



HAL
open science

Traduction Assistée par Ordinateur des Langues des Signes: élaboration d'un premier prototype

Marion Kaczmarek, Alix Larroque

► To cite this version:

Marion Kaczmarek, Alix Larroque. Traduction Assistée par Ordinateur des Langues des Signes: élaboration d'un premier prototype. Traitement Automatique des Langues Naturelles, 2021, Lille, France. pp.108-122. hal-03265902

HAL Id: hal-03265902

<https://hal.science/hal-03265902v1>

Submitted on 23 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Traduction Assistée par Ordinateur des Langues des Signes: élaboration d'un premier prototype

Marion Kaczmarek¹, Alix Larroque²

(1) LISN, Rue du Belvédère, 91400 Orsay, France

(2) Télécom Paris, 19 place Marguerite Perey, 91120 Palaiseau, France

marion.kaczmarek@lisn.upsaclay.fr, alix.larroque@telecom-paris.fr

RÉSUMÉ

La demande pour du contenu traduit en LSF est croissante depuis quelques années, mais l'offre est limitée par le faible nombre de traducteurs professionnels et l'absence d'outils de traduction assistée par ordinateur (TAO) dédiés pour les langues des signes (LS). Cet article s'intéresse à l'élaboration de tels outils. Après avoir étudié les méthodes de travail des traducteurs, nous avons établi un cahier des charges afin de développer un premier logiciel de TAO pour les LS. Nous avons procédé à la conception d'un tel système en développant des prototypes dits de basse fidélité avant d'implémenter une première version de logiciel fonctionnel. Nous établissons les fonctionnalités implémentées à la date de rédaction de cet article, et évoquons les fonctionnalités restant à être implémentées. Après un test du logiciel par les traducteurs professionnels, nous pourrions ensuite procéder à l'évaluation du système, afin d'améliorer son implémentation d'après leurs retours.

ABSTRACT

Computer-assisted Translation of Sign Languages: elaborating a first prototype.

The demand for Sign Language translated content has been growing in recent years, but the answer is quite limited by the scarcity of professional translators and the lack of dedicated computer-assisted translation (CAT) tools for sign languages. This article focuses on the development of such tools. After studying the working methods of translators, we established a set of specifications in order to develop a first CAT software for sign languages. To design such a system, we developed low-fidelity prototypes before implementing a first version of functional software. We list and explain the functionalities implemented at the time of writing, and discuss other functionalities still to be implemented. After testing by professionals, we aim at improving the application according to their feedback.

MOTS-CLÉS : Traduction assistée par ordinateur – Langue des signes - IHM

KEYWORDS: Computer-assisted translation – sign languages - HCI

1 Introduction

En France, la loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées parue en 2005 reconnaît la Langue des Signes Française (LSF), comme une langue à part entière, et comme une langue de la République au même titre que le français. De cette loi devrait découler une accessibilité totale en LSF pour les personnes sourdes (accueil dans les lieux publics, messages audio diffusés, information écrite). Renforcé par l'adoption en 2008 de la

Convention sur les Droits des Personnes Handicapées par les Nations Unies, le besoin pour du contenu accessible et donc traduit en LSF est croissant. Cependant, il n'y a en France encore que peu de traducteurs professionnels, et ces derniers ne sont en rien équipés comme le sont leurs collègues des langues vocales : aucun outil actuel de traduction assistée par ordinateur n'existe pour assister la traduction des langues signées.

1.1 Traduction assistée par ordinateur

Abrégée TAO, la traduction assistée par ordinateur désigne le recours à des aides logicielles par les professionnels de la traduction. Elle n'est pas à confondre avec la traduction automatique puisqu'ici, c'est bien l'opérateur humain expert qui reste aux commandes de la production. Elle se présente sous la forme d'utilitaires informatisés que le traducteur peut consulter (bases terminologiques, lexiques, glossaires etc), ou sous la forme d'environnements de travail dit intégrés, qui regroupent en une seule application de nombreuses fonctionnalités destinées à assister la tâche de traduction. Le traitement automatique des langues (TAL) y trouve une place prépondérante puisque l'on cherche à assister, si ce n'est automatiser les tâches de la traduction. Qu'il soit d'ordre syntaxique (délimitation de phrases), sémantique (traduction automatique, correcteurs orthographiques), ou par l'extraction d'informations, l'apport du TAL dans la TAO est conséquent. Démocratisée dans les années 80, elle fait aujourd'hui partie intégrante des métiers de la traduction au point que sa maîtrise tend à devenir un pré-requis pour rejoindre le monde du travail. A cheval entre informatique et traductologie, la TAO a contribué à faire évoluer le métier de traducteur. Cependant, ce constat ne s'applique pas à toutes les langues. C'est le cas des langues signées (LS), qui ne sont guère encore outillées, et dont nous détaillons les grands principes ci-après.

1.2 Les langues signées, ou langues des signes.

Les langues des signes (LS) sont des langues naturelles et orales qui utilisent les modalités visuo-gestuelle pour transmettre du sens, au travers d'articulateurs manuels, de mouvements du corps, du regard et des expressions du visage : les LS expriment plusieurs informations simultanément. Elles font usage de références spatiales persistantes dans leur organisation, qui peuvent être réutilisées dans la suite du discours. Discours qui est d'ailleurs organisé selon un ordre propre : on commence par une description préliminaire de la scène. On situe d'abord l'événement dans le temps, puis on introduit les lieux, les personnages. L'action en elle-même n'est traitée que dans un second temps. La pensée visuelle prévaut. Ainsi, l'ordre des informations présentes dans un discours diffère entre le français écrit et la LSF.

Les LS sont des langues iconiques, c'est-à-dire que les signes sont le plus souvent inspirés de la réalité (ce que la linguistique appelle l'iconicité, Cuxac, 1993). Par extension, elles sont parfois contraintes par la réalité, notamment lorsqu'il est question de géographie, ou de topologie. En effet, le contenu en LS se doit d'être visuel, et précis lorsqu'il se réfère à des relations existantes dans la réalité physique. Par exemple, lorsque l'on signe à propos de deux villes, et de la relation géographique qui les unit, le signeur doit convenablement situer les deux villes dans l'espace l'une par rapport à l'autre, en termes de distance et d'orientation.

Bien qu'il existe plusieurs systèmes pour décrire les LS sous forme graphique (SignWriting, HamNoSys par exemple, liens dans la sitographie), ils ne sont pas suffisamment utilisés par la communauté sourde au quotidien pour pouvoir être considérés comme des formes écrites. La façon la plus courante pour garder trace de la LS est l'enregistrement vidéo.

