Université de Franche-Comté

**Modélisation de questions pour un système QR dans le domaine de la santé.**

Par Ana Calatayud

Master Langues, Littératures et Civilisations Etrangères et Régionales (LLCER)

Parcours Traitement Automatique des Langues (TAL)

Directrice de mémoire : Dr. Iana Atanassova

Mémoire présenté à la Faculté des Sciences du Langage, de l’Homme et de la Société

25 mai 2018



Remerciements

**A mes professeures, Iana et Izabella**

Merci à Izabella Thomas et à Iana Atanassova pour leur enseignement. Un merci tout particulier au Dr. Atanassova d’avoir accepté de diriger ce mémoire et de m’avoir guidé, conseillé et encouragé.

**A mes parents, Christophe et Carole**

Même profanes, vous vous êtes intéressés et vous m’avez toujours encouragé. Votre soutien a été un moteur dans la réalisation de ce mémoire et j’espère vous rendre fiers. Sachez que ma gratitude et mon amour sont inconditionnels.

**A ma sœur, Aude**

Tu as toujours été présente, tu m’as toujours soutenu et encouragé. Sache que quand ton tour viendra ça sera réciproque. Merci à toi d’être une sœur géniale.

**A toi Léo**

Merci de m’avoir rassuré dans les moments de doute, de m’avoir poussé vers le haut quand j’avais envie de baisser les bras, de t’être intéressé bien que ça ne soit pas ton domaine, merci pour tout.

**A mes camarades, Yağmur, Elodie et Salah**

Bien plus que des camarades de classe vous êtes devenus des amis. Merci à vous d’avoir été là, merci de m’avoir soutenu. Ce mémoire, je le dédis à la promotion soudée que nous sommes.

**A Nathan**

Merci de m’avoir poussé dans cette voie, de m’avoir conseillé et de m’avoir aidé.

Table des matières

[I. Introduction 4](#_Toc514355990)

[1. Les systèmes question-réponse 4](#_Toc514355991)

[2. Problématique et objectifs. 6](#_Toc514355992)

[3. Etat de l’art : SQR dans le domaine de la santé. 6](#_Toc514355993)

[4. Concepts clés du sujet 9](#_Toc514355994)

[II. Méthodologie pour la modélisation des questions pour un système QR 11](#_Toc514355995)

[5. Hypothèses de recherche 11](#_Toc514355996)

[6. Domaine 13](#_Toc514355997)

[7. Ressources 14](#_Toc514355998)

[1) Corpus de questions 14](#_Toc514355999)

[2) Corpus de réponse 15](#_Toc514356000)

[8. Types de questions 15](#_Toc514356001)

[1) Les questions fermées. 15](#_Toc514356002)

[2) Les questions ouvertes 17](#_Toc514356003)

[9. Première modélisation 17](#_Toc514356004)

[1) Présentation de NLTK, Stanford CoreNLP et TreeTagger 18](#_Toc514356005)

[2) Segmentation. 18](#_Toc514356006)

[3) Lemmatisation. 19](#_Toc514356007)

[4) Analyse des questions 22](#_Toc514356008)

[III. Annexes et bibliographie 24](#_Toc514356009)

# Introduction

## Les systèmes question-réponse

De nos jours, l’accès à l’information est rendu aisé grâce à internet. On peut cependant aussi l’envisager comme un ‘cadeau empoisonné’ dans la mesure où la quantité de données qui y transitent est titanesque. Les moteurs de recherche tels que Google ou Yahoo ne facilitent pas la tâche puisqu’une simple requête engendre des milliers de résultats. L’utilisateur doit alors perdre un temps considérable à chercher la réponse à sa requête dans cet amas de données hétérogènes. C’est dans ce contexte que s’inscrivent les systèmes de question réponse (SQR). Un système question réponse permet à un utilisateur de formuler une requête en langue naturelle. Cette requête est ensuite analysée, puis des réponses potentielles sont recherchées dans un corpus composé en général de données et de supports hétérogènes. La réponse la plus pertinente est ensuite extraire puis retournée à l’utilisateur, le plus souvent sous forme de phrase formulée en langue naturelle.

