Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Entrées de dictionnaire multilingue pour traducteurs : méthode d'automatisation

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Technologies de la Traduction et Traitement de Données multilingues

par

Kirsten BERLAND

Directrice de mémoire :

Kata Gabor

Encadrante :

Miriam Hamidi

Année universitaire 2021/2022

Remerciements

Je remercie ma directrice de mémoire Kata Gabor, ainsi que les autres professeurs pour leur enseignement et leur accompagnement très appréciés durant mes études à l'Institut National des Langues et Civilisations Orientales.

Je remercie vivement mon encadrante de stage à l'Agence spatiale européenne Miriam Hamidi, qui m'a appris ce qu'est être terminologue, pour sa patience et son écoute tout du long de mon stage.

Enfin, je souhaite remercier mes collègues de stage pour m'avoir accueilli chaleureusement, pour la bonne humeur à chaque rencontre, et pour m'avoir fait vivre une expérience professionnelle exceptionnelle.

RÉSUMÉ

La terminologie est une part importante de la traduction. Elle permet de désambiguïser, mais permet également aux traducteurs de produire un document homogène et cohérent avec le texte de la langue d'origine. Les bases de données terminologiques, dictionnaires multilingues du traducteur, sont des ressources spécialisées dans cette application. Ce mémoire propose une méthode d'automatisation d'entrées de bases terminologiques dont l'approche consiste à extraire les définitions de termes donnés ainsi que la source de ces définitions, et de construire une structure XML compatible avec une base terminologique MultiTerm à partir des données. Les langues traitées ici sont l'anglais, l'allemand et le français, cependant le programme Python qui résulte de ce mémoire peut être adapté à diverses langues et diverses bases de données.

Mots clés : traduction, dictionnaire multilingue, SDL MultiTerm, définition, programmation Python, base de données terminologique, Beautiful Soup, XML, Traitement Automatique des Langues, site internet

Table des matières

Liste des figures

Liste des tableaux

Introduction

La terminologie est aujourd'hui une part importante de la traduction, et cela se reflète dans de nombreux domaines de recherche tels que la traduction automatique neuronale, la création de dictionnaires, et la création d'indexes. [Tomokiyo et al., 2006] ont par exemple lié des dictionnaires au format XML afin de faciliter l'accès des traducteurs aux expressions des documents à traduire et de leurs équivalents.

À travers ce mémoire, nous cherchons à élaborer une méthode d'automatisation de la recherche de définition de termes en allemand, français et anglais pour la traduction assistée par ordinateur. Plus précisément, les définitions extraites seront copiées dans des entrées de base de données terminologique. Le sujet du mémoire repose sur un stage effectué à l'Agence spatiale européenne en tant que terminologue, où la tâche principale était de mettre à jour la base terminologique de l'Agence.

Le premier chapitre de ce mémoire définit les expressions importantes et donne un aperçu succinct de l'état de l'art et de l'historique professionnel qui s'y rapporte. Le second chapitre présente plus en détail le contexte actuel dans lequel s'inscrit le projet du mémoire, et explique la méthode appliquée. Enfin, les résultats et limites observés sont discutés ainsi que les possibles améliorations et propositions de développement.

Première partie

Contexte général

CHAPITRE 1

CONTEXTE GÉNÉRAL

Sommaire

Pour une meilleure compréhension du mémoire, ce chapitre propose des définitions de termes qui apparaîtront au cours de la lecture. Le chapitre pose également le contexte historique des thèmes abordés.

1.1 Définitions

Traduction Le dictionnaire Larousse définit la traduction comme étant l'« action de traduire, de transposer dans une autre langue », ou bien comme l'« énonciation dans une autre langue (ou langue cible) de ce qui a été énoncé dans une langue (la langue source), en conservant les équivalences sémantiques et stylistiques. »

Mémoire de traduction Lorsqu'un traducteur doit travailler sur plusieurs documents d'un client traitant du même sujet, il est fort probable que des phrases ou des tournures de phrases reviennent souvent. Une mémoire de traduction permet au traducteur de garantir l'homogénéité des textes en utilisant la même formulation à chaque fois.

Traduction assistée par ordinateur La norme NF EN ISO 17100 définit la traduction assistée par ordinateur comme une « partie d'un flux de traduction au cours de laquelle diverses applications logicielles sont utilisées pour assister la tâche de traduction par l'homme » [PTS,].

Traduction automatique La traduction automatique apparaît dans le courant des années 1950, bien avant la traduction assistée par ordinateur. Des programmes ou des logiciels s'occupent seuls de la traduction, cependant sa pertinence et son efficacité laissent toujours à désirer, en particulier dans le domaine littéraire.

Terme Selon le Larousse, « terme » peut avoir comme signification dans un contexte linguistique : un « élément entrant en relation avec d'autres » ou bien un « mot considéré dans sa valeur de désignation, en particulier dans un vocabulaire spécialisé ».

Base de données terminologique Une base de données terminologique peut être sous différents formats tels le XML (*eXtensible Markup Language*) et TBX (*TermBase eXchange*). Habituellement, une « entrée » représente un concept et est composée d'au moins une expression, des synonymes et leurs traductions. Diverses informa-

tions sont ajoutées selon l'utilisation de la base de données, par exemple le domaine dont l'expression est spécifique, une définition, des exemples d'usage, la référence où l'expression a été trouvée [Kageura and Marshman, 2019].

Terminologue « Selon Bowker [2015], la terminologie concerne la collection, le traitement, la description et la présentation de termes, c'est-à-dire des éléments lexicaux appartenant à un domaine spécifique », expliquent [Hamidi and Grifoni-Winters, 2021]. Pour l'Agence spatiale européenne, le rôle du terminologue est aujourd'hui d'« entretenir, améliorer et mettre à jour ce qui est devenu la base terminologique de l'Agence, fournir de l'aide concernant la terminologie aux traducteurs et autres employés de l'ASE, et évaluer l'usage quotidien de la terminologie et des outils de traduction assistée par ordinateur dans la Section Traduction de l'ASE. »

1.2 Contexte

Il y a quelques années seulement sont apparus des outils de traduction assistée par ordinateur, ou TAO (*Computer-assisted translation*, CAT, en anglais), pour aider les traducteurs dans leur travail, tels que SDL Trados Studio [1], MemoQ [2] et Déjà Vu [3]. Certains sont spécifiques à la localisation, c'est-à-dire la traduction de logiciels. Ces outils repèrent le contenu textuel des documents à traduire, quel que soit leur format, et peuvent lier des mémoires de traduction, des traducteurs automatiques et des bases de données terminologiques aux projets.

« Le traducteur recherche habituellement des informations sur la syntaxe de mots, le sens contextuel, l'expression figée, la terminologie, le nom propre, le synonyme, l'ontologie, l'usage pragmatique de mots, etc. » expliquent [Tomokiyo et al., 2006]. Les outils de TAO permettent de dégager du temps pour les autres tâches en accélérant la recherche des informations terminologiques. Ils permettent au traducteur de travailler de manière plus fluide, plus pertinente et plus rapide par rapport à la traduction et la recherche manuelle.

Les outils de TAO sont utilisés non seulement par des traducteurs travaillant à leur compte, mais aussi par des équipes de traducteurs en entreprise. Généralement, les mémoires de traduction et les bases terminologiques sont sur des serveurs, disponibles à toute l'équipe, assurant que les expressions soient toujours traduites de la même façon par tous les traducteurs. C'est le cas de l'entreprise SAP [4] [Exel et al., 2020] et de l'Agence spatiale européenne [5] (ASE, *European Space Agency* en anglais), où un stage dans le rôle de terminologue a donné l'inspiration du sujet de ce mémoire.

En terminologie, il s'agit de rendre compte des formulations et mots qui n'apparaissent que dans certains contextes ou dont un sens particulier n'apparaît que dans

1. RWS. Trados Studio. Available at : https://www.trados.com/fr/products/trados-studio/ (Accessed : 31 December 2022).

2. memoQ. Page d'accueil. Available at : https://www.memoq.com/fr (Accessed : 31 October 2022).

3. Atril. Page d'accueil. Available at : https://atril.com/ (Accessed : 31 October 2022).

4. L'entreprise allemande SAP (pour Systemanalyse Programmentwicklung (développement de programmes d'analyse de système)) procure des services et des logiciels d'apprentissage automatique, d'Internet des Objets et d'analytique avancée aux entreprises. Available at : https://www.sap.com/ (Accessed : 9 November 2022).

5. Agence spatiale européenne. Page d'accueil. Available at : https://www.esa.int/ (Accessed : 31 October 2022).

ces domaines. Le plus souvent, ces termes sont des substantifs ou des groupes nominaux, et quelquefois des verbes.

[Kageura and Marshman, 2019] expliquent que « le traitement de termes, en traduction, est impératif pour éviter une mauvaise compréhension et pour respecter les responsabilités sociales. » En effet, fixer le sens d'un mot ou d'une expression et le lier à un contexte permet sa désambiguïsation et un usage constant et cohérent avec les documents d'origine, en plus d'améliorer la qualité de traduction automatique. Ainsi se justifie la création de bases de données terminologiques où une entrée correspond à un concept et indique au minimum le domaine spécifique aux termes de l'entrée.

La base terminologique de l'Agence spatiale européenne est actuellement maintenue par une terminologue grâce au logiciel de gestion terminologique MultiTerm de la société RWS. Les langues des 22 pays membres de l'Agence sont répertoriés dans cette base terminologique, cependant l'allemand, l'anglais et le français sont les langues les plus représentées, suivant la Résolution N°8 « Usage des langues » en annexe du Règlement intérieur du Conseil de l'ASE [European Space Agency, 2010].

L'objectif de ce mémoire est d'élaborer une méthode d'automatisation de la recherche de définition de termes en allemand, français et anglais pour la traduction assistée par ordinateur. Les termes et définitions extraits de cette façon serviront de ressource à une ou plusieurs bases de données terminologiques telles celles possibles sur MultiTerm.

1.3 État de l'art

A ce jour, aucune littérature n'a encore proposé une méthode d'automatisation du travail de terminologue, cependant plusieurs méthodes et projets traitant de la traduction et de dictionnaires multilingues ont été réalisés depuis la fin du XXème siècle. Un point commun des projets présentés ci-après est la préférence des auteurs concernant la structure des entrées de bases de données lexicales : le langage XML est désigné comme le plus efficace par rapport à l'accessibilité des données, que ce soit sur une interface web ou un outil de traduction assistée par ordinateur (TAO).

En 1999, Mangeot propose une interface HTML [1] (*HyperText Markup Language*) se nourrissant de trois dictionnaires multilingues, d'un dictionnaire monolingue et d'une base de données composée d'autres dictionnaires monolingues, dont les langues communes sont l'anglais et le français. L'utilisateur peut de cette façon comparer les articles de dictionnaires avec une seule fenêtre et choisir ce qui convient au mieux à ses besoins. Ce projet s'est conclu sur l'aspiration à former un unique dictionnaire à partir de tous ceux servant de ressource au premier projet.

[Mangeot et al., 2003] se sont inspirés de la conclusion du projet de [Mangeot, 1999] pour créer un dictionnaire public multilingue accessible à partir d'une plateforme internet écrite sur le logiciel ArtStudio (qui utilise le langage Java). Le dictionnaire, appelé Papillon, est modifiable par tous les utilisateurs.