1.3 LS et informatique

La langue des signes n'est pas étrangère aux domaines informatiques. Plusieurs études, bien que récentes, s'y sont déjà intéressées, que cela soit dans le cadre de la génération automatique des LS (développement d'avatar signant) ou de la vision par ordinateur des langues des signes (reconnaissance de signes). En termes d'avatar, on citera notamment les travaux concernant la signalisation à l'adresse des personnes sourdes et malentendantes dans les gares françaises (Paire-Ficout et al, 2013). A l'internationale, l'Université DePaul de Chicago travaille sur la question depuis plus de quinze ans, et propose aujourd'hui un des avatars les plus aboutis qui soient concernant le rendu naturel des signes (Wolfe et al, 2016). D'autres études s'intéressent à la traduction des LS, mais d'un point de vue automatique, que cela soit par l'usage de gants électroniques ou d'applications smartphone.

Avant de se lancer dans le développement d'une interface de traduction dédiée aux langues des signes, il convient de s'intéresser aux logiciels déjà existants. Cette première observation nous permettra d'identifier les points clé de la TAO, mais surtout de mieux comprendre pourquoi, en l'état, ne peuvent-ils pas supporter les langues des signes comme langue de travail. La seconde étape de cette étude préliminaire s'intéresse quant à elle au processus de traduction, du français écrit vers la LSF: quelles en sont les étapes ? Sont-elles systématiques, et produites dans un ordre total ou partiel ? Ainsi, nous pourrons ensuite réfléchir à la façon d'adapter les grands principes de la TAO tant aux particularités de la LSF qu'aux pratiques traductives.

2 État de l'art

2.1 Traduction assistée par ordinateur

Les environnements de travail intégrés (c'est à dire qui regroupe tous les outils et fonctionnalités en une seule interface) destinés à la TAO fonctionnent comme suit: le traducteur charge d'abord le texte source, qu'il doit traduire. Dans un second temps, le logiciel découpe automatiquement ce texte en unités plus petites, généralement du grain de la phrase ou de la proposition, appelées segments. A chacun de ces segments sources est associé en vis-à-vis un segment cible, initialement vide. Le traducteur procède alors à la traduction, segment par segment, en ayant éventuellement recours aux outils proposés par l'environnement.

Les logiciels actuels reposent sur trois grands principes. Le premier, une forme écrite éditable: l'intégralité du logiciel repose sur l'usage d'une forme écrite éditable. En effet, aussi triviale que puisse paraître l'observation, tout passe par l'écrit: le texte source bien entendu, mais également la traduction elle-même qui sera rendue sous forme écrite, les menus, mais également les outils d'assistance tels que les dictionnaires, les glossaires, ainsi que la mémoire de traduction.

En second, la mémoire de traduction: au fur et à mesure que le traducteur traduit les segments sources, la mémoire de traduction enregistre les paires de segments sources-cibles, c'est à dire que chaque segment d'origine est appairé avec sa traduction correspondante. Si l'un des segments sources est à nouveau rencontré, à l'identique ou sous une forme très proche, alors le logiciel suggère automatiquement la traduction précédemment mémorisée: le professionnel est alors libre d'accepter, de rejeter, ou d'accepter la suggestion avec modification. La mémoire de traduction permet de capitaliser sur le travail passé. Ces bases de données sont partageables, entre collègues

d'un même service ou même parfois fournies par les clients eux-mêmes, et permettent plus de cohérence dans le temps ainsi qu'au sein des productions d'un même service.

Enfin, c'est ce que nous avons appelé "le principe de linéarité": l'ordre du texte source n'est pas retravaillé. Les segments sont traduits dans l'ordre d'origine, en considérant que la concaténation des segments traduits correspond à la traduction des segments sources concaténés.

Les logiciels de TAO actuels ne peuvent pas supporter la LSF parce que ces trois points clés semblent tous poser un problème d'adaptation. En effet, les langues des signes ne disposent pas de formes écrites éditables. L'usage de la vidéo pose questions en termes d'adaptation d'une mémoire de traduction puisqu'elles ne sont pas requêttables, difficilement éditables, et plus complexes à stocker en quantité. De même qu'on ne peut coller bout à bout plusieurs extraits de vidéos différentes pour produire une traduction qualitative. Enfin, comme mentionné plus haut, le discours en LSF est organisé selon un ordre propre qui diffère de celui du français, impliquant que le principe de linéarité n'est peut-être pas valable ici. La section suivante s'intéresse aux pratiques professionnelles des traducteurs, de sorte à identifier leurs besoins ainsi que les obstacles qu'ils sont amenés à rencontrer au quotidien.

2.2 Processus de traduction français écrit-LSF

Peu importe les langues, le but premier de la traduction reste de transmettre un message. Dans le cas des langues des signes, cette transition implique également un changement de modalité: on passe au visuo-gestuel. De fait, traduire du français vers la langue des signes française suppose un passage par la pensée visuelle, c'est-à-dire mettre le sens en image. Cette étape supplémentaire, dite de déverbalisation, permet d'affranchir la traduction de l'influence des constructions de l'écrit sans tomber dans le transcodage. L'exercice de reformulation, couplé à l'usage de schémas, permet au traducteur d'extraire dans un premier temps le sens du message qu'il doit faire passer, et de le reconstruire dans un second temps dans une forme propice en langue cible. (Pierre Guitteny, 2007. D. Seleskovitch et M. Lederer, 2014). La traduction texte à signes se différencie d'emblée de la traduction texte à texte par le fait qu'elle ne se construit pas au fur et à mesure du processus.

En effet, elle commence par une phase de traitement du texte source, pour établir un "scénario" de la traduction, avant que le traducteur ne procède à se filmer d'une traite. Cette méthode implique une phase d'entraînement, et de mémorisation de la version finale par le traducteur, comme nous le verrons dans le paragraphe suivant.

Afin de déterminer plus spécifiquement les étapes de la traduction français/langue des signes française, Kaczmarek et Filhol, (2020) ont mené en parallèle deux études avec des professionnels du métier. Un brainstorming d'une part, pour les amener à réfléchir et verbaliser tant leurs besoins que les problèmes rencontrés au quotidien dans leurs pratiques professionnelles. Et des sessions de traductions filmées d'autre part, où deux binômes de traducteurs étaient filmés pendant qu'ils travaillaient sur la traduction de textes journalistiques. Ils ont ainsi pu dresser une liste des tâches inhérentes à la traduction en LS, mais également déterminer si elles étaient systématiques, et ordonnées.

La figure 1 est une synthèse des tâches identifiées en observant des traducteurs à l'œuvre. La première ligne liste les tâches identifiées, et la première colonne liste les six traductions analysées, à savoir les trois mêmes textes pour chacun des deux groupes (groupe B pour "beginner" et groupe E pour "expert"). Une case verte signifie que telle tâche a été observée durant telle traduction. On

remarque que la seule tâche systématique (et la plus chronophage) est celle de segmentation et d'ordre.