Corpus

Construction de requêtes

Loupe

Analyse de la question

Recherche de mots-clés

Extraction des réponses candidates

Classement des réponses candidates et synthèse

*Figure 1. Architecture basique d’un SQR.*

Les premiers systèmes de question-réponse, BASEBALL (Green et al. 1961), programmé pour répondre à des questions concernant le champion de baseball américain d’une certaine année et LUNAR (Woods), programmé pour répondre à des questions concernant la lune, furent développés autour des années 1960-1970. Ils possédaient cependant de nombreuses restrictions sur la manière d’utiliser la langue, mettant ainsi en question leur qualification de SQR puisque les requêtes doivent être formulées en langue naturelle. Les SQR font cependant un véritable bond en avant grâce aux conférences TREC (*Text REtrival Conference*) qui ont débuté en 1992 et qui s’intéressent à l’extraction d’information, un élément essentiel des SQR. Ce n’est cependant qu’en 1999 que ces conférences ont commencé à intégrer des campagnes d’évaluation de SQR. Grâce à ces conférences, de nombreux systèmes comme QRISTAL (Laurent and Séguéla 2004), QALC (Ligozat 2004), FIDGI (Moriceau and Tannier 2009) ou encore PiQASso (Antonio et al. 2001) ont vu le jour. Ces systèmes sont à domaine ouvert (*open-domain*), c’est-à-dire qu’ils peuvent répondre à n’importe quelle question générique, comme la date de naissance de Shakespeare ou bien la capitale de l’Ouzbékistan.

Il existe cependant des systèmes question-réponse à domaine restreint (*domain-dependent*). L’un des domaines encore peu pourvu en matière de SQR est celui de la santé et du médical. L’une des raisons qui pourrait expliquer ce vide est que le domaine de la santé est un domaine sensible dans le sens où les requêtes formulées doivent obtenir des réponses précises, non ambigües et surtout correctes. Par exemple, avec un système QR qui ne ferait qu’une analyse superficielle et rudimentaire, une requête telle que « qu’est-ce que l’anémie ? » recherchée dans un corpus qui comporterait la phrase « […] causé par le manque de magnésium ou l’anémie » obtiendra potentiellement comme réponse « un manque de magnésium », ce qui n’est évidemment pas correct et ce qui pourrait induire l’utilisateur en erreur. Une autre raison pourrait être celle de la difficulté de constituer une base de données suffisante et fiable dans un domaine restreint tel que cela là. Depuis le début des années 2000 cependant, quelques SQR dans ce domaine ont vu le jour.

## Problématique et objectifs.

Dans le contexte des systèmes de question-réponse, notre problématique s’articule autour de la modélisation des questions, c’est-à-dire le processus qui consiste à représenter la sémantique des questions sous une nouvelle forme afin de rendre possible la recherche de la réponse dans des corpus textuels. Cette étape précède celle de l’interrogation du corpus. Ainsi, nos recherches s’orientent vers deux des modules qui composent en général un SQR : la normalisation de la question et le processus d’analyse défini par (Embarek, 2008).

Notre objectif final est de proposer un modèle théorique et d’implémenter un système de question-réponse à destination de l’Etablissement Français du Sang. Ce système devra répondre aux questions des usagers, donneurs potentiels. Nous considérons que les questions seront posées sous formes écrites en langue naturelle dans une interface dédiée. Les informations permettant de formuler les réponses seront extraites par le système à partir de bases de données textuelles et autres ressources disponibles en ligne. Nous supposons que les utilisateurs du système QR n’auront pas de connaissances préalables ou particulières dans le domaine médical. Ainsi, les réponses du système devront être vulgarisées afin d’être accessible pour un large public.

## Etat de l’art : SQR dans le domaine de la santé.

Quand il s’agit de rechercher des informations dans le domaine médical, internet doit être utilisé avec précaution. En effet, l’utilisateur lambda n’est la plupart du temps pas à même de juger de la pertinence et de la fiabilité des informations médicales qu’il peut trouver en ligne. C’est sur cette problématique que la fondation suisse *Health on the Net Foundation[[1]](#footnote-1)* s’est penchée. (Cruchet, Gaudinat, and Boyer 2008) développent un SQR dans ce domaine. Leur hypothèse est qu’une question sur le thème de la santé est caractérisée par son type médical (symptômes, traitement, causes etc) et par le type de réponse attendu (booléenne, causale, spatiale etc). Le corpus utilisé est une centaine de questions collectées via des FAQ (*Frequently Asked Questions*) de divers forums spécialisés dans la santé. Contrairement aux SQR déjà existants, les questions n’ont pas été analysées suivant un *pattern-matching* mais par un apprentissage automatique supervisé, où des experts ont classifié une partie des questions pour créer le corpus d’entraînement. Aucune comparaison des deux méthodes n’est cependant faite ; on ne sait donc pas pourquoi celle-ci a été privilégiée. Un problème majeur a également été rencontré : celui de la taille du corpus. Dans le corpus d’entraînement, certaines classes étaient fortement sous-représentées par rapport à d’autres, ce qui a faussé leur classification.