[Tomokiyo et al., 2006] ont continué le projet Papillon en transposant en XML (accompagné d'un document de type définition, DTD) des dictionnaires monolingues du français et de l'anglais pour faciliter la traduction manuelle et automatique. Les deux dictionnaires sont liés par des annotations, c'est-à-dire que lorsque le traducteur tombe sur une expression enregistrée dans un des dictionnaires, le logiciel de tra-

1. Le HTML est un langage de structuration de pages internet.

duction donnera les informations et les équivalences de cette expression dans l'autre langue.

L'article de [Cisse et al., 2008] propose des réflexions à propos de dictionnaires multilingues. Les auteurs ont notamment évoqué l'importance de préciser le domaine d'usage des termes et leur fréquence d'occurrence, en plus du contexte et de la bidirectionnalité ou multidirectionnalité [1] des termes. Les auteurs suggèrent de représenter les fiches terminologiques aux formats XML, TBX ou GENETER et de les intégrer aux systèmes de bases de données grâce à OLIF.

En ce qui concerne l'extraction de contenu de sites internet, [Glez-Peña et al., 2014, Bhoir and Jayamalini, 2021] font la comparaison de différentes bibliothèques Python et de *frameworks*. Le premier article se concentre principalement sur les caractéristiques environnementales et les fonctionnalités de ces outils, tandis que le deuxième article présente une comparaison de la pertinence et de l'efficacité des bibliothèques Scrapy et Beautiful Soup.

Encore dans le sujet de l'extraction, [Peñas et al., 2001] se sont intéressés à l'indexation de ressources pédagogiques en espagnol à partir de termes spécifiques au domaine de ces documents. Le résultat de leur projet est une plateforme HTML : l'utilisateur (sont principalement ciblés les élèves et enseignants du premier et second degré d'enseignement) émet une requête à partir de mots clés et obtient les documents les plus pertinents en réponse à la requête. Le contenu des ressources a été extrait et segmenté en *tokens* (unités graphiques auxquelles sont attribué un rôle sémantique ou un rôle syntaxique) et annoté en partie du discours (par exemple substantif, verbe, adjectif, adverbe, etc.) puis les termes spécifiques ont été extraits suivant des patrons syntaxiques.

En ce qui concerne la traduction, la traduction automatique neuronale est un domaine de recherche très convoité depuis une vingtaine d'années. En 2002 par exemple, les recherches du laboratoire de l'Université de Californie du Sud ont conduit à fonder une entreprise spécialisée, Language Weaver, qui travaille maintenant avec RWS. La littérature traite également ce sujet, [Exel et al., 2020] explorent une méthode de traduction automatique neuronale alimentée d'une base de données terminologique et de mémoires de traduction en allemand, russe et anglais.

Selon [Kageura and Marshman, 2019], la fréquence d'apparition, le nombre de documents dans lesquels un terme apparaît et le terme dans son contexte sont des informations importantes pour le désambiguïser et pour déterminer s'il est utile de l'enregistrer dans une base terminologique. Certaines bases de données concernent les adresses, les numéros de téléphone, etc.

1.4 Conclusion

La terminologie est aujourd'hui une importante part de la traduction, et cela se reflète dans de nombreux domaines tels que la traduction automatique neuronale, et la création de dictionnaires et d'indexes.

Le rôle d'un terminologue est de recenser les termes représentant un même concept dans plusieurs langues, et de leur donner au moins un contexte d'usage, pour aider les traducteurs dans leur travail. Si possible, le sens des termes et la source de chaque terme est également donné. Un stage réalisé personnellement à l'Agence

1. On parle de multidirectionnalité lorsqu'un terme t_A d'une langue A a une équivalence t_B en langue B dont une traduction vers A peut être ta mais aussi un troisième terme s_A

spatiale européenne en tant que terminologue a permis de constater que ce travail était encore manuel, même dans le cadre d'une entreprise de grande envergure.

Ainsi, le projet de ce mémoire est d'élaborer une méthode de création automatique de fiches de base terminologique en passant par l'extraction de définitions. Le corpus utilisé, la méthode appliquée et les résultats obtenus sont présentés dans la partie suivante.

Deuxième partie

Expérimentations

CHAPITRE 2

MÉTHODE

Sommaire

Ce chapitre du mémoire décrit le contexte dans lequel s'inscrit le sujet, et explique les tâches réalisées au cours de la création du programme Python. Les résultats de ce programme seront illustrés à travers quelques exemples dans le chapitre 3. S'ensuit une discussion sur ces résultats ainsi que des propositions de développement de cette automatisation de création d'entrées de base terminologique.

2.1 Contexte

L'Agence spatiale européenne est une organisation multilingue, comme l'indique son nom : il est possible d'entendre et de lire chacune des langues des (en 2022) 22 États membres dans les bâtiments de l'Agence, et même plus. C'est pourquoi une section de son organisation est dédiée à la traduction : la Division Services linguistiques et procès-verbaux qui appartient au Cabinet du Directeur général. Dans cette division, environ 20 000 pages sont traduites chaque année, d'après les chiffres de 2018.

Un des métiers dans cette division est celui de terminologue, dont le rôle est défini dans le sous-chapitre Définitions. Dans l'objectif d'aider les traducteurs, le terminologue crée et met à jour des « fiches » de termes où des expressions sont définies, mises en contextes et accompagnées de leurs équivalents dans d'autres langues.

Une expression qui apparaît souvent dans un document et n'est pas traduit à chaque fois de la même façon peut rendre les textes sources et cibles inconsistants et perturber le lecteur. Créer une fiche terminologique pour un terme permet ainsi la cohésion des textes, surtout lorsque plusieurs traducteurs travaillent sur le même document, et permet d'éviter les fautes d'orthographe ainsi que les ambiguïtés.

Selon [Hamidi and Grifoni-Winters, 2021], les termes traités par un terminologue à l'ASE sont choisis par les traducteurs suivant la règle de Kara Warburton décrite dans le chapitre « Managing terminology in commercial environments » du *Handbook of Terminology* [2] : en fonction de la fréquence d'utilisation, du plongement, de la visibilité et de la difficulté de traduction du terme candidat.

2. Warburton K. (2015). Managing terminology in commercial environments. in : H. J. Kockaert, F.

Dans les années 1990, les termes et informations sur ces termes, qui alimentent aujourd'hui les bases terminologiques, étaient écrits à la main sur des cartes. Cela apportait la contrainte de place physique, une carte ne permettait pas de donner suffisamment d'information sur les termes qu'elle décrivait.

La fin des années 1990 marque l'arrivée de MultiTerm à l'ASE, distribué par SDL (maintenant nommé RWS), un outil de bases de données. Les cartes de terminologie sont alors transcrites dans une base terminologique intitulée « *ESA Termbase* ». Ce n'est qu'en 2009 que la base terminologique devient accessible à tous les membres de l'ASE à travers un serveur, principalement dans le but d'aider les nouveaux venus à comprendre les expressions spécifiques de leur domaine et à les employer correctement.

Jusqu'alors, la base terminologique n'était pas liée à l'environnement de traduction, c'est-à-dire que lorsqu'un traducteur tombait sur une expression qui lui était inconnue ou dont il n'était pas sûr de la traduction, il devait la chercher sur une autre plateforme (MultiTerm) et cela pouvait prendre du temps. Ainsi, « *ESA Termbase* » fût connectée au logiciel d'aide à la traduction SDL Trados Studio, également de RWS, en 2017. Le logiciel repère les termes enregistrés dans la base de données dans les documents sur lesquels travaillent les traducteurs et propose les traductions possibles. Grâce à une fenêtre de paramétrage, il est possible de déterminer quelles informations rendre visible sur l'interface. La Figure 2.1 est un exemple de paramétrage des champs à rendre visible sur SDL Trados Studio dans le contexte de la base « *ESA Termbase* ». Les champs proposés dans ces paramètres sont ceux décrits Figure 2.3.

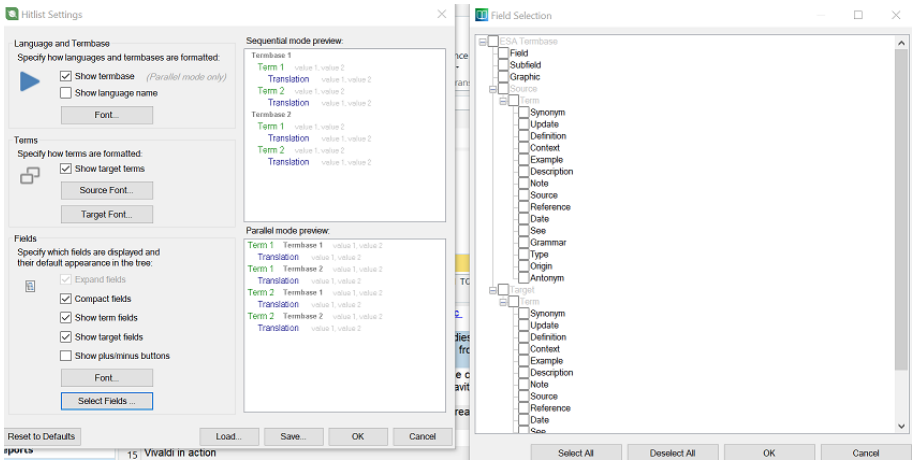


FIGURE 2.1 – Options des champs à afficher sur SDL Trados Studio

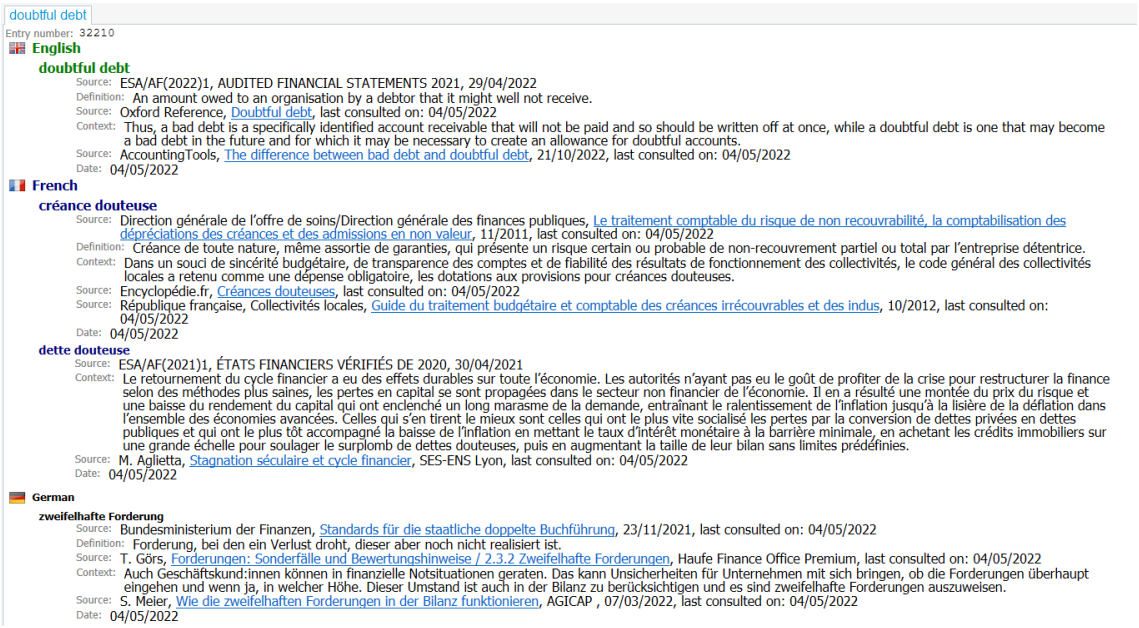
En octobre 2022, la base terminologique de l'Agence spatiale européenne comptait plus de 32 000 entrées. Une des entrées est représentée sur la Figure 2.2 : « *doubtful debt* », ou « créance douteuse » en français et « *zweifelhafte Forderung* » en allemand. Y sont donnés les champs de référence, de définition, de contexte et de date.