	Total time	Lexical search	Discuss signs	Map search	Def. search	Encyclo search	Picture search	Seg & order	Memory
TR 1-1 (B)	1h13								
TR 2-1 (B)	59 min								
TR 3-1 (B)	12 min								
TR 1-2 (E)	19 min								
TR 2-2 (E)	14 min								
TR 3-2 (E)	11 min								
Total time	3h08	49 min	37 min	13 min	2 min	18 min	2 min	66 min	2 min

Figure 1: Tableau illustrant les tâches observées durant des traductions fr-LSF, Kaczmarek & Filhol, 2019

Les résultats issus du brainstorming corroborent ces observations, les professionnels faisant mention de la nécessité d'accéder à du contenu en langue des signes plus facilement, qu'il s'agisse de lexique ou de traductions déjà élaborées. Ces deux études permettent également de prendre connaissance des méthodes de travail et du matériel utilisé: sollicitation de ressources sur internet, papier-crayon, prise de notes directement sur les textes... La segmentation et la modification de l'ordre des informations par exemple, passent par du repérage dans le texte, et des listings de sections numérotées. La section suivante fait le point sur les besoins identifiés et les fonctionnalités logicielles qui pourraient leur être associées.

2.3 Cahier des charges

En nous basant sur ces observations préliminaires, nous avons pu dresser une liste de fonctionnalités qu'un logiciel de TAO à destination des traducteurs en LS pourrait intégrer. La tâche de segmentation et d'ordre étant, d'après la figure 1, systématique mais également celle qui prend le plus de temps. En regard du principe de linéarité, qui ne semble pas admis de fait en LSF, et il apparaît complexe d'envisager une fonction de segmentation automatique pour le texte source. En revanche, le travail sur l'ordre (à savoir découper le texte et replacer les informations dans l'ordre propre de la LSF) peut être assisté plus facilement, tout comme le traitement du texte source ou l'accès aux ressources. Les paragraphes ci-après détaillent le cahier des charges que nous avons dressé pour le développement d'une première interface.

Gestion du texte source: Le logiciel doit être en mesure de charger et d'afficher le texte sur lequel travaille le traducteur. Il doit pouvoir prendre en charge les formats les plus communs, ainsi qu'en conserver la mise en page. En réponse aux suggestions émises lors du brainstorming quant à la possibilité de pré-traiter le texte, l'application propose les rudiments du traitement automatique, à savoir la reconnaissance de dates ainsi que la reconnaissance d'entités nommées.

Écriture et linéarité: Pour contourner deux des problèmes majeurs liés à l'adaptation des principes de la TAO pour les LS, à savoir l'absence de forme écrite ainsi que la non-application du principe de linéarité, nous suggérons un système de blocs mobiles et hiérarchisés. Le traducteur est libre de

créer autant de blocs que désiré, sous forme d'arbre. Ces derniers peuvent être librement agencés, et réorganisés à volonté. Chaque bloc peut générer plusieurs blocs enfants, eux-mêmes librement déplaçables, sachant qu'au déplacement, un bloc parent sera toujours déplacé avec ses blocs enfants. Ce système permet de faciliter la tâche d'ordre, puisque modifiable à l'infini sans altérer le contenu des blocs, évitant ainsi de tout reprendre à zéro à chaque nouvel essai. Pour contourner l'absence de forme écrite pour les LS, ces blocs peuvent accueillir différents types de contenu: de l'écrit (input au clavier), des images, ainsi que des vidéos. Les vidéos peuvent provenir soit de liens extérieurs et être importées au sein d'un bloc, soit être directement filmées à l'aide d'un outil webcam intégré à l'application.

Pour conserver une lisibilité qu'importe le degré de profondeur de l'arbre de blocs, un encart de navigation permet de dérouler ou non chaque branche de sorte à pouvoir rejoindre aisément un nœud donné dans l'arborescence.

Modules de ressources: L'application accueille un volet d'accès à différentes ressources. Chaque onglet reprend un type de recherche: lexicale, encyclopédique, ainsi qu'une carte interactive. Chacun des onglets agrège les résultats de recherche au sein de plusieurs ressources différentes. Un historique de recherche global est disponible, de sorte à pouvoir rapidement réitérer une recherche antérieure.

Prompteur: Enfin, pour considérer le processus de traduction dans son ensemble et également assister l'étape de production filmée, nous envisageons une fonction de type prompteur à intégrer à l'interface. Celle-ci permettrait au traducteur, une fois ses blocs remplis et organisés à sa guise, d'en sélectionner plusieurs de sorte à les faire défiler selon un rythme et un mode de présentation sélectionné (défilement vertical ou diaporama).

La partie suivante s'intéresse à l'élaboration d'un prototype complet basé sur ce cahier des charges, fonctionnalité par fonctionnalité.

3 Environnement de traduction intégré

C'est en suivant le cycle de conception centrée utilisateur constitué de 5 étapes (investigation, idéation, prototypage, évaluation et production), et au travers de plusieurs itérations que nous avons donc procédé à la conception d'un logiciel de TAO destiné aux langues des signes. Une première phase d'investigation a été réalisée via les études préliminaires citées ci-avant. Le processus d'idéation quant à lui a été accompli en analysant les besoins des utilisateurs pour en retirer des exigences auxquelles devra répondre notre logiciel. La prochaine étape est celle du prototypage, que la sous-section suivante détaille.

3.1 Prototypage

Dans l'optique d'explorer différentes options de design pour illustrer les fonctionnalités découlant des exigences du cahier des charges, nous avons alors entrepris de réaliser des prototypes dits de basse fidélité. Prenant la forme de prototypes papier animés (fig. 2), ces derniers nous ont permis de concrétiser notre vision rapidement afin de pouvoir projeter et réfléchir sur le design envisagé.

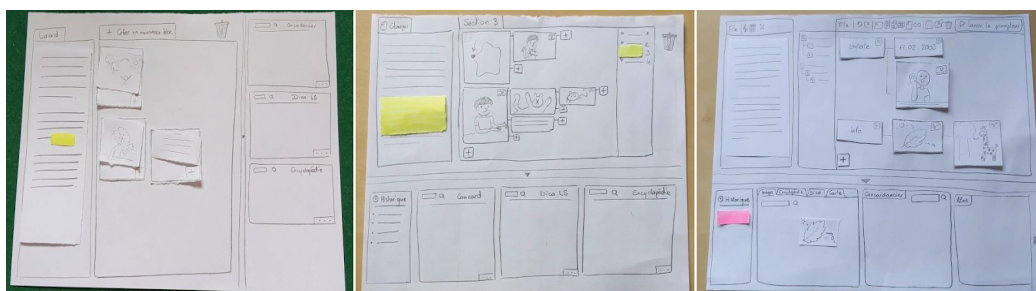


Figure 2: Premiers prototypes du système

Sur chacun de ces prototypes, nous illustrons les trois zones nécessaires à la TAO : la zone de texte source - dans laquelle on retrouvera le texte à traduire du français vers la LSF, la zone de travail principale - constituée de la zone à blocs évoquée précédemment, et enfin la zone des modules - qui constitue l'ensemble des ressources aidant à la traduction.