Une approche concernant la détection de questions similaire développée par (Abacha and Dina 2016) repose sur la reconnaissance de *text entailement*, c’est-à-dire de la relation directe entre deux segments de textes dont le sens de l’un est contenu dans le sens de l’autre.

Q1 (question d’un utilisateur) « A partir de quel âge est-ce que je peux donner mon sang ? »

Q2 (FAQ) « Puis-je donner avant mes 18 ans ? »

Q1 ↔ Q2.

Ce modèle de détection de la similarité est selon nous pertinente dans le sens où les questions posées par des utilisateurs dans des FAQ ou des forums sont rarement directes, peuvent être grammaticalement ou orthographiquement incorrectes et peuvent prendre plusieurs tournures différents (« est-ce que », inversion sujet-verbe etc).

(Rinaldi et al. 2004) ont quant à eux adapté un système QR déjà existant, à la base développé pour un domaine restreint, au domaine de la génomique. Leurs recherches mettent en lumière le problème de l’accès et de l’exploitation de données scientifiques, et notamment le fait qu’on ne puisse pas s’attendre à ce que les utilisateurs d’un SQR à domaine restreint soient à l’aise avec la terminologie du domaine en question. Ainsi, leur système ExtrAns est dédié aux systèmes restreints et possédant une terminologie conséquente. ExtrAns a pu être adapté au domaine de la génomique grâce à la collection et à l’analyse de documents traitant cette problématique. Le problème de la terminologie a pu être contourné grâce à l’annotation manuelle d’éléments terminologiques, permettant ainsi une analyse grammaticale correcte. Le domaine du don du sang étant lui aussi restreint, ces recherches vont nous permettre d’aborder (sinon de résoudre) le problème de la terminologie.

(Niu et al. 2003) s’intéressent quant à eux aux différences entre les systèmes question-réponse à domaine ouvert et ceux dédiés au domaine médical. Les systèmes généraux se concentrent majoritairement sur ce que l’on définit dans la langue anglaise comme étant les *wh-questions.* Cet ensemble de questions équivaut plus ou moins en français à la méthode QQOQCCP: Quoi, Qui, Où, Quand, Comment, Combien, Pourquoi, chaque mot interrogatif désignant un type de question qui peut être posée. Dans les systèmes QR du domaine médical en revanche, l’emphase est mise sur les traitements d’une maladie, les symptômes, les effets secondaires d’un traitement etc. Ainsi, l’identification du type de question doit se faire différemment. De plus, l’entité nommée *quand* n’aura pas la même signification dans un système général que dans un système médical.

SG : *quand a eu lieu la bataille d’Austerlitz ? Le 02 décembre 1805* → date précise

SM : *quand est-ce que je pourrais redonner ? dans 2 mois si vous êtes un homme, 3 si vous êtes une femme* → durée relative

Ils évoquent également le problème des sources de données. Pour qu’un système général soit performant, la réponse doit être validée grâce à plusieurs réponses candidates qui sont redondantes dans la base de données. Cette dernière doit contenir des informations provenant de plusieurs sources fiables. Dans un système médical en revanche, il est très peu probable que la même information soit présente à plusieurs endroits dans la base de données. Cette assertion est à relativiser dans le cas de notre système.

## Concepts clés du sujet

Dans un système question-réponse, la question doit tout d’abord passer par un processus de normalisation. (Jurafsky and Martin 2017) définissent trois étapes indispensables à la normalisation de texte.

