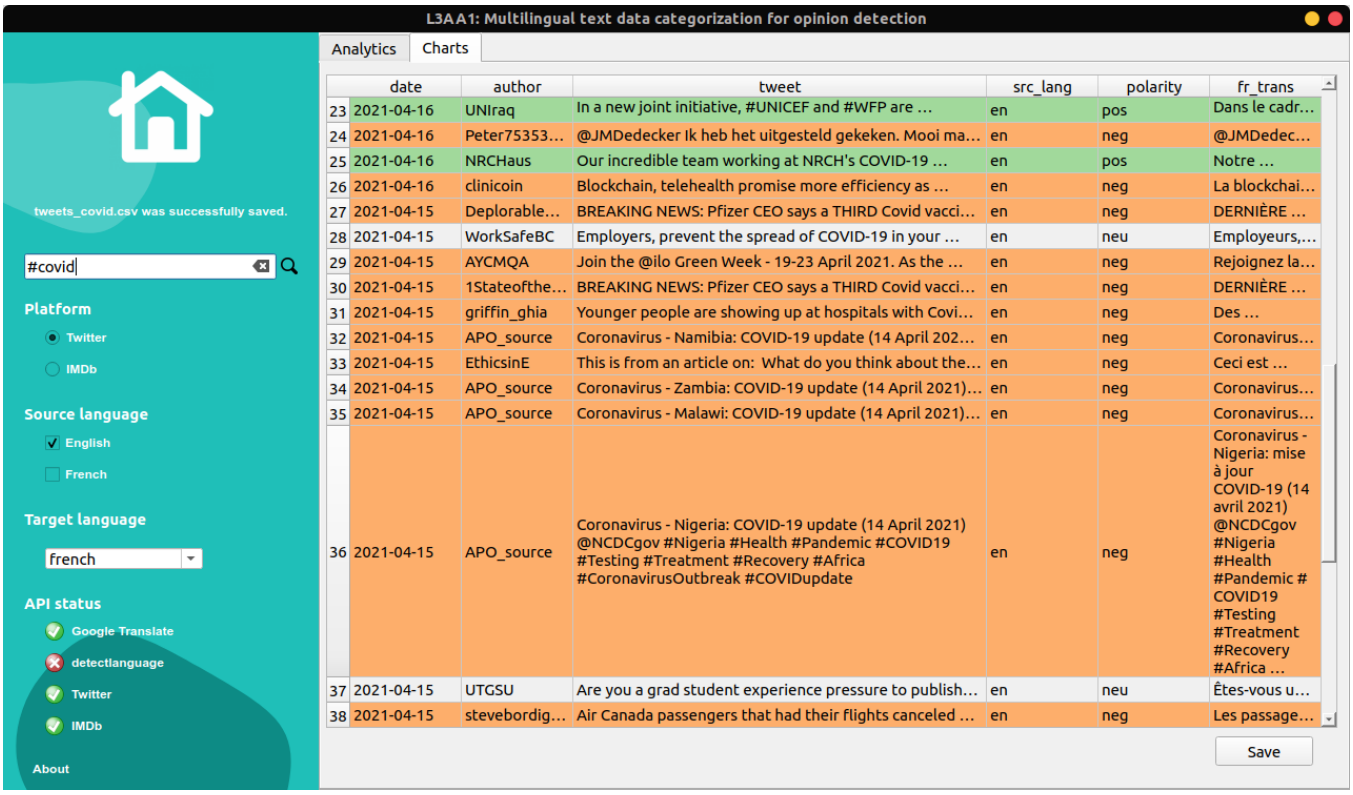


Figure 3 : *Vue du panneau* Analytics *après la recherche "#covid" avant la suppression du nuage de mots*

4.2. Onglet *Charts*

Le deuxième onglet de la fenêtre de résultats, l'onglet *Charts*, permet de visualiser les données extraites ou générées par nos algorithmes sous forme de tableau. Selon le type de requête passé, les colonnes de la table de données pourront varier. Ci-dessous, un exemple de visualisation de l'onglet avec une requête basique.