La structure de la base a été déterminée à sa création et ne peut plus être modifiée, à moins d'en créer une autre ou d'en supprimer les entrées existantes. La structure actuelle est décrite sur les Figures 2.3 et 2.4. Il est possible de rendre un champ obligatoire, de limiter un champ à une unique utilisation dans une fiche, et de

Steurs (Eds.), Handbook of Terminology, Vol. 1. Amsterdam/Philadelphia : John Benjamins Publishing Company, pp. 360-392.

déterminer le type du contenu des champs. Dans le cas de l'« *ESA Termbase* », toutes les zones peuvent être données plusieurs fois et aucune n'est obligatoire par défaut. Cela apporte un avantage lorsque le terminologue souhaite ajouter une entrée à la base terminologique en y écrivant seulement le terme, car l'expression vient juste d'être inventée par exemple.

Parmi les types de champs sont proposés le texte simple, la liste, le fichier, la date, etc. La base de données terminologique de l'ASE permet d'inclure un fichier dans le champ « Graphic », pour illustrer les termes de la fiche et apporter une meilleure compréhension de ceux-ci.


doubtful debt

Entry number: 32210

English

doubtful debt

Source: ESA/AF(2022)1, AUDITED FINANCIAL STATEMENTS 2021, 29/04/2022
Definition: An amount owed to an organisation by a debtor that it might well not receive.
Source: Oxford Reference, Doubtful debt, last consulted on: 04/05/2022
Context: Thus, a bad debt is a specifically identified account receivable that will not be paid and so should be written off at once, while a doubtful debt is one that may become a bad debt in the future and for which it may be necessary to create an allowance for doubtful accounts.
Source: AccountingTools, The difference between bad debt and doubtful debt, 21/10/2022, last consulted on: 04/05/2022
Date: 04/05/2022

French

créance douteuse

Source: Direction générale de l'offre de soins/Direction générale des finances publiques, Le traitement comptable du risque de non recouvrabilité, la comptabilisation des dépréciations des créances et des admissions en non valeur, 11/2011, last consulted on: 04/05/2022
Definition: Créance de toute nature, même assortie de garanties, qui présente un risque certain ou probable de non-recouvrement partiel ou total par l'entreprise détentrice.
Context: Dans un souci de sincérité budgétaire, de transparence des comptes et de fiabilité des résultats de fonctionnement des collectivités, le code général des collectivités locales a retenu comme une dépense obligatoire, les dotations aux provisions pour créances douteuses.
Source: Encyclopédie.fr, Créances douteuses, last consulted on: 04/05/2022
Source: République française, Collectivités locales, Guide du traitement budgétaire et comptable des créances irrécouvrables et des indus, 10/2012, last consulted on: 04/05/2022
Date: 04/05/2022

dette douteuse

Source: ESA/AF(2021)1, ÉTATS FINANCIERS VÉRIFIÉS DE 2020, 30/04/2021
Context: Le retournement du cycle financier a eu des effets durables sur toute l'économie. Les autorités n'ayant pas eu le goût de profiter de la crise pour restructurer la finance selon des méthodes plus saines, les pertes en capital se sont propagées dans le secteur non financier de l'économie. Il en a résulté une montée du prix du risque et une baisse du rendement du capital qui ont enclenché un long marasme de la demande, entraînant le ralentissement de l'inflation jusqu'à la lisière de la déflation dans l'ensemble des économies avancées. Celles qui s'en tirent le mieux sont celles qui ont le plus vite socialisé les pertes par la conversion de dettes privées en dettes publiques et qui ont le plus tôt accompagné la baisse de l'inflation en mettant le taux d'intérêt monétaire à la barrière minimale, en achetant les crédits immobiliers sur une grande échelle pour soulager le surplomb de dettes douteuses, puis en augmentant la taille de leur bilan sans limites prédéfinies.
Source: M. Aglietta, Stagnation séculaire et cycle financier, SES-ENS Lyon, last consulted on: 04/05/2022
Date: 04/05/2022

German

zweifelhafte Forderung

Source: Bundesministerium der Finanzen, Standards für die staatliche doppelte Buchführung, 23/11/2021, last consulted on: 04/05/2022
Definition: Forderung, bei den ein Verlust droht, dieser aber noch nicht realisiert ist.
Source: T. Görs, Forderungen: Sonderfälle und Bewertungshinweise / 2.3.2 Zweifelhafte Forderungen, Haufe Finance Office Premium, last consulted on: 04/05/2022
Context: Auch Geschäftskund:innen können in finanzielle Notsituationen geraten. Das kann Unsicherheiten für Unternehmen mit sich bringen, ob die Forderungen überhaupt eingehen und wenn ja, in welcher Höhe. Dieser Umstand ist auch in der Bilanz zu berücksichtigen und es sind zweifelhafte Forderungen auszuweisen.
Source: S. Meier, Wie die zweifelhaften Forderungen in der Bilanz funktionieren, AGICAP , 07/03/2022, last consulted on: 04/05/2022
Date: 04/05/2022

FIGURE 2.2 – Entrée « doubtful debt » de la base terminologique de l'ASE

Entry Structure	Mandatory	Multiple
Entry level		
Field		•
Subfield		•
Graphic		•
Index level		
Term level		
Synonym		•
Update		•
Definition		•
Context		•
Example		•
Description		•
Note		•
Source		•
Reference		•
Date		•
See		•
Grammar		•
Type		•
Origin		•
Antonym		•

FIGURE 2.3 – Structure de la base terminologie de l'ASE

Descriptive Fields

Name	History	Type	Picklist Values
Antonym		Text	
Context		Text	
Date		Text	
Definition		Text	
Description		Text	
Example		Text	
Field		Picklist	Astronautics | Astronomy | Astrophysics | Chemistry | Climatology | Corporate communication | Corporate governance | Corporate organisation | Economics | Education | Engineering | Environment | European Union | Finance | General | Geosciences | Ground segment | Industrial matters | Information systems | Internal use | International organisations | International relations | Legal matters | Life sciences | Linguistics/translation | Management | Material sciences | Mathematics | Navigation | Physics | Power | Propulsion | R&D | Safety | Security & Defence | Space agencies | Space segment | Space stations | Space transportation | Spacecraft | Telecommunications | Biology
Grammar		Picklist	n. | n.pl. | n.f. | n.f.pl. | n.m. | n.m.pl. | n.nt. | n.nt.pl. | adj. | adv. | vb.
Graphic		Multimedia File	
Materials Science		Text	
Note		Text	
Origin		Picklist	CZ | DA | DE | EN | ES | FI | FL | FR | IT | JA | NL | NO | None | PO | PT | RU | SV | check
Reference		Text	
See		Text	
Source		Text	
Subfield		Picklist	Accounting | Advisory bodies | Aerodynamics | Antennas | Ariane | Astrobiology | Atmosphere | ATV | Budget | Civil engineering | Commercialisation | Cooperation | Cosmology | Cryogenics | CTV | Data management | Data relay | Delegate bodies | Education | Electrical engineering | Electronic engineering | External audit | Financial planning | Fluid sciences | Future launchers | Geophysics | Hermes | Hydrology | Intellectual property | Internal audit | Launchers | Manned flight | Mechanical engineering | Medicine | Meteorology | Metrology | Ministerial Council | Nuclear physics | Oceanography | Optics | Orbitography | Outreach | Particle physics | Personnel | Planetary science | Plasma physics | Procurement | Project management | Quality management | Radars | Risk management | Robotics | Signalling | Solid-state physics | Soyuz | Space debris | Space probes | Space weather | Standardisation | Sustainable development | Technology transfer | Telemetry | Telescopes | Testing | Thermal engineering | Transport | Vega | Project planning | Space Rider | IXV | ...
Synonym		Text	
Type		Picklist	Abbreviation
Update		Text	

FIGURE 2.4 – Définition des champs de la base terminologique de l'ASE

[Tomokiyo et al., 2006] ont déterminé que la « zone pour noter l'auteur de la lexie, la date de description et révision et la zone de l'historique » étaient importants pour les utilisateurs de bases terminologiques. De nos jours, ces champs sont automatisés sur le logiciel ou la plateforme internet de MultiTerm lors de la création ou modification d'une entrée. En fonction du paramètre d'affichage, ces champs peuvent être cachés ou visibles, la préférence d'affichage des traducteurs influe donc sur ce qu'ils voient. Ainsi, malgré le fait que les dates de création et de modification d'une entrée soient enregistrées automatiquement, les champs « Date » et « Update » (mise à jour) ont été définis pour la base terminologique de l'Agence spatiale européenne pour les besoins des traducteurs.

Le projet sur lequel se concentre ce mémoire se situe dans le contexte d'un terminologue souhaitant alléger sa charge de travail et gagner en temps lors de la création ou modification d'entrées d'une base de données terminologique. Les langues de cette base sont l'anglais, l'allemand et le français, semblable à la base de données de l'Agence spatiale européenne.

2.2 Corpus

Le corpus qui alimente le projet de ce mémoire est composé de plusieurs sites de dictionnaires anglais, allemands et français en ligne, qui furent interrogés sans modification préalable (sans « nettoyage »). Ces sites sont accessibles librement et gratuitement à tous (hormis quelques pages où l'inscription est de rigueur), et sont en principe fiables dans leurs sources. Dans l'objectif du projet, une petite partie seulement est utilisée, extraite de certaines pages donnant lieu d'entrées de dictionnaire :

les définitions et le titre des pages qui sont extraits par le script Python résultant de ce projet. Le contenu des sites est expliqué en détail dans la partie Méthode.

Les dictionnaires français choisis pour le corpus sont le dictionnaire monolingue *Larousse* et le portail lexical du Centre National de Ressources Textuelles et Lexicales (CNRTL). Le premier est un site de la société des Editions Larousse dont le dictionnaire est disponible à l'adresse https://www.larousse.fr/dictionnaires/francais-monolingue. Le second est un « projet mené par le laboratoire ATILF [1] dont l'objectif est de valoriser des ressources linguistiques issues de différents projets de recherche au sein d'un portail unique. » Il regroupe dans sa section lexicographie (disponible à travers ce lien : https://www.cnrtl.fr/definition/) six bases de données de la langue française, certaines plus actuelles que d'autres :

1. Le Trésor de la Langue Française informatisé (TLFi),
2. Les dictionnaires de l'Académie Française (4ème, 8ème et 9ème édition),
3. La Base de données lexicographiques panfrancophone de l'université Laval de Québec,
4. La Base Historique du Vocabulaire Français du laboratoire ATILF,
5. Le Dictionnaire du Moyen Français (1330 - 1500) du laboratoire ATILF,
6. Le Du Cange (Moyen Âge) de l'École Nationale des Chartes.

Du côté des sites allemands, l'Académie des sciences de Berlin-Brandebourg a réalisé un projet qui a mené au Dictionnaire numérique de la langue allemande (DWDS, *Digitales Wörterbuch der deutschen Sprache*), il compte aujourd'hui plus de 230 000 mots [2]. Le DWDS rassemble un grand nombre de dictionnaires et présente pour chaque mot, si disponible : sa partie du discours, sa prononciation, des définitions et des exemples, et également des statistiques sur son usage au cours du temps, ainsi que son réseau lexical.