* La segmentation (ou tokenisation) des mots. Selon (Clark, Fox, and Lappin 2010), il s’agit de ‘casser’ les séquences de caractères présentes dans un segment de texte afin de localiser les limites d’un mot. Dans cette optique, deux types de langues se distinguent : celles séparés par un espace et celles qui ne le sont pas. Cette approche met en lumière le problème de la ponctuation (par exemple les mots séparés par un tiret, notamment dans les structures interrogatives) et celui des expressions comportant plusieurs mots.
* Normaliser les tokens présents dans le texte, c’est-à-dire le processus de lemmatisation, défini par (Clark, Fox, and Lappin 2010) comme étant la détermination de la racine d’un mot.
* La segmentation des phrases. Ce processus tend à poser un problème en cas de présence d’abréviations puisque le point pourrait alors être mal analysé.

Une fois le texte normalisé, il passe par un processus d’analyse qui comprend les étapes suivantes, définies par (Embarek, 2008).

* Le type de question. Il permet de classifier les questions (type booléen, définitoire etc). Ainsi, la question « comment se déroule un don ? » sera étiquetée comme question procédurale tandis que la question « pourquoi ne peut-on pas donner en dessous de 50kg ? » se verra attribuer la catégorie de question causale.
* La reconnaissance d’entités nommées. (Omrane, Nazarenko, and Szulman 2009) définissent les entités nommées (EN) comme des éléments qui soulignent les fonctions référentielles de certains segments de texte. Ainsi, les EN agissent en quelque sorte comme des noms propres puisqu’elles font référence à des lieux, des personnes ou encore des dates.
* Le type de réponse attendue. (Mendes and Moriceau 2004) ont dégagé dix types principaux, à savoir (1) vériconditionnelle, (2) catégorie conceptuelle, (3) quantitative, (4) qualitative, (5) procédurale, (6) définition, (7) comparaison et/ou évaluation, (8) causale, (9) but et (10) description. (Li and Roth 2006) ont quant à eux défini une taxonomie comprenant cinq grandes catégories et cinquante classes plus précises (voir annexes).

Le type de réponse attendu dépend évidemment du typage de la question et des entités nommées présentes dans la question.

* Le focus de la question. Il permet de savoir sur quel élément on doit baser la réponse. Par exemple, « quel est le poids minimal requis ? » aura comme focus le segment textuel *poids*.

Toutes ces étapes utilisent différents outils tels que des *parsers*, des *pattern-matchers*, des *POS-taggers* ou encore des annotations sémantiques, ces processus pouvaient varier en fonction du système développé.

Dans le schéma de traitement général adopté par la majorité des systèmes question-réponse, une fois la question finement normalisée et analysée, le système interroge un corpus qui est généralement composé de plusieurs bases de données hétérogènes afin d’en extraire les passages contenant potentiellement la réponse à la question. Ces passages vont ensuite être segmentés afin d’analyser les phrases une par une et de les classer Le classement peut être effectué selon différents critères, dont une liste non-exhaustive est présentée par (Jurafsky and Martin 2017).

* Le nombre d’entités nommées du bon type dans le passage (si la question attend une réponse de type DATE, les passages du corpus contenant le même type d’entité seront extraits comme des réponses potentielles)
* Le nombre de mots-clés de la question contenu dans le passage.
* La plus longue séquence de mots de la question qui apparaît à l’identique dans le passage (Qutilisateur : « **la prochaine collecte de sang a lieu** quand ? » / Pcorpus : « **la prochaine collecte de sang aura lieu** le [date] »)

La réponse classée première en fonction du ou des critères choisis est alors extraite du corpus puis renvoyée à l’utilisateur.

# Méthodologie pour la modélisation des questions d'un système QR dans le domaine médical

## Hypothèses de recherche

L’une de nos premières hypothèses est que l’orthographe joue un rôle capital dans un système question-réponse. Ceci est d’autant plus important que le système que nous cherchons à construire est à destination du grand public et que les utilisateurs sont susceptibles, en général, de saisir les questions avec des fautes d’orthographe. Si le segment textuel saisi par l’utilisateur comporte des erreurs de graphie, le module de lemmatisation ne parviendra pas à ramener les mots sous forme de lemmes. Dans ce cas, la requête sera mal interprétée, la question mal modélisée, et la réponse extraite ne correspondra certainement pas à la question posée par l’utilisateur. Pour pallier ce problème, l’utilisation de la distance de Levenshtein est envisagée afin de pouvoir corriger la saisie utilisateur durant l’analyse de la question.

