

RAPPORT DE VEILLE TECHNOLOGIQUE E

INTERFACES INTERACTIVES & NLP
POUR DES TRAVAUX SÉMANTIQUES

*VEILLE TECHNOLOGIQUE POURSUIVIE
PAR ÉRIC LAVOIE ET NADINE GIASSEN ST-AMAND
PROFESSEURS AU DÉPARTEMENT D'INFORMATIQUE DU CÉGEP DE MATANE*

*RAPPORT RÉDIGÉ PAR
NADINE GIASSEN ST-AMAND
B.ENG. COMPUTER ENGINEERING*

PRÉSENTATION DE L'ENTREPRISE ET DE SON PROJET

L'entreprise anime des séances de remue-méninge ("brainstorming"). Les informations sont placées dans des cases spécifiques sur un canevas et suite à des identifications de mots-clés et des associations d'idées un rapport multi-dimensions est généré pour le client.

Pour faciliter son travail l'entreprise souhaite numériser et automatiser plusieurs étapes du processus en implantant un logiciel comportant notamment une interface hautement interactive et des processus d'analyse intelligente des contenus.

OBJECTIFS SPÉCIFIQUES

TRAITÉS DANS LA VEILLE TECHNOLOGIQUE

Objectifs reliés à l'interface

- Dans l'interface de saisie de données, on souhaite pouvoir sémantiser le contenu en associant des mots-clés et des méta-informations à des contenus.
- Ces mots-clés créent des catégories virtuelles qui peuvent être visualisées et compilées sous forme de rapport. La sémantisation crée des axes d'analyse sous-lesquelles seront agrégées les informations d'une manière statistique.

Objectifs reliés au traitement intelligent

- On souhaite automatiser le travail d'analyse de la sémantique des contenus. On souhaiterait notamment détecter automatiquement des nuages de mots à partir des contenus et d'une base de données.
- On souhaite aussi dégager des contenus des tendances, des priorités et des pistes d'action.
- On souhaite automatiser la génération d'horaire selon la réponse à des questions sous la forme d'animateur virtuel

SOLUTIONS POSSIBLES

Différents choix technologiques sont possibles pour arriver au résultat escompté. Certaines pistes étaient déjà favorisées par l'entreprise.

Du côté Interface (aussi appelé côté-client dans certaines terminologies) on peut avoir un logiciel web ou non-web. Dans l'analyse l'hypothèse web a été privilégiée dans la recherche d'outils en raison de la grande interactivité de ceux-ci.

Du côté Intelligence (aussi appelé côté-serveur dans certaines terminologies) on peut utiliser un service d'intelligence ou une API distante (tel Google AI ou Amazon Comprehend) ou encore une librairie qu'on intègre directement à son code logiciel.

SOLUTIONS POUR LES INTERFACES

POSSIBILITÉS AVEC LES INTERFACES NON-WEB

Il est possible d'utiliser certaines bibliothèques interactives avec les interfaces non-web telles que LWJGL, JavaFX, QT et GTK. Mais il est aussi possible de transférer en application native les travaux réalisés avec des interfaces web en utilisant des framework tels que **Electron** ou **React Native**. On peut même les transférer sur des appareils mobiles avec les framework PhoneGap/Cordova. Il n'est donc peut-être pas nécessaire d'explorer plus loin les bibliothèques interactives non-web puisque les choix technologiques et les travaux web pourront être transposés en natif si nécessaire grâce à ces outils. (voir [Annexe 2 - Interface et interactivité en contexte non-web.pdf](#)) De plus les bibliothèques web vont plus loin en interactivité au grand bénéfice de l'entreprise.

POSSIBILITÉS AVEC LES INTERFACES WEB

Depuis l'amélioration des moteurs de rendu web presque tous basés sur WebKit devenu Blink, une certaine portabilité des travaux d'interface web est possible et il est possible de recommander sélection de bibliothèque JavaScript pour le développement des outils de l'entreprise.

Plusieurs bibliothèques d'interactivités ont été étudiées par la veille technologique pour le drag&drop telles que Dragula, Displace, Sortable, Draggable et Draggin. (Voir [Annexe 1 - Interface & Interactivité avec les technologies web.pdf](#))

CHOIX DU VEILLEUR TECHNOLOGIQUE DU CDRIN

Les critères utilisés pour choisir une technologie comprennent l'atteinte de l'objectif d'interactivité mais comporte aussi les critères de portabilité navigateur et aussi d'absence d'invasivité du code HTML pour une meilleure maintenabilité applicative.

Enfin un dernier critère est la minimisation des dépendances. Par exemple une bibliothèque implémentant le drag and drop par elle-même serait supérieure à celle dépendant d'une version particulière de JQuery, celle-ci pouvant être incompatible avec une autre version de JQuery nécessaire pour un autre outil de la page. De plus une bibliothèque sans dépendances est moins lourde (et moins nuisible en SEO).

- Tel que demandé par le client projetée pour le nuage de mot-clés & le drag and drop est Muuri car elle combine à la fois
- Outil suggéré pour le transfert en application native : <https://electronjs.org/> avec NodeJS

SOLUTIONS POUR LES ANALYSES INTELLIGENTES

FAISABILITÉ EN SE BASANT SUR LES EXEMPLES D'APPLICATIONS EXISTANTES

Il existe plusieurs exemples d'applications actuelles se rapprochant des besoins du client (Voir [Annexe 3 - Applications du NLP.pdf](#)). Par exemple certaines applications du crowdsourcing telles que l'analyse de réponses à des sondages et la compilation des commentaires d'appréciations de clients ressemblent à certains besoins de l'entreprise. Certaines applications de la veille technologique telles que générer des mots-clés automatiquement et classifier les éléments de textes sont presque exactement la même chose que le premier besoin en NLP. La prédiction de tendances futures pourrait être assimilée à la prédiction de pistes d'action. On conclut donc que les algorithmes et API NLP sont effectivement ce dont l'entreprise a besoin pour mener à bien ses automatisations.