4.2.1. Exemple de résultats



	date	author	tweet	src_lang	polarity	fr_trans
23	2021-04-16	UNIraq	In a new joint initiative, #UNICEF and #WFP are …	en	pos	Dans le cadr…
24	2021-04-16	Peter75353…	@JMDedecker Ik heb het uitgesteld gekeken. Mooi ma…	en	neg	@JMDedec…
25	2021-04-16	NRCHaus	Our incredible team working at NRCH's COVID-19 …	en	pos	Notre …
26	2021-04-16	clinicoin	Blockchain, telehealth promise more efficiency as …	en	neg	La blockchai…
27	2021-04-15	Deplorable…	BREAKING NEWS: Pfizer CEO says a THIRD Covid vacci…	en	neg	DERNIÈRE …
28	2021-04-15	WorkSafeBC	Employers, prevent the spread of COVID-19 in your …	en	neu	Employeurs,…
29	2021-04-15	AYCMQA	Join the @ilo Green Week - 19-23 April 2021. As the …	en	neg	Rejoignez la…
30	2021-04-15	1Stateofthe…	BREAKING NEWS: Pfizer CEO says a THIRD Covid vacci…	en	neg	DERNIÈRE …
31	2021-04-15	griffin_ghia	Younger people are showing up at hospitals with Covi…	en	neg	Des …
32	2021-04-15	APO_source	Coronavirus - Namibia: COVID-19 update (14 April 202…	en	neg	Coronavirus…
33	2021-04-15	EthicsinE	This is from an article on: What do you think about the…	en	neg	Ceci est …
34	2021-04-15	APO_source	Coronavirus - Zambia: COVID-19 update (14 April 2021)…	en	neg	Coronavirus…
35	2021-04-15	APO_source	Coronavirus - Malawi: COVID-19 update (14 April 2021)…	en	neg	Coronavirus…
36	2021-04-15	APO_source	Coronavirus - Nigeria: COVID-19 update (14 April 2021) @NCDCgov #Nigeria #Health #Pandemic #COVID19 #Testing #Treatment #Recovery #Africa #CoronavirusOutbreak #COVIDupdate	en	neg	Coronavirus - Nigeria: mise à jour COVID-19 (14 avril 2021) @NCDCgov #Nigeria #Health #Pandemic # COVID19 #Testing #Treatment #Recovery #Africa …
37	2021-04-15	UTGSU	Are you a grad student experience pressure to publish…	en	neu	Êtes-vous u…
38	2021-04-15	stevebordig…	Air Canada passengers that had their flights canceled …	en	neg	Les passage…

Figure 4 : *Vue de l'onglet* Charts *après avoir effectué la requête "#covid" en ayant sélectionné comme langue source* `english` *et comme langue cible* `french`

On remarque que les données en **orange** sont celles dont la polarité a été détectée comme étant **négative**, et celles en **vert** correspondent aux données pour lesquelles la polarité est **positive**. En **blanc** sont les données **neutres**.

Il est possible à l'utilisateur de **voir l'entièreté du contenu d'une ligne** en double-cliquant en bas du numéro de ligne. Ici, pour étendre la hauteur de la ligne, le bas de la case où figure le numéro de ligne 36 a été double-cliqué.

Il est possible de **faire défiler les données verticalement ou horizontalement,** avec les barres de défilement apparaissant respectivement à droite et en bas du tableau. Ici, seule la barre verticale est présente, la largeur de la table ne s'étendant que si nécessaire.

En haut à gauche, on peut voir que l'utilisateur a **enregistré ses résultats** dans un fichier CSV (voir la section *7. Exportation de fichiers CSV*).

Les **données ayant été catégorisées par nos algorithmes** apparaissent par défaut dans une colonne plus large que les autres.