- Sur la gauche de l'interface, la zone de texte source permet d'afficher le texte à traduire. Des actions sont disponibles pour lancer des recherches dans les ressources depuis le texte, toujours dans l'objectif de simplifier les tâches récurrentes dans le processus de traduction (dans le cas présent, les différents types de recherche).
- Au centre de l'interface, il est possible de créer des blocs, de les remplir avec du contenu et de les déplacer suivant une structure d'arbre. Cette structure hiérarchique réorganisable répond au besoin des traducteurs de pouvoir réarranger leur discours aisément, compte tenu que cette phase de réorganisation représente 30% du temps total passé sur la traduction en plus d'être la seule tâche systématiquement observée.
- Sur la droite de l'interface, la zone des modules permet au traducteur d'accéder à différentes ressources pour l'aider dans sa traduction comme des dictionnaires en ligne ou des encyclopédies, ceci afin de répondre au besoin des utilisateurs de centraliser les ressources.

Nous avons exploré plusieurs options de design à travers chacun de ces prototypes. Après la réalisation de chaque prototype, nous avons mené plusieurs échanges avec des professionnels du métier dans le but d'obtenir des retours extérieurs sur le système, afin d'identifier des failles de design potentielles et les résoudre. En utilisant les heuristiques de Nielsen dictant les principes d'utilisabilité d'une interface homme-machine (Nielsen, 1990), nous avons ainsi procédé à plusieurs itérations successives de prototypage et d'évaluation.

En prenant l'exemple du premier prototype : après évaluation heuristique, nous avons réalisé que la position du bouton permettant de créer un bloc enfant (appelé bouton +) était visuellement peu logique (à l'intérieur du bloc parent). En l'état, le bloc enfant créé apparaît à un endroit différent du bouton ayant entraîné sa création. Cela pose un problème de visibilité de l'état du système, et c'est source potentielle de confusion pour l'utilisateur. Pour améliorer l'intuitivité du système, nous avons donc décidé de déplacer le bouton de création de bloc enfant à l'extérieur du bloc parent, et le placer à l'endroit où le bloc enfant sera créé pour une meilleure visibilité. Ce changement est visible dès le deuxième prototype.

Afin de créer un troisième prototype, nous avons demandé à une interprète en LSF de traduire deux textes en utilisant une structure de blocs pour prendre ses notes, à l'image de ce que serait capable de faire le logiciel. Cet exercice nous a apporté une nouvelle perspective, et nous a notamment permis de noter une autre faille de design présente dans le deuxième prototype. Dans ce dernier, la taille des blocs enfants est directement conditionnée à celle du bloc parent, la taille du bloc parent étant la somme de la taille de chacun de ses blocs enfants afin de montrer la hiérarchie les liant. Or, en pratique, comme on l'observe dans l'exercice réalisé par l'interprète, les blocs ayant le contenu le

plus riche - et occupant donc la plus grande place - se trouvent bien souvent à la profondeur la plus élevée. D'après ces observations et afin d'optimiser la place occupée par les blocs, il a donc été décidé de lier la taille des blocs aux contenus qu'ils abritent, et de symboliser la relation hiérarchique qui les lie à leur parent par leur position dans l'interface.

3.2 Implémentation

Une exigence importante du cahier des charges établi est de pouvoir fournir aux traducteurs un logiciel facilement accessible et ne nécessitant pas d'installation ou mise à jour avant utilisation, ceci afin de faciliter l'introduction d'un outil informatique à des utilisateurs cibles peu familiers avec ces derniers. Ainsi, implémenter le logiciel sous la forme d'une interface web permet aux utilisateurs d'y accéder en suivant un simple lien URL. Nous avons décidé de réaliser le développement avec le framework web Angular en raison de sa puissance et de sa modularité. Au terme de son développement, et une fois déployée en ligne, l'application sera gratuitement accessible.

4 Résultats

Après avoir présenté les différentes étapes de conception du système, nous pouvons nous intéresser à l'état actuel de celui-ci, après quelques mois de développement. Il a bien sûr pour vocation d'évoluer dans le temps, après de nouvelles itérations du cycle de conception, selon les retours donnés par les utilisateurs cibles notamment. La figure 3 montre l'aspect de l'interface lorsqu'un projet est en cours. L'Annexe 1 montre l'interface vide.

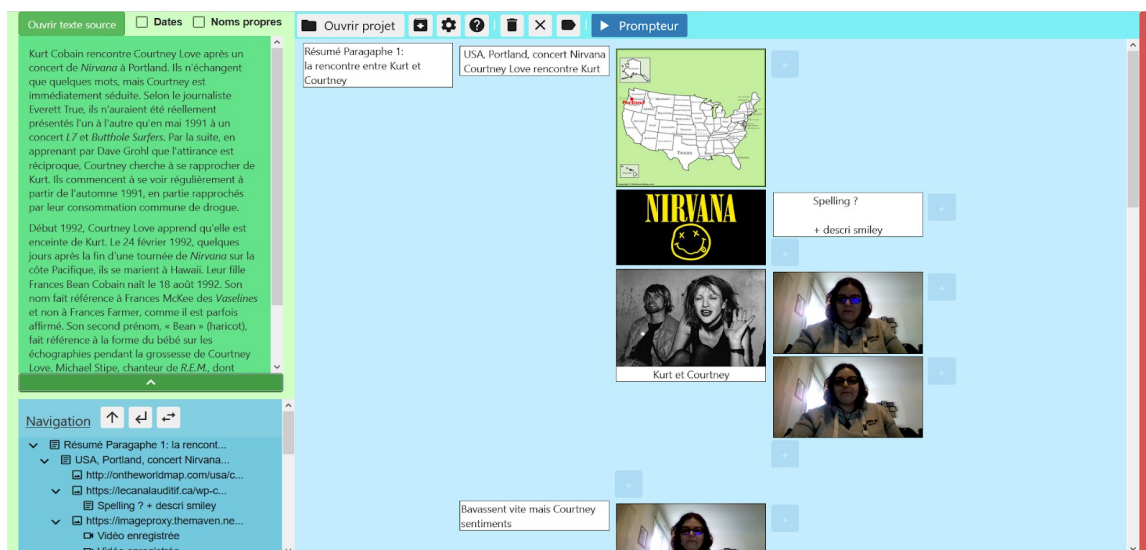


Figure 3: Capture d'écran de l'interface avec projet chargé

4.1 Liste des fonctionnalités implémentées

Voici la liste des fonctionnalités opérationnelles au moment de la rédaction:

Zone à blocs : Cœur de l'outil, la zone à blocs - au centre, en bleu sur la fig.3 - représente la zone d'édition. C'est là que l'utilisateur pourra créer de nouveaux blocs, les remplir avec du contenu de son choix, et les déplacer pour établir une hiérarchie libre à sa convenance. Cette zone permettant de prendre des notes, ainsi que d'insérer le résultat de différentes recherches réalisées au préalable par

l'utilisateur, remplit l'exigence d'avoir une structure modulaire, facilement réorganisable afin de faciliter la phase de réorganisation de la traduction, systématique et coûteuse en termes de temps (voir Annexe 4).