Le second dictionnaire allemand choisi pour ce projet est le *Duden*, il est aujourd'hui publié par la maison d'édition Cornelsen Verlag et comprend plus de 148 000 articles. Le dictionnaire *Duden*, écrit par Konrad Duden, fut publié pour la première fois en 1880.

Le *Cambridge Dictionary* est une plateforme de dictionnaires monolingues anglais et bilingues gérée par la maison d'édition Presses universitaires de Cambridge (Cambridge University Press, CUP) rattachée à l'Université de Cambridge en Angleterre, accessible sur internet depuis 1999. Le corpus qui alimente le *Cambridge Dictionary* compte aujourd'hui près de 2 milliards de mots. La section du *Cambridge Dictionary* étudiée ici rassemble les trois dictionnaires monolingues suivants : le *Cambridge Advanced Learner's Dictionary*, le *Cambridge Academic Content Dictionary*, et le *Cambridge Business English Dictionary*. Pour chaque entrée de dictionnaire sont présentés pour l'anglais britannique et l'anglais américain des définitions, des supports audios pour la prononciation du mot ou de l'expression, des exemples, les synonymes et antonymes, et des expressions associées au mot expliqué.

Collinsdictionary.com, de l'édition Collins, est également une plateforme de dictionnaires monolingue et bilingues dont la ressource compte environ 4,5 milliards de

1. Laboratoire de recherche public à Nancy. Analyse et Traitement Informatique de la Langue Française. Available at : https://www.atilf.fr/ (Accessed : 9 November 2022).
2. Berlin-Brandenburgische Akademie der Wissenschaften. Wörterbücher im DWDS. *Digitales Wörterbuch der deutschen Sprache*. Available at : https://www.dwds.de/d/woerterbuecher (Accessed : 9 November 2022).

mots, en ligne depuis 2007. On trouve sur le site également un thésaurus, un dictionnaire pédagogique et une partie dédiée à l'apprentissage de la grammaire anglaise. Lors de sa création fut intégré le *Collins English Dictionary* dont la première édition (papier) est parue en 1979.

2.3 Méthode

Le produit final du projet présenté dans ce mémoire est un script Python qui, à partir de liens menant à des sites de dictionnaires, de termes et de l'indication de leur langue, crée un fichier au format XML conforme à un type de base de données terminologique. Le script, à la fin du mémoire en annexe, se lance avec la commande `python search_definitions.py`.

En outre de la description de l'environnement de programmation dans lequel le projet a été effectué, chaque étape de ce programme Python sera décrite dans cette sous-partie du chapitre Expérimentations.

2.3.1 Environnement informatique

Le projet s'est déroulé avec l'environnement informatique suivant :
- Un terminal de commande Ubuntu (version 18),
- La version 3.6 du langage de programmation Python,
- Des bibliothèques et modules Python :
  - **bs4** (Beautiful Soup version 4.9) sert à extraire le contenu de structures HMTL et XML qui composent en particulier les sites internet, à l'aide d'un analyseur syntaxique (*parser*) ;
  - **urllib.request** permet à un code Python d'accéder au contenu de sites internet à travers leur adresse, ou URL (*Uniform Resource Locators*) ;
  - **xml.etree.ElementTree** est une interface de programmation d'application (API, *Application Programming Interface*) qui permet d'analyser et de créer des documents XML ;
  - **re** (de l'anglais *regular expressions*) fournit des fonctions de traitement de texte basé sur des expressions régulières, c'est-à-dire des séquences de caractères auxquels sont appliqués des opérations et des contraintes. Les fonctions nécessitent deux ou trois paramètres selon l'opération : l'expression à traiter, son remplacement et le corpus dans laquelle se trouve l'expression ;
  - **time** rassemble des fonctions qui se rapportent au temps, il est par exemple possible de représenter la date et l'heure actuelles de diverses façons et de calculer une durée.

En dehors de Beautiful Soup, les autres bibliothèques et modules sont installés par défaut. Beautiful Soup est installée avec la commande suivante sur un terminal : `pip install beautifulsoup4`.

Il existe une bibliothèque d'extraction de contenu de sites Python autre que Beautiful Soup, Scrapy. Néanmoins, Beautiful Soup est plus rapide que celle-ci et utilise la navigation par balise, ce qui rend la recherche du contenu à extraire plus précise et nécessite moins de ligne de code. La navigation et l'extraction de site internet est également possible grâce aux commandes UNIX (curl/wget, grep, sed, cut, paste, awk), mais pour les mêmes raisons que pour Scrapy, la bibliothèque Beautiful Soup a été retenue pour le script.

2.3.2 Algorithme

Premièrement vient la demande du code ISO de la langue ou des langues que l'utilisateur du programme Python souhaite traiter. Il peut se trouver sur Wikipédia [Collectif, 2022b]. Pour chaque langue donnée en « entrée » au programme sont également demandés les termes. Si le code d'une langue n'est pas conforme aux codes disponibles, une boucle interroge l'utilisateur jusqu'à l'obtention d'un code qui soit dans la liste de ceux permis par le script.

Deuxièmement, chaque terme est mis de côté, et une version adaptée aux adresses de sites internet est créée en remplaçant les caractères spéciaux (les lettres diacritées et les espaces) par leur « encodage pourcent » respectif qui correspond au pourcentage '%' suivi du code hexadécimal du caractère (source des codes hexadécimaux : *UTF8 chartable*). La diacritée 'è' par exemple, est ainsi remplacée par '%c3%a8'. Cette étape va permettre, pour la plupart des termes, d'obtenir les liens des entrées dictionnairiques en ligne qui correspondent au terme.

La troisième étape consiste en la création d'un dictionnaire Python où sont enregistrés les liens de la page de définition du terme donné de chaque dictionnaire. Ceci est réalisé dans une fonction nommée `addDicoLink()`. Elle prend en premier paramètre le contenu d'un fichier dans lequel est écrit un URL par ligne, et en second paramètre le terme transformé avec l'« encodage pourcent », puis la fonction concatène l'adresse et le terme.

Chaîne de caractères d'origine	**Chaîne de caractères après transformation**
rêve	r%c3%aave
https://www.larousse.fr/dictionnaires/francais/	https://www.larousse.fr/dictionnaires/francais/r%c3%aave
https://www.cnrtl.fr/definition/	https://www.cnrtl.fr/definition/r%c3%aave

TABLE 2.1 – Exemple avant-après des étapes 2 et 3

Ensuite, le programme essaie d'ouvrir les liens ainsi obtenus grâce au module urllib.request (début de la fonction `getTermInfo()`). Dans le cas où une erreur survient, l'utilisateur est prévenu par un message indiquant la nature du blocage. Le type d'une erreur en lien avec l'*Hypertext Transfer Protocol* [1] (HTTP, traduit littéralement par « protocole de transfert hypertexte ») est associé à un code de trois chiffres dont les plus connus sont 403 (l'accès à la page est interdit), 404 (la page n'existe plus ou est mal écrite) et 504 (le serveur ne réagit pas à la demande). L'article « Liste des codes HTTP » sur Wikipéda donne un aperçu de ces codes.

Une des raisons pour laquelle le programme n'aurait pas accès à un site dépend du droit d'auteur de ce site. Un site sait en général quand un utilisateur ouvre une de ses pages de manière non habituelle, non pas à travers un navigateur web, mais à l'aide d'un « robot ». Il consulte alors le fichier *robots.txt* disponible à la racine du site (https://www.larousse.fr/robots.txt, https://www.duden.de/robots.txt, etc.), et vérifie les droits : les pages auxquelles les robots marqués

1. D'après le Larousse, HTTP est un « protocole de communication entre internautes et serveurs du Web, pour la consultation et le transfert de documents de type hypermédia. »

d'un identifiant particulier sont interdit d'accès. Cet identifiant, ou User-agent, est défini dans le script Python et utilisé avec urllib.request.

Une fois le lien ouvert, le programme parcourt le *Document Object Model* (DOM) des pages web avec le *parser* **lxml** lié à Beautiful Soup pour récupérer le contenu voulu, c'est-à-dire les définitions, le titre de la page, et le nom du site. Le DOM est défini par [Wang, 2013] comme étant une « interface de programmation d'application (API) qui permet l'accès et la modification du format et du contenu de pages web enregistrées », il s'agit en général de la structure balistique des pages qui sont ici en HTML.

Le nom du site se trouve, pour certains des dictionnaires du corpus, dans une balise ayant un attribut `property="og:site_name"`, et également dans la balise « title » (titre) pour la majorité des dictionnaires. Les définitions sont fréquemment sous forme de liste (balise « li ») et/ou dans une balise « span » avec un attribut « class » (classe) associé à cette balise indiquant qu'il s'agit d'une définition. Ces informations ont été obtenues en étudiant le code source de quelques pages de notre corpus. Le contenu des balises listées ci-dessus est extrait à l'aide de Beautiful Soup et d'expressions régulières.

```
<conceptGrp>
    <concept></concept>
    <languageGrp>
        <language lang="EN" type="English" />
        <termGrp>
            <term></term>
            <descripGrp>
                <descrip type="Definition"></descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Source"></descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Date"></descrip>
            </descripGrp>
        </termGrp>
    </languageGrp>
    <languageGrp>
        <language lang="FR" type="French" />
        <termGrp>
            <term></term>
            <descripGrp>
                <descrip type="Definition"></descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Source"></descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Date"></descrip>
            </descripGrp>
        </termGrp>
    </languageGrp>
    <languageGrp>
        <language lang="DE" type="German" />
        <termGrp>
            <term></term>
            <descripGrp>
                <descrip type="Definition"></descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Source"></descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Date"></descrip>
            </descripGrp>
        </termGrp>
    </languageGrp>
</conceptGrp>
```

FIGURE 2.5 – Structure de base d'une entrée produite par le script

Un terme peut être polysémique, c'est-à-dire avoir plusieurs significations, et cela influe sur la difficulté de la traduction car un terme peut changer d'équivalent en fonction de son contexte. Par conséquent, les dictionnaires présentent une liste de définitions portant sur des thèmes différents, mais ces définitions sont quelques fois très similaires. De plus, afin de désambiguïser les termes, les entrées de base terminologiques peuvent être distinctes selon la définition et les ou l'équivalence dans les autres langues malgré qu'elles traitent du même terme. Pour ces raisons, le traducteur ou terminologue, utilisateur du programme, a donc le choix du dictionnaire et d'une définition grâce aux fonctions `chooseDictionary()` et `chooseDefinition()`.

Finalement, la définition et la référence (citation du dictionnaire) de chaque terme enregistrées, un fichier XML compatible avec MultiTerm est créé (voir Figure 2.5) pour le concept. La structure de ce fichier est inspirée des champs de la base terminologique de l'Agence spatiale européenne.

L'importance d'un document type définition (DTD, *Document Type Definition*) afin de comprendre une structure XML, est évoquée dans les articles de [Mangeot, 1999, Tomokiyo et al., 2006]. Ainsi, un DTD qui correspond à l'arbre XML construit pour la fiche terminologique a été écrit (voir l'Annexe A).

CHAPITRE 3

RÉSULTATS

Sommaire

3.1 Résultats

3.1.1 Extraction des définitions

L'évaluation des définitions candidates est divisée ici par langue.