4.2.2. Descriptifs des colonnes du tableau généré

Le tableau de données généré aura différentes colonnes selon la source des données et selon les différents paramètres renseignés par l'utilisateur. Voici un descriptif des différents types de colonnes :

- **`tweet` :** lorsque la source des données est Twitter, alors les *tweets* extraits sont placés dans la colonne `tweet`.
- **`review` :** lorsque la source des données est IMDb, alors les critiques de films extraites sont placées dans la colonne `review`.
- **`author` :** lorsque la source est soit IMDb soit Twitter, alors l'auteur du *tweet* ou de la critique est spécifié dans la colonne author.
- **`date` :** lorsque la source est soit IMDb, soit Twitter, alors la date de publication du *tweet* ou de la critique est spécifié dans la colonne `date, dans le format AAAA-MM-JJ`.
- **`*_trans` :** lorsqu'une traduction est demandée par l'utilisateur, les données catégorisées sont traduites dans la colonne correspondante. Ici, l'astérisque symbolise la chaîne de caractère associée à la langue cible dans la norme ISO639-1[1]. Dans l'exemple de la section précédente, on retrouve `fr_trans` pour une traduction en français.
- **`src_lang` :** il s'agit de la langue source détectée et de la langue qui a été paramétrée pour catégoriser les données textuelles correspondant à cette ligne. Deux valeurs sont possibles : `fr` et `en`, respectivement pour français et anglais dans la norme ISO639-1.
- **`polarity` :** résultat de la classification des données textuelles, cette colonne contient une chaîne de caractère attribuée par notre algorithme pour décrire la polarité du texte. Cette chaîne peut prendre trois valeurs : `pos`, `neg` ou `neu`, respectivement pour positif, négatif ou neutre.

[1] https://fr.wikipedia.org/wiki/Liste_des_codes_ISO_639-1

- Lorsque vous effectuez une recherche sur un fichier chargé localement, les différentes colonnes de ce fichier sont gardées à gauche de la table. Les colonnes citées précédemment sont ajoutées sur la droite du tableau.

5. Formulation de requêtes

La manipulation du logiciel repose en grande partie sur la formulation par l'utilisateur de requêtes textuelles. Ces requêtes sont construites à partir d'un **modèle syntaxique précis**, permettant à l'utilisateur différents degrés de granularité en fonction du type de résultats qu'il veut obtenir.

Il existe trois sources de données possibles pour notre logiciel : les données extraites de **Twitter**, celles extraites d'**IMDb**, et les fichiers **CSV** fournis par l'utilisateur. La syntaxe des différentes requêtes dépend de la source choisie par l'utilisateur. Certains **décorateurs** de mots-clés sont communs aux différents types de sources, tandis que d'autres sont spécifiques à la source précisée. Par exemple, lorsque Twitter est choisi comme source, les requêtes devront préciser un hashtag, ou un compte Twitter.

Chacune des sources est associée à une barre de recherche de la fenêtre principale. La barre de recherche du ***dashboard*** permet quant à elle de choisir entre IMDb et Twitter en fonction de l'état des boutons radios (voir *figure 1*).

Certains paramètres doivent être précisés pour qu'une recherche soit considérée comme valide. Chaque requête doit donc contenir au moins l'un d'entre eux pour être acceptée. Ces paramètres sont alors dits axiomatiques à la recherche, et sont symbolisés ici **en rouge**.

Certains décorateurs peuvent être précisés en plaçant un caractère spécial en début de mot (le croisillon pour préciser les ***hashtags***), au début et à la fin d'une série de mots (les guillemets pour préciser le nom d'un film), ou alors en précisant explicitement le paramètre choisi. Ces filtres explicites se construisent sous la forme suivante : `filtre:valeur`, où `filtre` correspond à l'un des filtre explicite décrit ci-dessous, et `valeur` correspond à l'une des valeurs possibles pour le filtre explicite choisi.