L'utilisateur peut créer un nouveau bloc en cliquant sur le bouton + lié au parent. Il lui est ensuite possible de le remplir avec différents contenus. Lorsqu'un bloc n'est pas encore rempli, des icônes permettant à l'utilisateur d'ajouter facilement du contenu sont affichées à l'intérieur, guidant le traducteur dans l'utilisation d'une structure peu habituelle. Ces icônes apparaissent à moitié fondues dans l'arrière-plan du bloc afin de ne pas les confondre avec du contenu réel que l'utilisateur aurait inséré par la suite (voir Annexe 3).

Actions sur les blocs : Il est possible d'ajouter du contenu au bloc sous plusieurs formes.

- Du texte, entré par l'utilisateur au clavier. Le traducteur peut ainsi entrer des notes dans des blocs, à l'image de posts-its. Couplé avec les autres types de contenu, ce texte peut également servir de légende.
- Des images, insérées via leur adresse URL ou à l'aide d'un drag and drop depuis l'image jusqu'à un bloc vide. Le traducteur peut ainsi insérer les images qu'il juge pertinentes à la préparation de sa traduction, l'aspect visuel des sujets de discussion étant primordial en LSF.
- Des vidéos, insérées depuis d'autres sites web via leur adresse URL, ou enregistrées par l'utilisateur lui-même par sa webcam, via un popup d'enregistrement dans l'outil. Cette fonctionnalité permet à l'utilisateur d'insérer des vidéos en ligne (Des définitions ou signes en LSF par exemple), ou de s'enregistrer soi-même, afin de prendre des notes directement en LSF, fonctionnalité sollicitée par les traducteurs interviewés lors des premières études.

Un seul bloc peut regrouper plusieurs contenus. Dans ces cas-là, les contenus sont réorganisables à l'intérieur d'un même bloc à l'aide d'un menu contextuel, si l'utilisateur veut par exemple afficher le texte au-dessus ou en-dessous de la vidéo. De même, il est possible pour l'utilisateur de supprimer ou de modifier individuellement chaque contenu du bloc, de supprimer tous les contenus afin de le vider tout en gardant la structure de l'arbre intacte, ou de supprimer le bloc-même ainsi que tous ses enfants. Ces deux dernières actions (suppression et vidage) impliquent l'effacement de contenus et/ou l'altération de la structure; elles sont donc accompagnées d'un avertissement pour s'assurer que leur déclenchement n'était pas accidentel.

D'autres actions sur les blocs sont possibles, comme la distribution du contenu d'un bloc parent vers des blocs enfants, ou l'inverse, laissant à l'utilisateur la possibilité de récupérer les contenus de tous les enfants et de les insérer dans le bloc parent.

Gestion du texte source : L'utilisateur peut charger un fichier .txt ou .docx. Une fois chargé, des fonctions simples de détection d'entités (noms propres et dates) sont disponibles. En effet, lors de l'observation des traducteurs professionnels, ces derniers avaient souvent le réflexe de surligner toutes les dates et/ou noms propres, et ce de manière systématique. Ces fonctions permettent donc d'automatiser la tâche. Actuellement, ces fonctions sont gérées par des expressions régulières, mais dans le futur, nous aimerions ajouter des fonctions de TAL plus poussées afin de détecter des expressions idiomatiques ou des expressions temporelles plus subtiles. (eg. "à la fin des années 80", "il y a 3 jours"...), mais aussi l'extraction automatique d'un résumé du texte source pour remplir un ou plusieurs blocs.

Modules ressources : Différents modules ressources sont accessibles depuis le logiciel, contenus dans un tiroir rétractable afin de laisser la zone principale d'édition lisible lorsque l'utilisateur n'a

plus besoin d'effectuer de recherches. Les modules actuellement disponibles sont : un module de recherche lexicale, un module de recherche encyclopédique, et un module de recherche cartographique (voir Annexe 2). Un historique commun à chacun de ces modules permet à l'utilisateur de retrouver facilement des recherches déjà effectuées, et permet de les relancer sans avoir à retaper la requête. Le module de recherche lexicale permet pour l'instant de regrouper les dictionnaires en lignes français-LSF Elix et SpreadTheSign (liens dans la sitographie). Le module de recherche encyclopédique contient Wikipédia. Dans le futur, l'utilisateur pourrait personnaliser ces modules lui-même et renseigner les sites qu'il aimerait voir accessibles. Le module de recherche cartographique contient une carte interactive sur laquelle l'utilisateur peut accrocher des épingles afin de sauvegarder certaines localisations. Il est également possible de prendre des captures d'écran pour ensuite les insérer dans des blocs de la zone d'édition. Cette fonctionnalité répond au besoin de l'utilisateur de pouvoir visualiser des lieux les uns par rapport aux autres. Des fonctions de TAL pourront ensuite permettre d'extraire les noms de lieux dans le texte source et les épinglez automatiquement sur la carte dès le chargement du texte.

Fonctions système : L'export de projet au format .ZIP permet à l'utilisateur de sauvegarder l'avancement de son projet pour conservation, ou pour le partager avec des collaborateurs. Les images et vidéos générées par l'utilisateur, telles que les captures de cartes ou les enregistrements à la webcam, sont enregistrés respectivement au format .PNG et .MP4. Une simple extraction du projet zippé permet alors à l'utilisateur de récupérer ces fichiers images et vidéos si jamais il désire y accéder en dehors du logiciel. Les ressources puisées en ligne par l'utilisateur (images tirées d'encyclopédie, vidéos tirées de dictionnaires...) ne sont pas sauvegardées dans le fichier du projet afin de ne pas en augmenter inutilement la taille, seuls les liens sont conservés. Cette fonction de sauvegarde répond au besoin des utilisateurs de pouvoir différer et partager leur travail.

Un tutoriel intégré donne à l'utilisateur la possibilité de consulter à tout instant un popup expliquant les différentes parties de l'interface, afin de faciliter la prise en main du logiciel.

4.2 Liste des fonctionnalités à implémenter

Le logiciel étant toujours en cours de développement, d'autres fonctionnalités sont envisagées. Ces dernières seront le fruit de nouvelles études avec les utilisateurs cibles. Certaines sont déjà prévues :

Concordancier intégré : Part importante de tout logiciel de TAO, la mémoire de traduction que l'on souhaite conserver pour les LS prend la forme d'un concordancier bilingue (Kaczmarek & Filhol, 2020), basé sur des brèves journalistiques. Celui-ci, actuellement accessible en ligne sur une plateforme dédiée, permet à l'utilisateur de faire des requêtes de mots ou ensemble de mots afin de pouvoir visualiser les extraits en LSF (en vidéo) en contexte. L'intégrer au logiciel de TAO pour les LS permettrait ainsi l'ajout d'une ressource lexicale supplémentaire.