Dictionnaires de français

Les termes français « ethnographie », « curateur », « inaugurer », « phosphorer », « agriculture raisonnée », « mécène », « conservateur », « diaspora », « long-métrage » sont tirés des articles en ligne « Les artistes, ces créateurs de lieux en Afrique » et « Sara Sadik, défricheuse de banlieue » de R. Azimi dans Le Monde. L'évaluation du programme se fait avec ces termes et s'appuie sur la mesure du rappel et de la précision, car propices à l'évaluation de classification binaire (définition, pas définition). Les formules du rappel et de la précision sont les suivantes :

$$Précision = \frac{nb\ vrais\ positifs}{nb\ vrais\ positifs\ +\ nb\ faux\ positifs} \tag{3.1}$$

$$Rappel = \frac{nb\ vrais\ positifs}{nb\ vrais\ positifs\ +\ nb\ faux\ négatifs} \tag{3.2}$$

Les faux positifs, faux négatifs et vrais positifs de ces neuf termes, résultant de la comparaison entre les définitions visibles sur les sites et proposées par le script, sont présentés dans le Tableau 3.2. La limite du programme étant de 10 propositions par site pour éviter de perturber l'utilisateur avec un amoncellement d'information, le comptage des définitions donné dans le tableau s'arrête conformément aux définitions candidates.

Selon le Tableau 3.1, le script n'a renvoyé que deux définitions candidates du site CNRTL pour le terme « curateur », alors que l'on peut en compter neuf sur la page du site. En effet, les définitions sont séparées en deux sections sur le navigateur et le contenu de la deuxième n'est pas disponible sur le code source de la page, alors que le programme ne consulte que ce code source.

	Nombre de définitions sur le site		**Nombre de définitions candidates**	
Terme	**CNRTL**	**Larousse**	**CNRTL**	**Larousse**
ethnographie	1	1	1	1
curateur	9	4	2	4
inaugurer	7	4	7	8
phosphorer	3	1	3	2
agriculture raisonnée	0	1	0	2
mécène	1	1	2	1
conservateur	5	6	7	10
diaspora	3	2	3	3
long-métrage	0	1	0	1
Total	29	21	25	32

TABLE 3.1 – Comparaison du nombre de définitions par dictionnaire pour le français

Les vrais positifs sont les candidats du programme qui sont effectivement des définitions pour le terme donné, tandis que les faux négatifs sont les définitions qui n'ont pas été extraites. Les faux positifs sont les propositions qui ne sont pas des définitions, ou les définitions du terme. Les vrais négatifs n'ont pas été calculés par contrainte de temps. Il s'agit notamment des exemples et des synonymes, des définitions d'autres termes et expressions affichés sur les pages web, et des hyperliens.

		Réelle définition		
		Positif	**Négatif**	**Total**
Candidat	**Positif**	41	15	56
	Négatif	6	/	6
	Total	47	15	62

TABLE 3.2 – Matrice de confusion à partir des définitions et des propositions de définition (candidat) pour le français

	Rappel		**Précision**	
	CNRTL	**Larousse**	**CNRTL**	**Larousse**
	0,807	0,952	0,875	0,625
Moyenne	0,872		0,732	

TABLE 3.3 – Rappel et précision pour les neuf termes en français

Avec plus de 70%, les mesures du rappel et de la précision sont élevées et reflètent l'efficacité et la pertinence de l'extraction de définitions par Beautiful Soup en ce qui concerne les dictionnaires de français. D'après le Tableau 3.3, la précision est plus faible pour le *Larousse* que pour le CNRTL, cela s'explique par les exemples qui sont considérés comme des définitions par le programme Python du fait que le corpus n'ait pas été nettoyé et que les balises entourant les exemples sont très similaires à celles des définitions. Malgré tout, l'utilisateur obtiendra généralement toutes, ou presque toutes les définitions des deux sites.

Dictionnaires d'allemand

L'évaluation du programme sur des termes en allemand a été réalisé avec les équivalents des termes français mentionnés plus haut.

On remarque que pour la majorité des termes, le script n'a pas trouvé de définition à partir du *Duden*. Et pour cause, il s'avère que le script ne couvre pas l'extraction de *Duden* dans le cas d'une seule définition sur le site, mais ne traite que les listes. Cela se remarque aussi dans la mesure de la précision et du rappel où les deux tiers des termes n'avaient qu'une définition *Duden*.

Les moyennes du rappel et de la précision dans le cas des sites allemands sont toutes deux égales à 0,806, à quelques millièmes de décimales près. Malgré la faible performance pour l'autre dictionnaire, le traitement de DWDS présente de très bons résultats.

	Nombre de définitions sur le site		**Nombre de définitions candidates**	
Terme	**DWDS**	**Duden**	**DWDS**	**Duden**
Ethnografie	2	1	2	0
Betreuer	1	1	1	0
einweihen	2	3	2	3
brüten	5	5	7	5
nachhaltige Landwirtschaft	0	0	0	0
Mäzen	1	1	0	0
Verwalter	4	1	8	0
Diaspora	1	2	1	2
Spielfilm	1	1	1	0
Total	17	15	22	10

TABLE 3.4 – Comparaison du nombre de définitions par dictionnaire pour l'allemand

		Réelle définition		
		Positif	**Négatif**	**Total**
Candidat	**Positif**	25	6	31
	Négatif	6	/	6
	Total	31	6	37

TABLE 3.5 – Matrice de confusion à partir des définitions et des propositions de définition (candidat) pour l'allemand

	Rappel		**Précision**	
	DWDS	**Duden**	**DWDS**	**Duden**
	0,937	0,666	0,714	1
Moyenne	0,806		0,806	

TABLE 3.6 – Rappel et précision pour les neuf termes en allemand

Dictionnaires anglais

Le dictionnaire de Collins ayant interdit l'accès du programme à son site au bout de quelques tentatives de connexion, cette section de l'évaluation se concentre sur le dictionnaire de Cambridge. Encore une fois, une équivalence de chaque terme en français est étudiée en anglais pour les mesures de précision et de rappel.

Terme	**Nombre de définitions**	**Nombre de définitions**
ethnography	1	1
guardian	6	5
inaugurate	6	5
beaver away	1	1
integrated farming	0	0
sponsor	10	10
curator	2	2
diaspora	5	5
feature-length film	0	0
Total	31	29

TABLE 3.7 – Comparaison du nombre de définitions par dictionnaire pour l'anglais

		Réelle définition				
		Positif	**Négatif**	**Total**	**Rappel**	**Précision**
Candidat	**Positif**	29	0	29	0,935	1
	Négatif	2	/	2		
	Total	31	0	31		

TABLE 3.8 – Matrice de confusion à partir des définitions et des propositions de définition (candidat) pour l'anglais

Le Tableau 3.7 permet d'observer une constance : le script trouve le même nombre de définition que listé dans le site pour quasiment chaque terme. Ceci est appuyé par les mesures très élevée de précision et de rappel donnant de plus l'information que tous les candidats relevés sont bien des définitions relatives au terme donné.

3.1.2 Accéder aux sites

A la suite de plusieurs tests du script sur une même page du *Collins English Dictionary*, l'accès du programme à la page fut bloqué pour respecter les droits d'auteur. Comme mentionné plus haut en effet, les robots ne peuvent se connecter à une page de ce site qu'une seule fois. Un message dont le suivant est un exemple apparaissait pour cette raison : HTTPError: 403 for https://www.collinsdictionary.com/dictionary/english/. La structure du message est celle fixée ligne 146 du script.

```
Date: 15/11/2022
Language or languages separated by ';' [EN/FR/DE]: EN;FR
EN term or terms separated by ';': tourism


Charging dictionary links...
Getting term definitions...


Dictionaries searched.

0. TOURISM | English meaning - Cambridge Dictionary, https://dictionary.cambridge.org/dictiona
ry/english/tourism
        0. The business of providing services such as transport, places to stay, or entertainm
ent for people who are on holiday.
        1. The business of providing services, such as transportation, places to stay, or ente
rtainment, for tourists
        2. The business of providing services such as transport, places to stay, or entertainm
ent for people on holiday.

Number of the reference to keep: 0
Number of the definition to keep: 0
FR term or terms separated by ';': tourisme


Charging dictionary links...
Getting term definitions...



Dictionaries searched.

0. Définitions : tourisme - Dictionnaire de français Larousse, https://www.larousse.fr/diction
naires/francais/tourisme
        0. Action de voyager, de visiter un site pour son plaisir.
        1. Ensemble des activités, des techniques mises en œuvre pour les voyages et les séjou
rs d'agrément.

1. TOURISME : Définition de TOURISME, https://www.cnrtl.fr/definition/tourisme
        0. Activité d'une personne qui voyage pour son agrément, visite une région, un pays, u
n continent autre que le sien, pour satisfaire sa curiosité, son goût de l'aventure et de la d
écouverte, son désir d'enrichir son expérience et sa culture.
        1. Exercer une activité en dilettante, en dehors de tout professionnalisme.
        2. Ensemble des activités touristiques (séjours, voyages d'agrément).
        3. Tourisme de sports d'hiver.
        4. Forme de tourisme rural.
        5. Industrie se consacrant à tous les besoins engendrés par les déplacements des touri
stes (moyens de communication, transports, structures d'accueil, aménagement des sites) et à t
outes les questions d'ordre économique, juridique, financier, social que soulève ce domaine (a
ccueil des personnes, apport de devises, balance commerciale, statistiques, etc.) organisé, st
ructuré et réglementé au niveau national et régional.
        6. Organisme chargé de renseigner les touristes sur toutes les questions qu'ils se pos
ent (hébergement, visites, transports, etc.).
        7. Organisme dont la vocation première est de promouvoir et de vendre les produits lié
s au tourisme.
        8. Hôtel répondant à certains critères de confort, le classant en hôtel de grand touri
sme (3 étoiles), hôtel de tourisme (2 étoiles) ou hôtel de moyen tourisme (1 étoile).
        9. Voiture, avion que l'on utilise pour son usage personnel, dans un but privé et non
collectif.

Number of the reference to keep: 1
Number of the definition to keep: 2
Entry number [type - if none]: -
Writting XML file...
XML file ready.
```

FIGURE 3.1 – Exemple d'exécution du programme

3.1.3 Structure XML

Les fiches de termes produites par le programme python ont été évaluées en contexte d'une base de données créée sur le logiciel MultiTerm 2021, dont la description des champs, similaires à ceux de la « *ESA Termbase* », est représentée Figure 3.2. Dans un premier temps, des entrées ont été créées de manière manuelle afin d'étudier la structure du fichier résultant de l'exportation d'une fiche et de reprendre cette structure pour le fichier XML généré par le programme. Puis au cours de l'expérience, les fichiers créés ont été importés dans la base de données pour certifier leur efficacité.

Le fichier de sortie s'adapte à la structure de la base de données du moment qu'elle comprenne un champ de définition. De plus, la structure XML est facilement modifiable dans le programme python, par exemple pour changer le nom des champs ou ajouter d'autres champs.