5.1. Paramètres communs à toutes les requêtes

- **`search` :** Contient les mots-clés que l'utilisateur veut rechercher. Si plusieurs mots-clés sont notifiés, alors chaque phrase extraite devra contenir chacun des mots-clés. Par défaut, tout mot n'étant associé à aucun décorateur est considéré comme étant un paramètre de `search`.

  Paramètre axiomatique aux recherches locales uniquement.

  Valeur par défaut : ∅

  Exemples de syntaxe : `lorem ispum`

- **`lang` :** La langue dans laquelle vous souhaitez extraire vos résultats. Par exemple, si la langue `french` est spécifiée, alors seuls les résultats en français seront extraits. Si plusieurs langues sont spécifiées, alors les résultats dans ces différentes langues seront extraits. Ces paramètres peuvent être spécifiés grâce aux deux checkboxes sous *Source language* dans le *dashboard*. La détection de langue est effectuée via l'**API** detect-language[2]. En l'état, le logiciel utilise une version gratuite de l'API. Celle-ci est limitée à mille requêtes par jour. L'état de disponibilité de l'API detect-language est spécifié sous *API status* dans le *dashboard*. Si vos crédits d'API quotidiens ont été épuisés, alors `lang` servira à spécifier la langue dans laquelle sont les données que vous analysez. Les valeurs données à ce paramètres peuvent être des noms de langues en anglais ou peuvent respecter la norme ISO639-1. Pour plus d'informations, consulter la section 3.2.4. *Source language.*

  Valeur par défaut : `english`

  Exemples de syntaxe : `lang:fr lang:english`

- **`target` :** Les langues cibles dans lesquelles seront traduites les résultats extraits. Ce paramètre peut être spécifié grâce au menu déroulant sous *Target language*. Cette fonctionnalité est disponible dans la limite des possibilités offertes par la version courante du logiciel, à savoir une version d'utilisation de l'API Google Translate. En l'état, le logiciel utilise une version gratuite de l'API. Celle-ci peut supporter une certaine fréquence de requêtes avant d'être notifiée comme étant indisponible. L'état de

[2] https://detectlanguage.com/

disponibilité de l'API Google Translate est spécifié sous *API status* dans le *dashboard*. Les valeurs prises par ce paramètres peuvent être choisies via le menu déroulant, ou en respectant la norme ISO639-1.

Valeur par défaut : ∅

Exemples de syntaxe : `target:fr target:maori target:zulu target:en`

- **`maxentries` :** le nombre maximum d'entrées qui sera extrait pour votre requête. Les valeurs pouvant être prises par ce paramètre sont comprises entre `1` et `1000`. Rappel : les données extraites sont limitées avec l'API Twitter.

  Valeur par défaut : `50`

  Exemples de syntaxe : `maxentries:1 maxentries:1000`

5.2. Paramètres spécifiques aux recherches locales

- **`colname` :** Par défaut, l'algorithme cherchera à travers votre fichier la première occurrence d'un des mots-clés de `search`. Si la colonne trouvée n'est pas la bonne, l'algorithme pourra dans la plupart des cas vous en notifier. Dans ce cas, il vous est possible de spécifier la colonne dans laquelle sont contenues les données grâce au décorateur `colname`.

  Valeur par défaut : ∅

  Exemple de syntaxe : `colname:text colname:tweets`

- **`type` :** peut prendre pour valeur `reviews` ou `tweets`. Selon ce que vous choisissez, les données seront analysées comme des critiques de film ou comme des tweets. Les résultats seront plus précis si le type associé est le bon.

  Valeur par défaut : `tweets`

  Exemple de syntaxe : `type:tweets type:reviews`

5.3. Paramètres spécifiques aux recherches Twitter

- **@ :** Paramètre pour spécifier un utilisateur **Twitter**. Il doit être mis au début d'un mot. Les données extraites ne contiennent que des **tweets** rédigés par l'utilisateur spécifié.