Prompteur : Mentionné lors de la première étude, un prompteur aiderait l'utilisateur à la production finale de la traduction. La forme exacte de celui-ci est encore indéterminée, et sa définition nécessite de nouveaux entretiens avec les utilisateurs.

L'implémentation en ligne du prototype est en cours au moment de la rédaction. Il convient dès lors de s'intéresser à son évaluation. La partie suivante détaille les pistes envisagées dans ce sens.

5 Évaluation

L'évaluation du prototype n'est pas encore réalisée au moment de la rédaction, mais elle est prévue, auprès d'un public de professionnels. Elle comprendra deux volets: une partie subjective (ou qualitative), qui a pour objectif de récolter les avis des utilisateurs, et une partie objective (quantitative), qui nous permettra d'analyser et d'évaluer les performances de l'application d'un point de vue extérieur.

5.1 Méthodologie

Un seul protocole est envisagé pour les deux types d'évaluation. Ce dernier nécessite deux groupes de professionnels : un groupe témoin, qui n'utilisera pas l'application, et un groupe test qui lui devra l'utiliser lors de la tâche proposée. Ladite tâche consiste en la traduction de plusieurs textes, du français écrit vers la LSF, et se déroule en deux parties. La première concerne les deux groupes, à qui l'on demande de traduire trois textes, avec l'aide de l'application pour le groupe test, et sans pour le groupe témoin. Les textes sont identiques entre les deux groupes, tendent vers un style journalistique et comportent deux textes dits courts (d'une dizaine de lignes, dont un de rodage) et un texte long (environ une page et demie au format portrait). Ces textes ont été choisis pour leur complexité, et les besoins de recherches annexes qu'ils pourraient susciter. La tâche de traduction va de la découverte du texte à traduire à l'étape de production filmée.

La seconde partie ne concerne que le groupe test, à qui un accès à l'application sera fourni pour une période de trois semaines. Durant cette période, le groupe test sera encouragé à réutiliser l'application dans un contexte plus libre, si possible à plusieurs reprises, sans textes imposés

5.2 Évaluation subjective

L'évaluation subjective du prototype passe par la collecte de feedback auprès des utilisateurs. Pour ce faire, nous avons établi un questionnaire de type System Usability Scale (SUS, Brooke, 1986). Il consiste en une dizaine de questions dont les réponses seront un degré attribué sur une échelle de Likert à cinq niveaux (pas du tout d'accord, pas d'accord, neutre, d'accord, tout à fait d'accord). Le questionnaire est remis aux professionnels ayant fait usage de l'application directement à la fin de la première partie. Il interroge les éléments suivants: l'aisance de la prise en main de l'application, les fonctionnalités jugées les plus utiles par le traducteur, les faiblesses de l'application, la satisfaction générale de l'utilisateur avec l'outil. Le questionnaire est également soumis aux utilisateurs au terme de la seconde partie, après trois semaines d'utilisation libre, de sorte à pouvoir comparer les résultats, et évaluer un éventuel impact de la récurrence d'utilisation.

D'autre part, nous envisageons de faire évaluer les productions des deux groupes, témoins et tests, par un troisième groupe d'experts qui lui n'aurait pas eu à effectuer la tâche de traduction. Le groupe d'experts devra attribuer une note à chaque traduction produite lors de la première partie de l'évaluation, de sorte à pouvoir analyser l'impact de l'utilisation de l'application sur la qualité de la traduction. La notion de qualité de traduction étant elle-même subjective, il reviendrait au groupe d'experts d'établir au préalable une liste de critères fixes à appliquer à chaque traduction. Pour faciliter la tâche des évaluateurs, qui devront de fait annoter des vidéos pour évaluer les traductions, nous envisageons de leur fournir un manuel d'annotations commun pour lisser leurs retours

5.3 Évaluation objective

L'évaluation objective du prototype repose elle sur des mesures indépendantes des participants. La première concerne le temps. Chaque traduction pour chaque utilisateur est chronométrée, dans le but d'établir une moyenne par texte par groupe, et de pouvoir évaluer si oui ou non l'application offre un gain de temps notable lors du processus de traduction.

D'autre part, plusieurs compteurs de clics seront implémentés dans l'application. A chaque fois qu'un utilisateur utilise telle ou telle fonctionnalité du logiciel, le compteur incrémente de un. L'analyse de ces scores permettra d'évaluer la pertinence des fonctionnalités intégrées dans l'application, fonction de sa récurrence d'utilisation (comparaison des scores entre les différentes traductions). Un écart type permettra de lisser l'influence du paramètre personnel pour les cas où une minorité d'utilisateurs auraient généré la majorité des utilisations comptabilisées.

6 Conclusion

Cet article a abordé les points clés d'une TAO adaptée pour les langues des signes. Ayant d'abord pris connaissance des réalités du métier et de ses particularités auprès des professionnels concernés, ainsi qu'ayant brossé un tableau qui résume les grands principes de la TAO, nous avons pu nous atteler à marier les deux. De l'élaboration d'un cahier des charges à l'implémentation en ligne d'une application fonctionnelle, le développement d'un tel logiciel représente un enjeu tant traductologique qu'informatique puisque les contraintes posées de part et d'autre nous incitent à repenser un domaine déjà existant. Le métier n'étant actuellement pas outillé, il est apparu complexe pour nos collaborateurs traducteurs et interprètes en langues des signes de s'imaginer quels types d'assistance un outil informatique pourrait leur apporter. De ce fait, certains choix (de design ou de fonctionnalités) lors de la conception du logiciel sont des partis pris, mais ont toujours vocation à être discutés et évalués par les professionnels concernés: il était nécessaire de trancher, et de proposer des éléments concrets pour les donner à évaluer.

Le processus de conception étant un cycle, de prochaines itérations permettront de continuer à améliorer l'ergonomie et l'utilité du système. De nouvelles perspectives issues des retours des professionnels une fois le logiciel testé, ainsi que des données collectées au travers de nouvelles études nous permettront de poursuivre le développement d'une application qui se veut issue de la collaboration entre chercheurs et professionnels de terrain. Il s'agira notamment d'inclure de nouveaux modules de ressources et de nouvelles fonctionnalités tels qu'un prompteur ainsi que l'accès direct à un concordancier bilingue en ligne. Nous envisageons également d'ouvrir une plus grande place au TAL dans notre environnement de travail intégré. Des modules empruntés au TAL des langues écrites comme évoqué précédemment, telle que la génération automatique de résumés ou encore l'extraction de pourcentages et la génération automatique de diagrammes d'une part, mais également des modules du traitement automatique des langues signées (TALS). En effet, bien que récentes, ces études s'intéressent à la formalisation des LS et à l'élaboration de modèles descriptifs pour en faciliter le traitement automatique. Des fonctionnalités comme la reconnaissance automatique de signes isolés dans une vidéo, l'aide à la segmentation du flux et le sous-titrage automatique, ou l'anonymisation des signeurs permettraient à notre environnement de travail de pouvoir également s'utiliser dans l'autre sens de langues (LSF vers le français écrit).