Entry Structure

	Mandatory	Multiple
Entry level		
Domain		•
Subdomain		•
Image		•
Language level		
Term level		
Source		•
POS		•
Definition		•
Context		•
Description		•
Example		•
Date		•
Update		•

Descriptive Fields

Name	History	Type	Picklist Values
Context		Text	
Date		Text	
Definition		Text	
Description		Text	
Domain		Picklist	Finance | Environnement | Business | Medecine | Studies | Geography | History | Biology | Chemistry | Informatics | Linguistics | Mechanics
Example		Text	
Image		Multimedia File	
POS		Text	
Source		Text	
Subdomain		Picklist	Management | Linguistics | Market | Safety & Health | Oceans & Lacs
Update		Text	

FIGURE 3.2 – Définition des champs de la base terminologique de test

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE conceptGrp SYSTEM "Termbase_entry.dtd">
<conceptGrp>
    <languageGrp>
        <language lang="EN-GB" type="English" />
        <termGrp>
            <term>tourism</term>
            <descripGrp>
                <descrip type="Definition">The business of providing services
                such as transport, places to stay, or entertainment for
                people who are on holiday.</descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Source">&lt;xref Ulink="
                https://dictionary.cambridge.org/dictionary/english/tourism"
                &gt;TOURISM | English meaning - Cambridge Dictionary&lt;/xref
                &gt;, last consulted on: 15/11/2022</descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Date">15/11/2022</descrip>
            </descripGrp>
        </termGrp>
    </languageGrp>
    <languageGrp>
        <language lang="FR" type="French" />
        <termGrp>
            <term>tourisme</term>
            <descripGrp>
                <descrip type="Definition">Ensemble des activit&#233;s
                touristiques (s&#233;jours, voyages d'agr&#233;ment).</descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Source">&lt;xref Ulink="
                https://www.cnrtl.fr/definition/tourisme"&gt;TOURISME : D
                &#233;finition de TOURISME&lt;/xref&gt;, last consulted on:
                15/11/2022</descrip>
            </descripGrp>
            <descripGrp>
                <descrip type="Date">15/11/2022</descrip>
            </descripGrp>
        </termGrp>
    </languageGrp>
</conceptGrp>
```

FIGURE 3.3 – Fiche terminologique de « tourism » (tourisme)

3.2 Discussion

Les essais effectués pour le bon développement du script ont été concentrés sur une liste très réduite de termes pour assurer le respect des droits d'auteur du corpus. Le Dictionnaire de Cambridge, par exemple, prévient l'utilisateur de ne pas « extraire ou stocker le contenu du Site sur un serveur ou autre dispositif de stockage, ou de ne pas créer de base de données électronique en téléchargeant et conservant le contenu du Site. » Les droits d'utilisation des sites sont habituellement décrits et disponibles parmi les liens du menu ou en pied-de page sous le titre *Conditions générales d'utilisation* ou *Informations légales*.

Une autre limite du corpus concerne les noms propres et expressions figées. Une majorité de ces cas n'ont pas d'article qui leur soit destiné dans les dictionnaires qui composent le corpus, et cela apporte un frein à l'objectif principal de ce mémoire. Toutefois, une adaptation plus poussée du script pour contrer cet obstacle serait réalisable. De même, le script est écrit de telle façon qu'un utilisateur puisse facilement ajouter la prise en charge d'autres langues.

Diverses manières de repérer les définitions ont été considérés lors de la conception du projet, en dehors de la navigation par éléments et attributs HTML, dont la

recherche par patrons syntaxiques (séquences de parties du discours) avec un modèle d'apprentissage automatique neuronal (semblable à [Peñas et al., 2001]) et l'extraction entièrement par expressions régulières. La seconde méthode imposerait l'application de multiples conditions dans le script car les sites ne présentent pas les définitions de la même façon et cela produirait beaucoup de faux positifs. La première proposition donne de son côté une bonne perspective d'amélioration du script pour distinguer les définitions des autres composants textuels des dictionnaires. Il est néanmoins important de prendre en compte le coût élevé en temps d'apprentissage et d'exécution de la machine qu'apporterait l'écriture d'un modèle neuronale.

Deux autres sujets d'amélioration seraient (1) de rendre le formulaire du script plus esthétique et accessible en le transposant au format HTML, et (2) d'automatiser l'ajout du champ qui indique le domaine d'usage d'une entrée terminologique. Certaines définitions consultées au cours du projet sont précédées du domaine où elles sont spécifiques, une expression régulière permettrait assurément de récupérer cette information et de l'ajouter à la structure XML.

Sur un sujet annexe, [**?**] développe une méthode d'indexation de documents utilisant le calcul de spécificité des termes TF-IDF (*term frequency-inverse document frequency*). L'application de cette méthode sur les définitions une fois extraites permettrait de gagner encore du temps sur le choix de la définition à mettre dans la fiche. Cependant pour assurer la qualité des résultats, un grand corpus d'au moins 10 000 *tokens* serait imposé, ainsi qu'un important investissement temporel.

Conclusion générale

La traduction, qu'elle soit manuelle ou automatique, requiert un travail de recherche important du contexte culturel de la langue source et de ou des langues cibles, et des termes qui reflètent ces cultures. C'est pourquoi de nombreux travaux de recherche sur la traduction automatique et la dictionnairique ont été réalisés les vingt dernières années en traitement automatique des langues.

Inspiré des articles de ces recherches et des tâches de terminologie réalisées au cours d'un stage au sein de l'Agence spatiale européenne, ce mémoire propose une méthode d'automatisation de l'extraction de définitions et de la création d'entrées de base terminologique.

A l'issue de ce mémoire est produit un programme en langage Python qui prend en charge la construction de fiches terminologiques d'au moins trois langues (français, anglais, allemand) au format XML en passant par d'extraction de définitions à partir de sites internet à travers des balises HTML et des expressions régulières. Différentes manières de faire ont été étudiées avant de se décider pour cette méthode pour raison de familiarité, d'efficacité et de contrainte temporelle. Cependant il serait intéressant d'effectuer une comparaison de ces méthodes.

Les expériences sur une liste fixée de termes a permis d'observer les limites du programme, mais aussi son efficacité quant à l'objectif principal du projet. Le programme présente notamment une limite dans l'adaptation aux sites de dictionnaires. Il doit également respecter les droits d'auteurs et les conditions d'utilisation.

De nombreuses améliorations sont possibles : rendre le script plus agréable à utiliser (plateforme HTML), appliquer l'automatisation à la décision des définitions (TF-IDF), rendre le script plus fluide par rapport aux sites en créant un modèle d'apprentissage neuronal pour distinguer les définitions.

BIBLIOGRAPHIE

[Arrami, 2021] Arrami, S. (2021). Parser. https://definitions-digital.com/developpeur/parser. – Non cité.

[Atril,] Atril. Page d'accueil. https://atril.com/. – Non cité.

[Azimi, 2022a] Azimi, R. (2022a). Les artistes, ces créateurs de lieux en afrique. https://www.lemonde.fr/afrique/article/2022/11/13/les-artistes-ces-createurs-de-lieux-en-afrique_6149649_3212.html. – Non cité.

[Azimi, 2022b] Azimi, R. (2022b). Sara sadik, défricheuse de banlieue. https://www.lemonde.fr/m-le-mag/article/2022/11/14/sara-sadik-defricheuse-de-banlieue_6149721_4500055.html. – Non cité.

[Beautiful Soup,] Beautiful Soup. Beautiful soup documentation. https://beautiful-soup-4.readthedocs.io/en/latest/. – Non cité.

[Berlin-Brandenburgische Akademie der Wissenschaften,] Berlin-Brandenburgische Akademie der Wissenschaften. Digitales wörterbuch der deutschen sprache. https://www.dwds.de/. – Non cité.

[Bhoir and Jayamalini, 2021] Bhoir, H. and Jayamalini, K. (2021). Web Crawling on News Web Page using Different Frameworks. *International Journal of Scientific Research in Science and Technology*, pages 513–519. – Cité page 16.

[Cambridge Dictionary, 1999] Cambridge Dictionary (1999). Cambridge dictionary | english dictionary, translations thesaurus. https://dictionary.cambridge.org/. – Non cité.

[Cisse et al., 2008] Cisse, M., Diagne, A., Van Campenhoudt, M., and Muraille, P. (2008). Repenser le dictionnaire électronique multilingue dans un contexte d'aménagement linguistique. In *Actes du colloque de la Société française de terminologie*, pages 47–70, Paris. Société française de terminologie. – Cité page 16.

[CNRS, 2005] CNRS (2005). Accueil. https://cnrtl.fr/. – Non cité.

[Collectif, 2022a] Collectif (2022a). Liste des codes http. https://fr.wikipedia.org/wiki/Liste_des_codes_HTTP. – Non cité.

[Collectif, 2022b] Collectif (2022b). Liste des codes iso 639-1. https://fr.wikipedia.org/wiki/Liste_des_codes_ISO_639-1. – Cité page 27.

[Collins English Dictionary, 2007] Collins English Dictionary (2007). Collins online dictionary | definitions, thesaurus and translations. https://www.collinsdictionary.com/. – Non cité.

[Duden,] Duden. Duden | sprache sagt alles. https://www.duden.de/. – Non cité.

[European Space Agency,] European Space Agency. Home page. https://www.esa.int/. – Non cité.

[European Space Agency, 2010] European Space Agency (2010). *SP-1317, ESA CONVENTION AND COUNCIL RULES OF PROCEDURE*. ESA Communications, ESTEC, Netherlands, 7 edition. – Cité page 15.

[Exel et al., 2020] Exel, M., Buschbeck, B., Brandt, L., and Doneva, S. (2020). Terminology-Constrained Neural Machine Translation at SAP. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 271–280, Lisboa, Portugal. European Association for Machine Translation. – Cité pages 14 et 16.

[Glez-Peña et al., 2014] Glez-Peña, D., Lourenço, A., López-Fernández, H., Reboiro-Jato, M., and Fdez-Riverola, F. (2014). Web scraping technologies in an API world. *Briefings in Bioinformatics*, 15(5):788–797. – Cité page 16.

[Hamidi and Grifoni-Winters, 2021] Hamidi, M. and Grifoni-Winters, E. (2021). From index cards to neural machine translation: steps towards harmonising European space terminology. page 14, Dubai, United Arab Emirates. – Cité pages 14 et 21.

[Kageura and Marshman, 2019] Kageura, K. and Marshman, E. (2019). Terminology extraction and management. In *The Routledge Handbook of Translation and Technology*. Routledge. – Cité pages 14, 15 et 16.

[Laboratoire de recherche public à Nancy,] Laboratoire de recherche public à Nancy. Analyse et traitement informatique de la langue française. https://www.atilf.fr/. – Non cité.

[Larousse,] Larousse. Larousse.fr : encyclopédie et dictionnaires gratuits en ligne. https://www.larousse.fr. – Non cité.

[Mangeot, 1999] Mangeot, M. (1999). Accès unique à des dictionnaires hétérogènes. In Clas, A., editor, *Proc. LTT'99*, volume 1, page 3, Beyrouth, Lebanon. – Cité pages 15 et 29.

[Mangeot et al., 2003] Mangeot, M., Bilac, S., and Thevenin, D. (2003). Construction collaborative d'un dictionnaire multilingue : le projet Papillon. In *JSF'2003 Journées Scientifiques Francophones*, page 3, National Olympic Memorial Youth Center, Tokyo, Japan. – Cité page 15.

[memoQ,] memoQ. Page d'accueil. https://www.memoq.com/fr. – Non cité.

[Peñas et al., 2001] Peñas, A., Verdejo, F., and Gonzalo, J. (2001). Corpus-based terminology extraction applied to information access. In *Proceedings of the Corpus Linguistics 2001 conference*, volume 13, Special issue, Lancaster University (UK). UCREL Technical Paper. – Cité pages 16 et 38.

[Phister, 2011] Phister, B. (2011). L'informatique, une aide pour le traducteur. https://doi.org/10.4000/traduire.251. – Non cité.

[PTS,] PTS. Les outils de traduction assistée par ordinateur (TAO). https://www.translations-by-engineers.com/blog-traductions/bon-a-savoir/outils-de-traduction-assistee-par-ordinateur-tao/?utm_source=www.translations-by-engineers.com&utm_medium=social&utm_campaign=OpenGraph&utm_content=og%3Aurl. – Cité page 13.

[Python documentation, a] Python documentation. time — time access and conversions. https://docs.python.org/3.7/library/time.html. – Non cité.

[Python documentation, b] Python documentation. urllib.request - extensible library for opening urls. https://docs.python.org/3.7/library/urllib.request.html. – Non cité.

[robotstxt.org, 2000] robotstxt.org (2000). About /robots.txt. http://www.robotstxt.org/robotstxt.html. – Non cité.

[Roulois, 2020] Roulois, A. (2020). Web scraping. https://github.com/Alex-bzh/python-M1TAL/blob/main/5.web-corpus/0.web-scraping.ipynb. – Non cité.

[RWS, a] RWS. Language weaver. https://www.rws.com/language-weaver/our-history/. – Non cité.

[RWS, b] RWS. Trados studio. https://www.trados.com/fr/products/trados-studio/. – Non cité.

[Tomokiyo et al., 2006] Tomokiyo, M., Weyer-Brown, P., and Mangeot, M. (2006). Représentation Sémantique de Lexique pour un Dictionnaire de Traduction Manuelle et Automatique. In *Actes des Aspects méthodologiques pour l'élaboration de lexiques unilingues et multilingues*, page 12, Bertinoro, Forli, Italy. – Cité pages 9, 14, 15, 24 et 29.

[Tran, 2022] Tran, P. (2022). Encodage d'une url : ce qu'il faut savoir pour le seo. https://clustaar.com/blog/encodage-url-seo/. – Non cité.

[tutorialspoint,] tutorialspoint. Python – date time. https://www.tutorialspoint.com/python/python_date_time.htm. – Non cité.

[UTF8 chartable,] UTF8 chartable. Utf-8 encoding table and unicode characters. https://www.utf8-chartable.de/. – Non cité.

[Wang, 2013] Wang, P. S. (2013). *Dynamic Web Programming and HTML5*. Chapman and Hall/CRC, New York. – Cité page 28.

ANNEXE A

FICHIERS DU PROJET

FIGURE A.1 – Programme Python (search_definitinos.py)