Notre deuxième hypothèse est qu’une segmentation ainsi qu’une lemmatisation automatique et des analyses morpho-syntaxique, en utilisant des librairies logicielles existantes, produira des résultats satisfaisants. Nous avons prévu d’utiliser trois modules, NLTK[[2]](#footnote-2), Stanford CoreNLP[[3]](#footnote-3) et TreeTagger[[4]](#footnote-4), implémentés en Python et Java. Les deux premiers modules permettent notamment de segmenter, d’étiqueter, de lemmatiser, d’identifier les entités nommées et de construire des arbres de dépendance, ce qui permettra un gain de temps considérable, tandis que le troisième module permet d’étiqueter les *parts-of-speech* et d’obtenir les lemmes.

Notre troisième hypothèse est que pour analyser le corpus de questions, la méthode de modélisation la plus efficace sera celle du *pattern-matching* (filtrage par motif en français). Il s’agit, suite à l’analyse d’une question, de lui attribuer un ‘motif’ défini au préalable. Ces motifs vont dépendre du type de la question, de la présence d’entités nommés, de la polarité de la question (fermée/ouverte) etc. Les différents motifs possibles et les types de réponses correspondantes seront définis dans notre méthodologie. En plus du filtrage par motif, une méthode d’apprentissage automatique supervisée pourra être envisagée pour la modélisation des questions.

Un problème important est l’extraction des informations relatives à la question à partir de la base de données textuelle. Une fois la question modelisee, il s’agit d’identifier les segments textuels disponibles dans le corpus qui portent des informations pertinentes. Pour cette tache nous proposons d’utiliser un procede inspire du domaine de la recherche d’informations, notamment le modele vectoriel (citer Salton, je ne sais plus quelle annee – verifier sur google scholar) avec ponderation des termes TD-IDF.

Nous proposons donc, à partir d’un corpus de questions posées par des utilisateurs, d’étudier les différents types de questions, structures linguistiques et motifs, afin de construire un algorithme qui permet d’obtenir le ‘motif’ d’une question sous une forme que nous allons définir par la suite. Ce ‘motif’ devra contenir des informations sur le type de la question, le type de réponse attendue, le focus de la question, les entités nommées présentes dans la question ainsi que d’autres informations nécessaires afin de construire une réponse pertinente.

## Domaine

Nous avons choisi de travailler sur les questions-réponses dans le domaine du don de sang, en s’appuyant sur les informations fournies par EFS en France. Cet objectif nous tient à cœur car selon nous, l’une des raisons qui pourrait expliquer la baisse des réserves de sang en France est la mésinformation des citoyens. Grâce au système de question-réponse que nous produirons, nous espérons simplifier l’accès aux informations concernant le don du sang et ainsi inciter plus de personnes à le faire. Il arrive que certaines personnes se rendent à une collecte dans le cadre d’un don, mais soient refusées du fait qu’elles ne remplissent pas certaines conditions. Ces conditions devraient pouvoir être non seulement connues mais comprises à l’avance par tous.

De plus, le site de l’EFS[[5]](#footnote-5) n’est pas organisé de façon optimale en ce qui concerne l’accès aux informations relatives au don du sang ; plusieurs rubriques portent par exemple le même nom mais ne sont pas situées au même endroit. Un système QR permettrait de regrouper toutes ces informations et d’y avoir accès à partir d’une seule et même interface, facilitant ainsi l’accès aux informations pour les utilisateurs.

La raison qu’aucun système question-réponse n’ait jusqu’ici été développé dans ce domaine est que le don de sang est un domaine assez restreint et que ce besoin n’a pas encore été identifié.

## Ressources

Nous decrivons ici les ressources externes que nous utiliserons pour la construction du système. Deux types de ressources seront utilises : corpus de questions et corpus de reponses.

#### Corpus de questions

Le corpus de questions comprendra des exemples de questions qui nous serviront a elaborer et evaluer notre methode d’analyse des questions, notamment en etudiant l’ensemble de motifs de questions possibles.