Plus de contenus traduits induit une meilleure accessibilité pour les personnes signantes, ainsi qu'une visibilité accrue pour les langues des signes. Nous espérons que ces travaux trouveront une suite, et qu'ils susciteront l'intérêt de nouvelles sphères quant aux LS.

Références

- BROOKE J., (1986). "SUS: a "quick and dirty" usability scale". In P. W. Jordan; B. Thomas; B. A. Weerdmeester; A. L. McClelland (eds.). *Usability Evaluation in Industry*. London: Taylor and Francis.
- CUXAC C. (1993). « Iconicité des Langues des Signes. » in: *Faits de langues*, n°1, Mars 1993. Motivation et iconicité. p. 47-56.
- GUITTENY P. (2007) ; « Langue des signes et schémas » in : *Traitement automatique des langues*, Volume 48, n°2/2007, p. 1 à 10
- KACZMAREK M., FILHOL M. (2020). Alignments Data Base for a Sign Language Concordancer, in *proceedings of 12th International Conference on Language Resources and Evaluation (LREC 2020)*, p. 60696072
- NIELSEN, J., MOLICH, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April)*, p. 249-256
- PAIRE-FICOUT L, SABY L., Alauzet A. *et al.*, « Quel format visuel adopter pour informer les sourds et malentendants dans les transports collectifs ? », *Le travail humain*, 2013/1 (Vol. 76), p. 57-78. DOI : 10.3917/th.761.0057.
- SELESKOVITCH D., LEDERER M. (2014) « Interpréter pour traduire. » *5^e édition revue et corrigée, 2014*, Les Belles Lettres (1^{re} édition, 1984).
- WOLFE, R., ETHIMIOUS, E, GLAUERT, J., HANKE, T., MCDONALD, J., & SCHNEPP, J. (2016) eds. Special issue: recent advances in sign language translation and avatar technology, *Springer International Publishing*, 2016

Sitographie

Elix : <https://dico.elix-lsf.fr/>

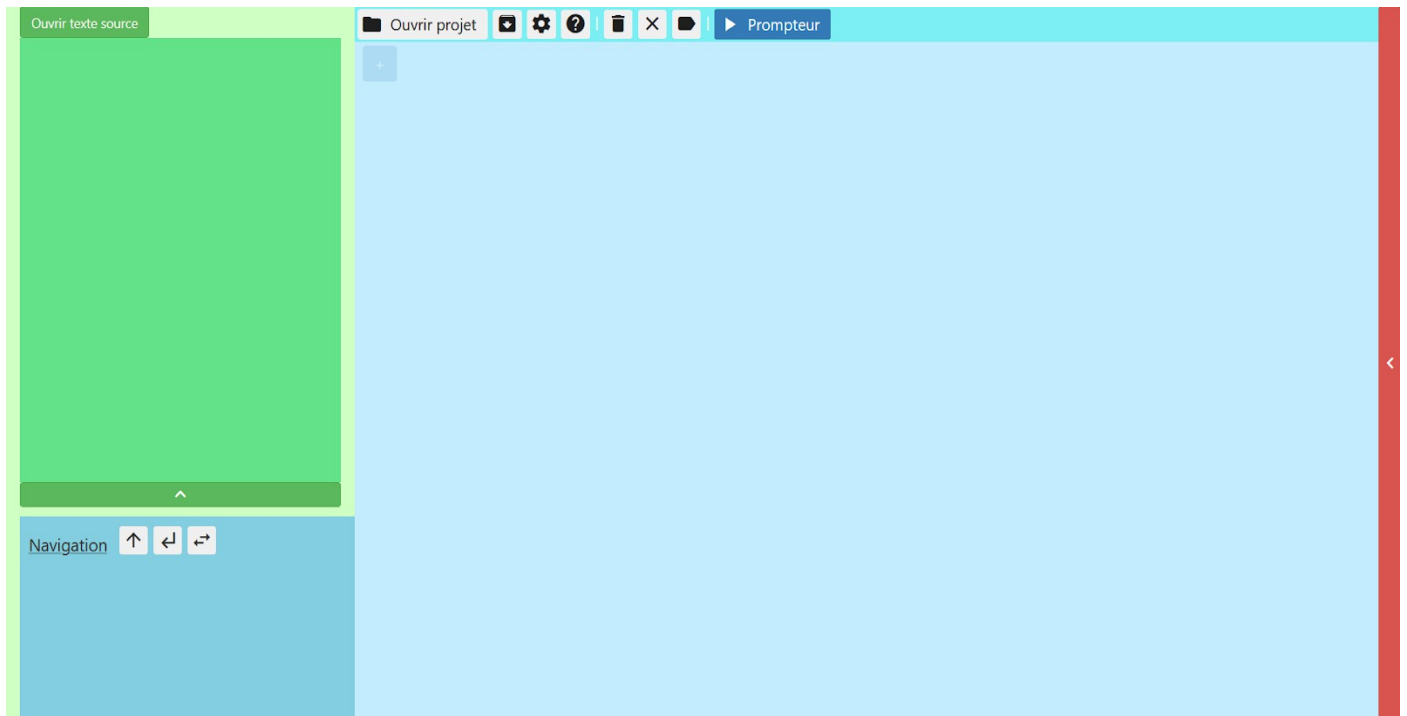
HamNoSys: <http://www.sign-lang.uni-hamburg.de/dgs-korpus/index.php/hamnosys-97.html>