  Valeur par défaut : ∅

  Exemples de syntaxe : `@potus @policenationale`

- **# :** Paramètre pour spécifier un ***hashtag***. Il doit être mis au début d'un mot. Les données extraites ne contiennent que des *tweets* contenant le hashtag spécifié.

  Valeur par défaut : ∅

  Exemples de syntaxe : `#covid #2021`

5.4. Paramètres spécifiques aux recherches IMDb

- **"" :** Les guillemets permettent de spécifier un nom de film. Le nom du film doit être entré entre deux guillemets. Il est possible que le titre ne suffise pas à trouver le film recherché (homonymes, etc.), dans ce cas-là, voir les décorateurs suivants.

  Valeur par défaut : ∅

  Exemples de syntaxe : `"Titanic" "Avatar"`

- **year :** Année de sortie du film, qui permet de préciser la recherche. Sans assez d'informations sur le film, c'est celui qui est sorti le plus récemment qui sera analysé.

  Valeur par défaut : ∅

  Exemple de syntaxe : `year:1962`

- **`id :`** Paramètre pour spécifier l'identifiant **IMDb** d'un film. Ce paramètre peut être utile quand préciser l'année du film n'a pas suffit à réduire le nombre de possibilités à 1. Les identifiants des films IMDb peuvent être trouvés sur le site d'IMDb[3].

  Valeur par défaut : ∅

  Exemples de syntaxe : `id:tt0054518 id:tt54518 id:0054518 id:54518`

5.5. Exemples de requêtes

Voici des idées de requêtes pour Twitter :

- #vacances france
- #france lang:en
- @nasa #mars

Pour IMDb :

- “gattaca”
- “titanic” year:1997
- “titanic” year:2012
- id:tt7286456 (identifiant du film “Joker” sorti en 2019)

6. Importation de fichiers CSV

L’importation, puis l’analyse, sans encombre des fichiers **CSV** par le logicielle dépend du contenu et de la structure du fichier soumis par l’utilisateur. Les fichiers doivent être sélectionnés en cliquant sur le bouton `Import CSV` puis en allant sélectionner le fichier choisi.

[3] pour la syntaxe exacte, voir https://fr.wikipedia.org/wiki/Mod%C3%A8le:Imdb_titre

Le fichier sélectionné doit avoir ses colonnes séparées par des **virgules**, et le contenu de chaque cellule doit être à l'intérieur d'un **couple de guillemets**. La **première ligne** du fichier doit contenir une ligne de titres de colonne.

Dans l'éventualité où le fichier contiendrait des données temporelles, celles-ci ne pourront être traitées que si elles respectent le formalisme `AAAA-MM-JJ` (par exemple, `2021-04-13`). Chacune des lignes doit être complète, si l'une des cases est vide, alors c'est la ligne entière qui ne sera pas prise en compte.

Si le fichier choisi a pu être chargé correctement, la barre de recherche locale apparaît avec le texte fictif suivant : `Search nom_du_fichier.csv`. Autrement, un message d'erreur apparaît sous le bouton `Import CSV`.

7. Exportation de fichiers CSV

Après analyse, les résultats obtenus peuvent être enregistrés au format **CSV** avec le nom de votre choix. Cette fonctionnalité est accessible en cliquant sur le bouton `Save` de l'onglet *Charts* (voir section *4.2. Onglet Charts*). Une fenêtre de dialogue s'ouvre, vous devrez alors naviguer jusqu'à l'emplacement souhaité et rentrer le nom de votre fichier avant validation.

Si le fichier a pu être enregistré correctement, alors un message de validation apparaîtra dans le *dashboard*.

Le format d'enregistrement du fichier est classique : les chaînes de caractères sont repérées par des **guillemets**, et les colonnes sont séparées par des **virgules**.