En ce qui concerne la construction du corpus de questions, nous allons en premier lieu exploiter les ressources présentes sur le site internet de l’Etablissement Français du Sang et les regrouper sous forme de catégories. En rassemblant les différentes rubriques, nous obtenons les catégories suivantes :

* **Critères principaux** : cette catégorie concerne l’âge, le poids etc.
* **Etat de santé**: cette catégorie concerne les contre-indications au don, les antécédents médicaux etc.
* **Mode de vie** : cette catégorie concerne la sexualité, la consommation d’alcool et/ou de drogue, les voyages à l’étranger etc.
* **Déroulement d’un don** : cette catégorie concerne la durée d’un don, la quantité de sang prélevée etc.

Les rubriques mentionnees ci-dessus contient des donnees sous formes de descriptions breves / textes courts / de qqs paragraphes / descriptions techniques / explications…. Redigees a destination des utilisateurs.

Nous envisageons d’accumuler l’ensemble de ces informations sous forme d’un document XML qui aurait comme éléments les différentes catégories citées ci-dessus ainsi que les sous-catégories qu’elles renferment. Cela permettrait d’avoir un corpus structuré et de faciliter la recherche des réponses des questions par la suite.

<rubrique id=’’criteres\_principaux’’>

<question type= ‘’age’’>puis-je donner avant mes 18ans ?</ question >

< question type=’’poids’’>puis-je donner si je pèse 45 kg ?</ question >

</rubrique>

Figure XXX : exemple de structure XML pour les ressources textuelles sur le don de sang

Le problème des questions présentes dans les FAQ du site de l’EFS est qu’elles sont présentées en langue ‘écrite’ tandis que nous supposons que les questions posées par les utilisateurs seront présentées en langue ‘orale’ qui est utilisee naturellement dans le situations du quotidien. Ainsi, il est rare que les utilisateurs se servent de la structure avec inversion sujet-verbe lorsqu’ils soumettent une question au système alors que c’est sous cette forme qu’elles se trouvent dans les FAQ. De cet fait, il nous faudra nous baser sur les recherches sur le *text entailement* d’(Abacha and Dina 2016).

En exploitant uniquement les questions présentes dans les FAQ du site de l’EFS, nous obtenons un corpus de 34 questions, ce qui n’est pas suffisant. En revanche, en additionnant les différentes questions présentes sur les différents sites d’information autour du don de sang (voir l’annexe 2), nous obtenons un corpus de 79 questions. La plupart des questions sont redondantes, mais cela est un atout puisque les réponses seront alors validées dans plusieurs documents.

#### Corpus de réponse

Nous appellerons ici « corpus de réponse » la base de données textuelles contenant des informations que nous allons exploiter pour en extraire les réponses de questions dans notre système. Le corpus de réponse sera construit en exploitant les ressources présentes sur le site de l’EFS, mais également les ressources présentes sur d’autres sites internet (des exemples sont cités en annexe). Ce corpus doit être construit avec vigilance et rigueur, surtout en ce qui concerne la fiabilité des sources, car une mauvaise réponse dans ce domaine peut avoir de lourdes conséquences pour l’utilisateur. Ainsi, nous nous restreindrons à des sites internet spécialisés dans le domaine de don de sang afin d’être sûre de la véracité des informations fournies dans ces textes.

## Types de questions

Dans cette partie, nous dressons la liste des types de questions que nous avons observé dans les FAQ de l’EFS.

#### Les questions fermées.

Elles attendent une réponse de type booléen, c’est-à-dire une validation ou au contraire une invalidation. Ces dernières peuvent être de type *nécessairement vrai* (la nécessité) ou bien de type *contingentement vrai* (la possibilité)– c’est ce qu’on appelle la logique aléthique (Alliot et al. 2002). Un énoncé sera *nécessairement vrai* s’il est validé ou invalidé en permanence. Par exemple, la proposition A « on ne peut jamais donner son sang si l’on a moins de 18ans » est *nécessairement vrai*; on écrira alors □A. En revanche, un énoncé sera *contingentement vrai* s’il n’est validé que dans certaines conditions. Par exemple, la proposition B « je dois emmener des documents spécifiques lors du don ? » est *contingentement vrai* puisque cela dépend s’il s’agit d’un premier don ou non ; on écrira alors ◇B. Ainsi, au niveau sémantique, une réponse à une question fermée fera partie de l’ensemble E {p, ¬p, □p, ◇p, ¬□p, ¬◇p} (Langlet and Clavel 2015).

Au niveau morpho-syntaxique, en français, les questions fermées peuvent être représentées de différentes façons.