SignWriting : <https://www.signwriting.org/>

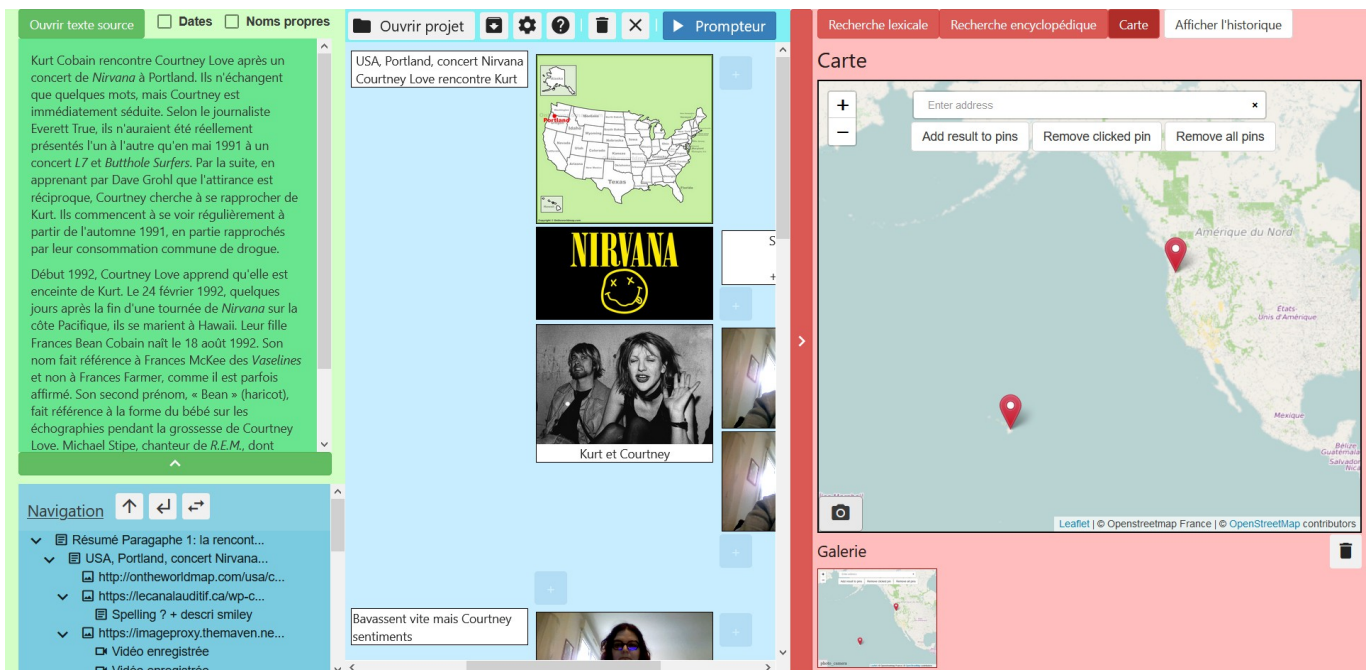
SpreadTheSign : <https://www.spreadthesign.com/fr.fr/search/>

Wikipédia : https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal

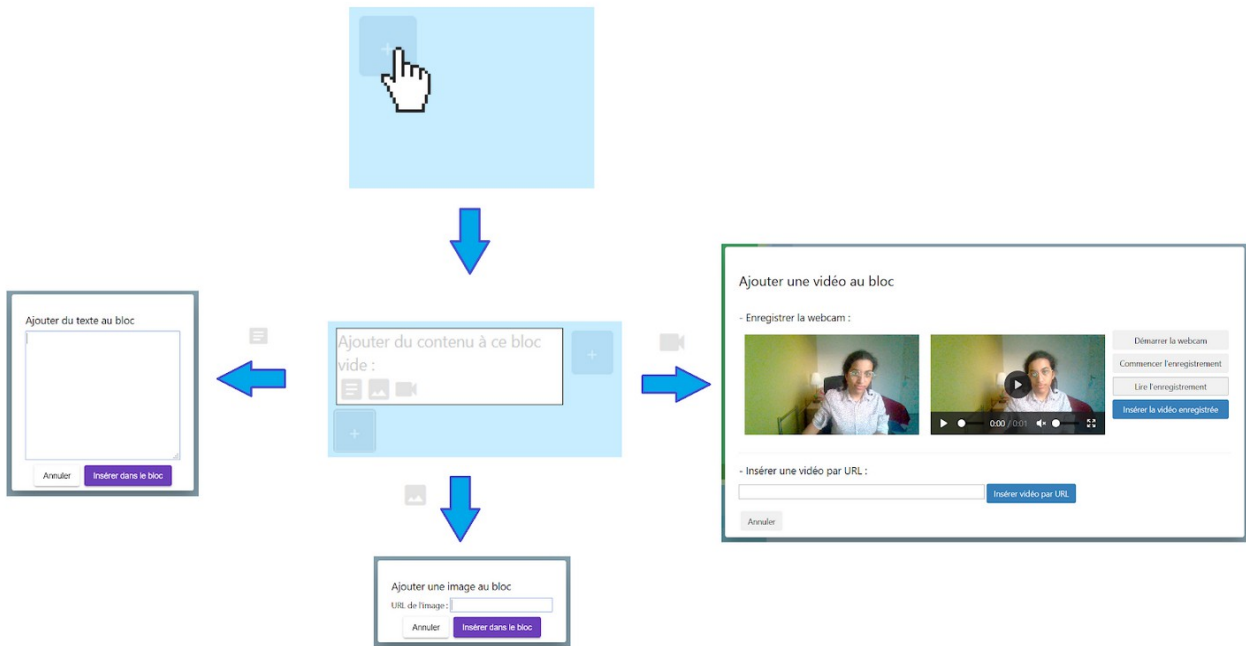
Annexe



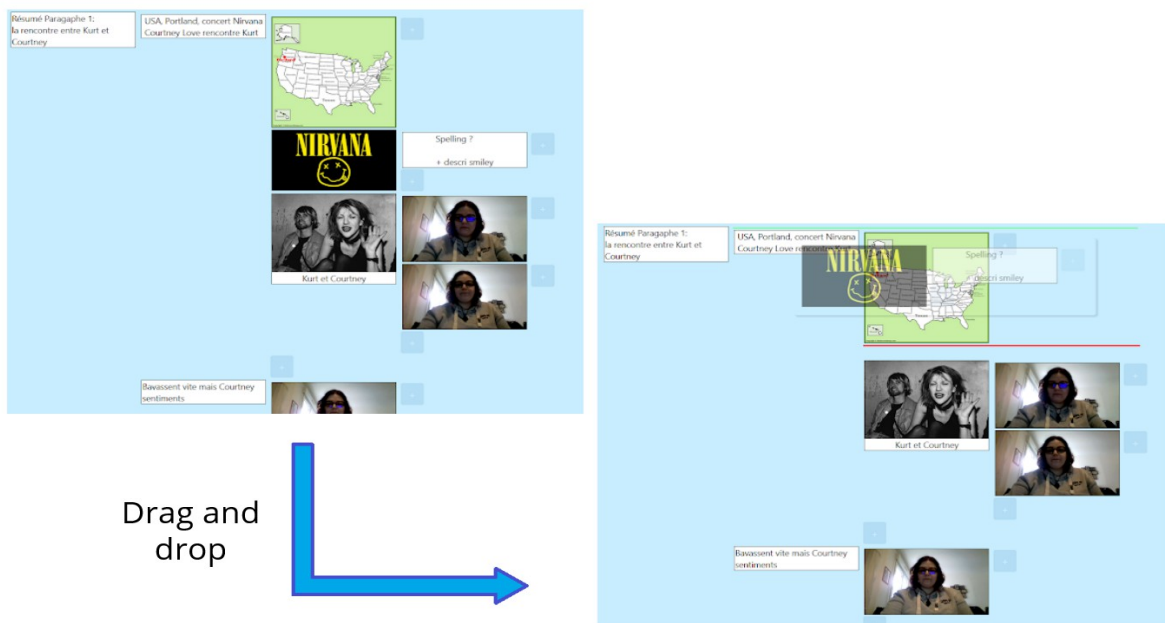
Annexe 1: Capture d'écran de l'interface à l'ouverture de la page



Annexe 2: Le module carte



Annexe 3: Différentes possibilités de contenu pour un bloc



Annexe 4: Le drag and drop permet de réagencer l'ordre des blocs