SOLUTION DE CHOIX D'ALGORITHMES

Les algorithmes NLP sont devenus une branche de l'intelligence artificielle tellement c'est la seule manière de faire utilisée de nos jours. (Voir [Annexe 4 - Algorithmes du NLP.pdf](#)). Ceux-ci comprennent des algorithmes d'analyse syntaxique qui préparent les données et des algorithmes d'analyse sémantiques qui sont le nerf de la guerre. Parmi les algorithmes d'analyse syntaxique on retient le fameux Étiquetage contextuel (Part-of-speech tagging) qui semble nécessaire mais qui ne s'utilise pas seul. La lemmatisation, la racinisation et possiblement la segmentation en morphème sont triviaux et préparatoires et devront être utilisés de toute manière.

En ce qui concerne le choix de l'algorithme d'analyse sémantique, la plupart des algorithmes mentionnés (sauf Jaccard évidemment) sont utilisables pour atteindre le but recherché. Le choix de l'algorithme exact pourra faire l'objet d'un travail plus approfondi d'essai et d'erreur avec des données types lors des travaux préliminaires de développement. On repère quand même les techniques de k-moyennes, de SIF (Smooth Inverse Frequency), de Similarité de cosinus et WMD (Word Mover Distance) comme étant intéressantes. Mentionnons cette source pour les algorithmes de mesure de degré de similarité dans des textes : <https://medium.com/@adriensieg/text-similarities-da019229c894>

SOLUTION DE CHOIX D'APIS

Différentes API ont été repérées (Voir [Annexe 5 - Les APIS de programmation NLP.pdf](#)). Parmi celles-ci, celles qui retiennent notre attention sont d'une part les API des grands joueurs de l'infonuagique qui possèdent de grande quantité de données et de pouvoir de calcul ([Google Natural Language API](#) et [Amazon Comprehend](#)), d'autre part les API Open-Source pour la pérennité ([Apache OpenNLP](#), [Stanford CoreNLP](#) en Java et [NLTK](#) en python). Enfin, dans les API web, [AutoTag](#) est intéressant pour générer des tag automatiquement dans un texte.

CHOIX DU VEILLEUR TECHNOLOGIQUE DU CDRIN

C'est Google Natural Language API, Apache OpenNLP et AutoTag qui seraient les premiers sur le banc d'essai pour leur gratuité, leur qualité et leur promesse de pérennité si le projet démarrait ici.

SOLUTIONS D'ARCHITECTURES

Afin de rendre le tout flexible et robuste, et de pouvoir faire des mises à jour sur l'intelligence sans toucher à l'interface, et inversement de pouvoir faire des mises à jour sur l'interface sans toucher à l'intelligence, certaines architectures sont recommandées.

Ces hypothèses d'architecture supposent qu'un service d'intelligence distant sera adopté.

Abbréviations:

MVC = Modèle Vue Contrôleur

SOA = Service Oriented Architecture

DAO = Data Accès Object

ARCHITECTURE MVC EN COMMUNICATION DIRECTE

Est-il possible d'utiliser une architecture Modèle-Vue-Contrôleur et de communiquer directement avec le service de données tout en utilisant un service distant d'intelligence et une interface de type web ?

Oui il est possible de faire ceci puisque les API de plusieurs fournisseurs d'intelligence ont des interfaces de communication JavaScript. Par exemple [TensorFlow.js](#) communique avec Google AI pour les gros traitements.

ARCHITECTURE SOA AVEC DAO AVEC OU SANS MVC

La SOA consiste à organiser une interface de question-réponse en format de requêtes web avec lesquelles les réponses sont les données traitées en format XML ou JSON au choix. Il serait possible donc de découper les traitements intelligents de l'entreprise de leurs interface de manière à pouvoir d'une part :

- pouvoir réutiliser ces algorithmes dans plusieurs applications différentes
- pouvoir sauvegarder les pré-traitements dans une base de données centrale
- pouvoir mettre à jour facilement les interfaces sans toucher aux traitements - ce qui est un requis très important étant donné la fluidité et l'instabilité des bases de code javascript d'interface
- pouvoir conserver côté service le secret professionnel des algorithmes intelligents

Est-il possible d'utiliser une architecture de service, c'est-à-dire d'isoler le traitement des données de leurs interfaces à l'aide d'un service ?

La réponse est oui peu importe la technologie. D'une part si une technologie non-web est utilisée pour implémenter ces algorithmes, il est toujours possible d'utiliser une passerelle CGI ou une de ses [alternatives](#) (Mojolicious, Dancer2, Catalyst).

CHOIX DU VEILLEUR TECHNOLOGIQUE DU CDRIN

L'architecture SOA est recommandée pour ce genre de travaux afin de garantir la robustesse, de faciliter la division du travail d'équipe, d'améliorer la maintenabilité des logiciels et pour faciliter les nouveaux développements. **Il serait donc conseillé d'implémenter les détections de nuages de mots-clé sur un service web de données complètement séparé de l'application.** Il en est de même pour les détection de tendances et les pistes d'action.

Rapport préparé le 27 septembre 2019 par Nadine Giasson St-Amand pour le CDRIN
à partir de la veille technologique sur NLP (par Nadine Giasson St-Amand) et de la veille technologique
sur les interfaces interactives (par Éric Lavoie).