```python
# Kirsten BERLAND
# Master 2 NLP specialised in Translation
# Execute with: python search_definitions.py

# Import libraries
import re, time # Regular expressions and time
import urllib.request # To access web sites
from bs4 import BeautifulSoup # To fetch web sites' content

# Fonctions
"""
addDicoLink() takes a term and a list of dictionary links from a file as parametres
to create a link to dictionary entries which is then added to dicoLinks.
"""
def addDicoLink(dico, query):
    print("Charging dictionary links...")
    index = 0 # interger which will serve as key
    for line in dico:
        dicoLinks[index] = {}
        # Each line has a link.
        # Adding the term searched to the link makes it a dictionary entry
        newlink = line.strip() + query
        # The links are saved in a dictionary for later
        dicoLinks[index]["Link"] = newlink
        index += 1

"""
getTermInfo() searches definitions of a term from dictinary sites.
It returns the number of sites where one or more definitions are found
to determine, later in the script, if the user can choose a dictionary
and definition for the term.
"""
def getTermInfo():
    print("Getting term definitions...\n")
    hasDefinition = len(dicoLinks)
    # print(hasDefinition)
    for num, info in dicoLinks.items():
        link = dicoLinks[num]["Link"]
        # print(f'--{num}. {link}\n')
        try:
            # Get access to the sitepage
            request = urllib.request.Request(link, headers = headers)
            # Load HTML document
            with urllib.request.urlopen(request) as page:
                # Get the HTML content
                html = page.read()
                soup = BeautifulSoup(html, 'lxml')
                # Find the name of the site (<tag property="og:site_name"></tag>)
                site = soup.find(property="og:site_name")
                # print(site)
                pagetitle = soup.title.string
                if pagetitle:
                    pagetitle = pagetitle.strip()
                    dicoLinks[num]["Page title"] = pagetitle
                else:
                    dicoLinks[num]["Page title"] = term
                if site:
                    # print("Site: " + site.get("content"))
                    dicoLinks[num]["Site name"] = site.get("content")
```

```
                # Find tags called <span> or <li> (the most frequent for definitions
                # in the links used) and where the attribute class corresponds
                # to the following regular expression
                HTMLtag = re.compile("(span|li)")
                regex = re.compile(".*(D|d)efinition")
                definitions = soup.find_all(HTMLtag, class_ = regex)
                definitions = definitions[:10] # Keep only the 10 first tags found
                listDefinitions = []
                # print(definitions)
                if definitions:
                    for candidate in definitions:
                        candidate = candidate.get_text()
                        candidate = candidate.strip()
                        # Removal of everthing but the first line in the tag
                        # because they are mostly examples or synonyms
                        candidate = re.sub("\r?\n.+", "", candidate)
                        candidate = re.sub(r"\W:[\W\w]*", "", candidate)
                        # Some tag content may begin with a number, which we remove.
                        if re.match(r"\d+\.\s", candidate):
                            candidate = re.sub(r"^\d+\.\s", "", candidate)
                        candidate = candidate.capitalize()
                        # The following caracters are found in french dictionaries
                        # they are replaced by the adequate quotation marks
                        if re.match(r",,", candidate):
                            candidate = re.sub(r",,", "", candidate)
                            candidate = candidate.capitalize()
                            candidate = "« " + candidate
                            candidate = re.sub("``", " »", candidate)
                        if candidate[-1] not in ['.', "»", "\""]:
                            candidate = candidate + '.'

                        if candidate not in listDefinitions:
                            listDefinitions.append(candidate)
                    # print(listDefinitions)
                else:
                    regex = re.compile("Bedeutung-.+")
                    definitions = soup.find_all(HTMLtag, id = regex)
                    # print(definitions)
                    definitions = definitions[:10]
                    listDefinitions = []
                    for candidate in definitions:
                        candidate = candidate.get_text()
                        candidate = candidate.strip()
                        # Removal of everthing but the first line in the tag
                        # because they are mostly examples or synonyms
                        candidate = re.sub("\r?\n.+", ".", candidate)
                        candidate = re.sub("\.+$", ".", candidate)
                        candidate = candidate.capitalize()

                        if candidate not in listDefinitions:
                            listDefinitions.append(candidate)
                    # print(listDefinitions)

                    if listDefinitions == []:
                        regex = re.compile("def .+")
                        definitions = soup.find_all("div", class_ = regex)
                        # print(definitions)
                        definitions = definitions[:10]
                        listDefinitions = []
                        for candidate in definitions:
                            candidate = candidate.get_text()
                            candidate = candidate.strip()
                            # Removal of everthing but the first line in the tag
                            # because they are mostly examples
                            candidate = re.sub("\r?\n.+", "", candidate)
                            candidate = candidate.replace(":", ".")
                            candidate = candidate.capitalize()

                            if candidate not in listDefinitions:
                                listDefinitions.append(candidate)
                        # print(listDefinitions)

                if listDefinitions == []:
                    print("No definition found in " + link)
                    hasDefinition -= 1 # Decrement the number of sites where definitions where found

                dicoLinks[num]["Term definitions"] = listDefinitions
            # print(hasDefinition)
            # if site:
                # print(f'{num}: {dicoLinks[num]["Site name"]}, {dicoLinks[num]["Page title"]}, {dicoLinks[num]["Link"]}\n')
            # else:
                # print(f'{num}: {dicoLinks[num]["Page title"]}, {dicoLinks[num]["Link"]}\n')

        # In case the web site is not accessible, write an error message
        except urllib.error.HTTPError as e:
            hasDefinition -= 1
            print(f"HTTPError: {e.code} for {link}\n")
    # print(dicoLinks)
    return hasDefinition

```

```
"""
chooseDictionary() allows the user to pick a dictionary for the definition of a term.
It returns a string and an integer: 'source' is the reference of the sitepage
that will be added to the MultiTerm supported file,
and 'reference' is the key of the source selected in the dicoLinks dictionary.
"""
def chooseDictionary():
    # Enumerate the sites and the definitions found
    for num, site in dicoLinks.items():
        if "Page title" in site:
            print(f'\n{num}. {site["Page title"]}, {site["Link"]}')
            defNum = 0
            for definition in site["Term definitions"]:
                print(f'\t{defNum}. {definition}')
                defNum += 1

    reference = input("\nNumber of the reference to keep: ")
    reference = int(reference)
    if "Site name" in dicoLinks[reference]:
        source = dicoLinks[reference]["Site name"] + ', <xref Ulink="' +
        dicoLinks[reference]["Link"] + '">' + dicoLinks[reference]["Page title"] +
        '</xref>, last consulted on: ' + date
    else:
        source = '<xref Ulink="' + dicoLinks[reference]["Link"] + '">' +
        dicoLinks[reference]["Page title"] + '</xref>, last consulted on: ' + date
    # print(f"_____\nEntry:\n{lang}\n{term}\n{source}\n_____")
    return (reference, source)

"""
chooseDefinition() asks the user for the definition of a term,
and return the definition as a string thanks to 'reference', an integer,
that indicates from which dictionary is the definition.
"""
def chooseDefinition(reference):
    # List of the dictionary's definition proposals
    listOfDef = dicoLinks[reference]["Term definitions"]
    definitionNum = input("Number of the definition to keep: ")
    definition = listOfDef[int(definitionNum)]
    return definition

# Variable definition
headers = {'User-agent' : 'kberland.etudes@gmail.com'} # Connection identifitator

termsEN = []
termsFR = []
termsDE = [] # Those lists will be used to make the MultiTerm entry file

languages = ["EN", "FR", "DE"]

hasDefinition = 0 # Determine the number of sites with definitions

begintime = time.time()
date = time.strftime("%d/%m/%Y", time.localtime())
print("Date: " + date)

request_languages = input("Language or languages separated by ';' [EN/FR/DE]: ")
request_languages = request_languages.upper()
request_languages = request_languages.split(";")
# print(request_languages)

# Loop for each language requested
for lang in request_languages:
    lang = lang.strip()
    # print(lang)

    # Loop for when the language code is not understood
    while lang not in languages:
        lang = input(f"!!! {lang} not supported.\nLanguage [EN/FR/DE]: ")
        lang = lang.upper().strip()
        # Next step proposition in the project:
        # would you like to add a new language to the system?

    terms = input(f"{lang} term or terms separated by ';': ")
    terms = terms.split(";")
    # print(terms)
    print('\n')
    if lang == "FR":
        for term in terms:
            termsFR.append(dict()) # Create a dictionary for the term

            term = term.strip() # Remove superfluous space
            termsFR[-1]["term"] = term

            # Term adapted to URL standard by replacing special characters with their code
            # taken from https://www.utf8-chartable.de/
            query = re.sub(' ', '%20', term)
            toreplace = ['è','é','ê','ë','ô','ö','à','â','î','ï','û',
                         'ù','œ','Œ','È','É','Ê','Ë','Ô','Ö','À','Â','Î','Ï']
            replaceby = ['%c3%a8','%c3%a9','%c3%aa','%c3%ab','%c3%b4',
                         '%c3%b6','%c3%a0','%c3%a2', '%c3%ae','%c3%af',
```