8. Messages d'erreur

Les messages d'erreur peuvent être présentés sous deux formes : les voyants dans la section *API status* et les retours textuels dans les deux zones prévues à cet effet, indiquées par les ellipses dans la figure 1 de la section *3. Fenêtre principale*. Etant donné que les différents scénarios possibles quant aux voyants en bas du *dashboard* sont discutés en section *3.2.6 API status*, nous nous attarderons ici exclusivement sur les messages d'erreur textuels cités précédemment.

8.1. Au démarrage

Au démarrage, le logiciel doit initialiser l'environnement de travail nécessaire au bon traitement des requêtes. Si l'un des fichiers n'est pas correctement chargé, ou si l'utilisateur n'est pas connecté à Internet, alors l'utilisateur en est notifié par un message rouge en dessous du bouton *Home*. Dans le cas contraire, si tout s'est bien déroulé au démarrage, alors un message en blanc notifie l'utilisateur que tout s'est bien passé.

8.2. Concernant les requêtes

L'extrême majorité des messages d'erreurs qui seront rencontrés par l'utilisateur feront suite à l'envoi d'une requête par ce dernier. En effet, il existe de nombreux cas dans lesquels la requête formulée par l'utilisateur ne pourra pas aboutir sur des résultats. Ces messages s'afficheront dans le menu principal ou dans le *dashboard* selon la zone de saisie utilisée pour envoyer la requête. En voici la liste exhaustive :

- **La requête contient des caractères interdits.** Les mots-clés de base peuvent contenir des caractères alphanumériques, des tirets et des tirets bas. Pour les noms de films viennent s'ajouter les caractères suivants : **`:;'/!?().`**. Afin de préciser plusieurs filtres ou plusieurs mots-clés, veuillez les séparer par des espaces.

- **La machine n'est pas connectée à Internet.** Dans ce cas-là, si une requête est lancée sur IMDb ou sur Twitter, ou si la requête locale a besoin d'une API pour être menée à bien, alors l'utilisateur sera notifié.

- **L'utilisateur a lancé une requête sans paramètre axiomatique (voir la section *5. Formulation de requêtes*).**

- **L'utilisateur cherche à effectuer une requête sur IMDb, mais n'a pas de versions récentes ni de Google Chrome ni de Firefox installées** (voir la section API status et le document *Manuel d'installation* pour plus d'informations).

- **L'utilisateur utilise des décorateurs incompatibles (voir section *5. Formulation de requêtes*) avec la source de données choisie.**

- **L'utilisateur a entré une valeur interdite ou erronée pour un paramètre.**

- **L'utilisateur a décoché toutes les langues disponibles dans le panneau *Source language*** et est notifié avec un message d'avertissement mais peut quand même envoyer des requêtes.

- **Les crédits d'API sont épuisés (voir section 9. Limitations des API).**

- **L'utilisateur a tenté d'envoyer une requête multiple** alors qu'elles ne sont pas encore implémentées (voir le document *Conception détaillée*).

- La requête formulée est valide mais **l'extraction des données n'a pas pu produire de résultats.**

- La requête formulée est valide, mais **une erreur inattendue s'est produite lors de l'extraction des données** : l'utilisateur est notifié.

- La requête formulée est valide, l'extraction a pu se faire correctement mais **l'affichage des graphiques a rencontré une erreur** : l'utilisateur est notifié et seulement l'onglet *Charts* affichera des résultats.

8.3. Importation et exportation de fichiers CSV

La lecture et l'écriture des fichiers **CSV** peut poser problème dans le cas d'un chemin erroné, ou dans le cas d'un fichier corrompu. Dans ces différents cas, l'utilisateur sera notifié à travers des messages textuels en rouge affichés dans l'interface.