```
                    '%c3%b9','%c3%bb','oe','oe', '%c3%88', '%c3%89',
                    '%c3%8a', '%c3%8b', '%c3%94', '%c3%96', '%c3%80',
                    '%c3%82', '%c3%8e', '%c3%8f']
            for i in range(len(toreplace)):
                query = re.sub(toreplace[i], replaceby[i], query)
            # print(query)

            # Finding links for the term
            dicoLinks = {}
            file = open("dictionaries/dicoFr.txt", 'r')
            dico = file.readlines()
            addDicoLink(dico, query)
            # print(dicoLinks)
            hasDefinition = getTermInfo()
            print("\nDictionaries searched.")
            # print(hasDefinition)
            # If a definition is found...
            if dicoLinks != {} and hasDefinition != 0:
                # Choose a dictionnairy to keep
                reference, definitionsrc = chooseDictionary()
                termsFR[-1]["definitionsrc"] = definitionsrc
                # Choose a definition to keep
                definition = chooseDefinition(reference)
                termsFR[-1]["definition"] = definition
            else:
                print("No definition found.")
        # print(termsFR)

    elif lang == "DE":
        for term in terms:
            termsDE.append(dict()) # Create a dictionary for the term

            term = term.strip()
            termsDE[-1]["term"] = term

            # Term adapted to URL standard
            query = term.replace(' ', '%20')
            toreplace = ['ä', 'ä', 'ö', 'ö', 'ü', 'Ü']
            replaceby = ['ae', 'Ae', 'oe', 'Oe', 'ue', 'Ue']

            for i in range(len(toreplace)):
                query = re.sub(toreplace[i], replaceby[i], query)
            # print(query)

            # Finding links for the term
            dicoLinks = {}
            file = open("dictionaries/dicoDe.txt", 'r')
            dico = file.readlines()
            addDicoLink(dico, query)
            # print(dicoLinks)
            hasDefinition = getTermInfo()
            print("\nDictionaries searched.")
            # print(hasDefinition)
            # If a definition is found...
            if dicoLinks != {} and hasDefinition != 0:
                # Choose a dictionnairy to keep
                reference, definitionsrc = chooseDictionary()
                termsDE[-1]["definitionsrc"] = definitionsrc
                # Choose a definition to keep
                definition = chooseDefinition(reference)
                termsDE[-1]["definition"] = definition
            else:
                print("No definition found.")
        # print(termsDE)

    elif lang == "EN":
        for term in terms:
            termsEN.append(dict()) # Create a dictionary for the term

            term = term.strip()
            termsEN[-1]["term"] = term

            # Term adapted to URL standard
            query = term.replace(' ', '%20')

            # Creating links for the term
            dicoLinks = {}
            file = open("dictionaries/dicoEn.txt", 'r')
            dico = file.readlines()
            addDicoLink(dico, query)
            # print(dicoLinks)
            hasDefinition = getTermInfo()
            print("\nDictionaries searched.")
            # print(hasDefinition)
            # If a definition is found...
            if dicoLinks != {} and hasDefinition != 0:
                # Choose a dictionnairy to keep
                reference, definitionsrc = chooseDictionary()
                termsEN[-1]["definitionsrc"] = definitionsrc
```

```
                # Choose a definition to keep
                definition = chooseDefinition(reference)
                termsEN[-1]["definition"] = definition
            else:
                print("No definition found.")
        # print(termsEN)

# Transposition to MultiTerm supported XML
import xml.etree.ElementTree as ET
filename = "my_termbase_entry.xml"

ID = input("Entry number [type - if none]: ")

with open(filename, 'w') as xmlfile:
    xmlfile.write("<?xml version=\"1.0\" encoding=\"UTF-8\"?>")
    xmlfile.write("<!DOCTYPE conceptGrp SYSTEM \"Termbase_entry.dtd\">")

with open(filename, 'ab') as xmlfile:
    print("Writting XML file...")
    # Root element to tree structure
    root = ET.Element('conceptGrp')
    tree = ET.ElementTree(root)
    if ID.isdigit(): # Checks if all the characters in the variable are digits
        conceptID = ET.SubElement(root, 'concept')
        conceptID.text = ID

    if 'EN' in request_languages:
        languageGrp_EN = ET.SubElement(root, 'languageGrp')
        lang_EN = ET.SubElement(languageGrp_EN, 'language', {'type' : 'English', 'lang' : 'EN-GB'})
        # <languageGrp><language type="English" lang="EN-GB"></language></languageGrp>
        for term in termsEN:
            termGrp_EN = ET.SubElement(languageGrp_EN, 'termGrp')
            term_EN = ET.SubElement(termGrp_EN, 'term')
            term_EN.text = term["term"]
            # <termGrp><term>...</term></termGrp>

            if "source" in term.keys():
                descripGrp_EN_Source = ET.SubElement(termGrp_EN, 'descripGrp')
                descrip_EN_Source = ET.SubElement(descripGrp_EN_Source, 'descrip', {'type': 'Source'})
                descrip_EN_Source.text = term["source"]
                # <descripGrp><descrip type="Source">...</descrip></descripGrp>

            if "definition" in term.keys():
                descripGrp_EN_Definition = ET.SubElement(termGrp_EN, 'descripGrp')
                descrip_EN_Definition = ET.SubElement(descripGrp_EN_Definition, 'descrip', {'type': 'Definition'})
                descrip_EN_Definition.text = term["definition"]
                # <descripGrp><descrip type="Definition">...</descrip></descripGrp>

            if "definitionsrc" in term.keys():
                descripGrp_EN_Source = ET.SubElement(termGrp_EN, 'descripGrp')
                descrip_EN_Source = ET.SubElement(descripGrp_EN_Source, 'descrip', {'type': 'Source'})
                descrip_EN_Source.text = term["definitionsrc"]
                # <descripGrp><descrip type="Source">...</descrip></descripGrp>

            descripGrp_EN_Date = ET.SubElement(termGrp_EN, 'descripGrp')
            descrip_EN_Date = ET.SubElement(descripGrp_EN_Date, 'descrip', {'type': 'Date'})
            descrip_EN_Date.text = date
            # <descripGrp><descrip type="Date">...</descrip></descripGrp>

    if 'FR' in request_languages:
        languageGrp_FR = ET.SubElement(root, 'languageGrp')
        lang_FR = ET.SubElement(languageGrp_FR, 'language', {'type' : 'French', 'lang' : 'FR'})
        # <languageGrp><language type="French" lang="FR"></language></languageGrp>
        for term in termsFR:
            termGrp_FR = ET.SubElement(languageGrp_FR, 'termGrp')
            term_FR = ET.SubElement(termGrp_FR, 'term')
            term_FR.text = term["term"]
            # <termGrp><term>...</term></termGrp>

            if "source" in term.keys():
                descripGrp_FR_Source = ET.SubElement(termGrp_FR, 'descripGrp')
                descrip_FR_Source = ET.SubElement(descripGrp_FR_Source, 'descrip', {'type': 'Source'})
                descrip_FR_Source.text = term["source"]
                # <descripGrp><descrip type="Source">...</descrip></descripGrp>

            if "definition" in term.keys():
                descripGrp_FR_Definition = ET.SubElement(termGrp_FR, 'descripGrp')
                descrip_FR_Definition = ET.SubElement(descripGrp_FR_Definition, 'descrip', {'type': 'Definition'})
                descrip_FR_Definition.text = term["definition"]
                # <descripGrp><descrip type="Definition">...</descrip></descripGrp>

            if "definitionsrc" in term.keys():
                descripGrp_FR_Source = ET.SubElement(termGrp_FR, 'descripGrp')
                descrip_FR_Source = ET.SubElement(descripGrp_FR_Source, 'descrip', {'type': 'Source'})
                descrip_FR_Source.text = term["definitionsrc"]
                # <descripGrp><descrip type="Source">...</descrip></descripGrp>

            descripGrp_FR_Date = ET.SubElement(termGrp_FR, 'descripGrp')
            descrip_FR_Date = ET.SubElement(descripGrp_FR_Date, 'descrip', {'type': 'Date'})
            descrip_FR_Date.text = date
            # <descripGrp><descrip type="Date">...</descrip></descripGrp>
```

```
    if 'DE' in request_languages:
        languageGrp_DE = ET.SubElement(root, 'languageGrp')
        lang_DE = ET.SubElement(languageGrp_DE, 'language', {'type' : 'German', 'lang' : 'DE'})
        # <languageGrp><language type="German" lang="DE"></language></languageGrp>
        for term in termsDE:
            termGrp_DE = ET.SubElement(languageGrp_DE, 'termGrp')
            term_DE = ET.SubElement(termGrp_DE, 'term')
            term_DE.text = term["term"]
            # <termGrp><term>...</term></termGrp>

            if "source" in term.keys():
                descripGrp_DE_Source = ET.SubElement(termGrp_DE, 'descripGrp')
                descrip_DE_Source = ET.SubElement(descripGrp_DE_Source, 'descrip', {'type': 'Source'})
                descrip_DE_Source.text = term["source"]
                # <descripGrp><descrip type="Source">...</descrip></descripGrp>

            if "definition" in term.keys():
                descripGrp_DE_Definition = ET.SubElement(termGrp_DE, 'descripGrp')
                descrip_DE_Definition = ET.SubElement(descripGrp_DE_Definition, 'descrip', {'type': 'Definition'})
                descrip_DE_Definition.text = term["definition"]
                # <descripGrp><descrip type="Definition">...</descrip></descripGrp>

            if "definitionsrc" in term.keys():
                descripGrp_DE_Source = ET.SubElement(termGrp_DE, 'descripGrp')
                descrip_DE_Source = ET.SubElement(descripGrp_DE_Source, 'descrip', {'type': 'Source'})
                descrip_DE_Source.text = term["definitionsrc"]
                # <descripGrp><descrip type="Source">...</descrip></descripGrp>

            descripGrp_DE_Date = ET.SubElement(termGrp_DE, 'descripGrp')
            descrip_DE_Date = ET.SubElement(descripGrp_DE_Date, 'descrip', {'type': 'Date'})
            descrip_DE_Date.text = date
            # <descripGrp><descrip type="Date">...</descrip></descripGrp>

    # Create XML file out of the tree structure
    tree.write(xmlfile)

print("XML file ready.")

endtime = time.time()
duration = endtime - begintime
print("Time spent: " + str(duration))
```

FIGURE A.2 – Document type définition de fichiers XML pour une entrée de base terminologique

```
<!ELEMENT conceptGrp (concept*,languageGrp+)>

<!ELEMENT concept (#PCDATA)>

<!ELEMENT languageGrp (language, termGrp)>

<!ELEMENT  language EMPTY>
<!ATTLIST  language
    lang CDATA #REQUIRED
    type CDATA #REQUIRED>

<!ELEMENT  termGrp (term, descripGrp+)>

<!ELEMENT  term (#PCDATA)>

<!ELEMENT  descripGrp (descrip)>

<!ELEMENT  descrip (#PCDATA)>
<!ATTLIST  descrip
    type CDATA #REQUIRED>
```

Abréviations

API	Application Programming Interface
ASE	Agence spatiale européenne
CNRTL	Centre National de Ressources Textuelles et Lexicales
DOM	Document Object Model
DTD	Document Type Definition
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
TAO	Traduction Assistée par Ordinateur
TBX	TermBase eXchange
URL	Uniform Resource Locators
XML	eXtensible Markup Language

INDEX