9. Limitations des API

En l'état, le logiciel est complètement gratuit : il ne nécessite aucun frais de fonctionnement propre. Les différents services en ligne utilisés (les différentes **API**) limitent les capacités du

logiciel, et donc le potentiel des usages de l'utilisateur. Pour une meilleure expérience, l'utilisateur est donc recommandé d'utiliser ses crédits d'API avec parcimonie. Bien évidemment, la question des limitations de ces API ne se pose uniquement dans le cas où la machine sur laquelle tourne le logiciel est connectée à Internet. Il est possible à l'utilisateur d'utiliser ses propres clés d'API dans le cas où il choisirait d'exécuter le programme à partir du code source, ou de compiler lui-même le programme. Ces possibilités sont discutées plus en détail dans le *Manuel d'installation*.

Trois API nécessitent d'être mentionnées ici : celles de Twitter, de Google Translate et de detectlanguage.

9.1. Twitter

La version utilisée dans le logiciel est la version *Standard* de l'API Twitter. Cela signifie que les requêtes passées par l'utilisateur ne peuvent pas excéder environ cinq cent tweets par heure. Passé cette limite, il vous faudra attendre avant de pouvoir à nouveau passer des requêtes. Les tweets extraits auront forcément été publiés durant les sept derniers jours, l'API ne permettant d'avoir accès qu'à ces derniers.

9.2. Google Translate

La version utilisée dans le logiciel est la version gratuite de l'API Google Translate. L'utilisateur est donc limité dans le volume de données qu'il peut envoyer aux serveurs de Google. Concernant les fonctionnalités de traduction automatique, l'utilisateur ne pourra donc pas effectuer plus de 1 000 requêtes par heure, et pas plus de 5 000 caractères par requête.

9.3. detectlanguage

Le système de classification par polarité ne supporte pour le moment que deux langues : le français et l'anglais. Dans le cas des données importées, la langue source des données doit être déduite du jeu de données. Plusieurs solutions sont proposées à l'utilisateur :

- Si le jeu de données n'est qu'en français ou anglais, alors il est possible de spécifier la langue source des données en cochant la langue correspondante sous *Source language* voir section *3.2.4*.
- Si le jeu de données est en français et anglais, alors la langue source peut-être détectée en utilisant l'API detectlanguage.

L'API detectlanguage, dans sa version gratuite, permet de traiter environ un millier de requêtes par jour, et est donc nécessaire seulement lorsque l'utilisateur importe un fichier bilingue.

10. Glossaire

API *:* en informatique, une interface de programmation d'application ou interface de programmation applicative (souvent désignée par le terme API pour Application Programming Interface) est un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels (source : Wikipedia).

CSV *: Comma-separated values*, connu sous le sigle CSV, est un format texte ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules (source : Wikipedia).

dashboard : terme anglais correspondant au français "tableau de bord", dans notre interface, il s'agit du panneau de gauche (voir section 6 du document *Cahier de recette*).

décorateur : On appelle ici décorateur toute chaîne de caractère ou caractère placée avant ou après un mot et servant à changer la sémantique associée à ce mot. Par exemple, un hashtag est caractérisé par un décorateur : le croisillon "#" situé en début de mot.

GIF : Le Graphics Interchange Format (littéralement, "format d'échange d'images"), plus connu sous l'acronyme GIF est un format d'image numérique couramment utilisé sur l'Internet (source : Wikipedia).

IMDb *:* Internet Movie Database (littéralement « Base de données cinématographiques d'Internet »), abrégé en IMDb, est une base de données en ligne sur le cinéma mondial, sur la télévision, et plus secondairement les jeux vidéo (source : Wikipedia).

scraping *:* le web scraping (parfois appelé harvesting) est une technique d'extraction du contenu de sites Web, via un script ou un programme, dans le but de le transformer pour permettre son utilisation dans un autre contexte, par exemple le référencement (source : Wikipedia).

Twitter *:* réseau social de microblogage géré par l'entreprise Twitter Inc. Il permet à un utilisateur d'envoyer gratuitement des micromessages, appelés tweets, sur internet, par messagerie instantanée ou par SMS. Ces messages sont limités à 280 caractères (source : Wikipedia).

webdriver : logiciel permettant de naviguer sur Internet de manière automatisée.