* **L’inversion sujet-verbe**. Comme dans de nombreuses langues, le mode interrogatif de la langue française demande une inversion sujet-verbe. Cependant, cette tournure tend de plus en plus à être synonyme d’un haut niveau de langue et donc à être mise de côté. *Cela pose-t-il un problème ?*
* **« est-ce que »**. Etymologiquement, cela vient de l’inversion de « c’est que », mais cela n’a aujourd’hui plus la même signification. Cette tournure est largement utilisée dans les structures interrogatives.
* *Est-ce que cela pose un problème ?*
* **La structure conditionnelle**. Une structure conditionnelle peut être utilisée dans le cadre d’une demande d’information. Techniquement, elle n’appelle pas de marque d’interrogation.

*Je me demandais si cela posait un problème.*

* **Une simple marque d’interrogation**. A l’oral, cette structure se manifeste par une intonation montante. A l’écrit, il s’agit simplement de mettre un point d’interrogation à la fin de la question.

*Cela pose un problème ?*

Il faut cependant tenir compte du fait que la réponse à une question fermée ne se limite pas à « oui » ou « non ». Dans la majorité des cas, des informations supplémentaires permettant d’expliquer la polarité de la question seront nécessaires. En effet, le principe du système QR que nous cherchons à construire est de pouvoir apporter une réponse **complète** à l’utilisateur ; il ne doit donc pas se contenter de réponse positive ou négative, mais d’expliquer pourquoi il a répondu de cette manière.

#### Les questions ouvertes

Par opposition aux questions fermées, les questions ouvertes sont celles qui n’attendent pas une réponse booléenne ; elle est au contraire totalement libre. La réponse à une question ouverte se trouvera donc dans les mots-clés et/ou dans les entités nommées présentes dans la question.

Qutilisateur: « Combien faut-il peser pour donner son sang ? »

Dans l’exemple de question ouverte ci-dessus, le système devra reconnaître l’entité nommée *combien* et l’associer au verbe *peser*. Ainsi, les passages à extraire contenant une réponse potentielle devront contenir un poids. Cependant, ce qui s’applique aux questions fermées s’applique aussi aux questions ouvertes. Si l’on reprend l’exemple ci-dessus, le système ne peut pas se contenter de retourner une réponse de type poids ; il doit aussi retourner un passage, c’est-à-dire un texte explicatif qui puisse justifier la réponse retournée.

## Première modélisation

Avant de pouvoir modéliser les questions afin qu’elles soient traitables par l’ordinateur, nous devons les normaliser, c’est-à-dire effectuer une série de traitement afin de rendre les données de la question exploitables.

#### Présentation de NLTK, Stanford CoreNLP et TreeTagger

La bibliothèque NLTK (*Natural Language Toolkit*) est une bibliothèque Python développé en 2005 par Steven Bird et Edward Loper qui permet notamment de segmenter, d’identifier les entités nommées, de construire des arbres de dépendance, d’étiqueter mais intègre aussi une fonction d’apprentissage automatique de classification de textes, ce qui peut être intéressant dans le cadre de notre système.

Stanford CoreNLP est une librairie Java développé par un groupe de professeur, postdoctorants et étudiants de l’université de Stanford. Ce module reconnaît les entités nommées, la coréférence, la dépendance et permet d’étiqueter, de segmenter et d’analyser la polarité sentimentale d’un texte et intègre notamment une fonction appelé *Bootstrapped Entity Learning*, qui permet un apprentissage par motif

TreeTagger est un outil développé en 1995 par Helmut Schmid. Cet outil permet l’étiquetage et la lemmatisation de segments textuels. L’atout de cet analyseur réside dans sa capacité à étiqueter et à lemmatiser de nombreuses langues telles que l’allemand, l’anglais, le français, l’italien, le néerlandais, l’espagnol et beaucoup d’autres.

#### Segmentation.

Toute normalisation d’un segment textuel commence par sa segmentation, c’est-à-dire sa division en phrase et/ou en mot en fonction de l’*input* de l’utilisateur. Dans le cas d’une segmentation en phrase, il n’est pas suffisant de considérer la ponctuation (le point, les points de suspension etc) comme marqueur de clôture de phrase, de même qu’il ne faudra pas considérer la capitalisation d’une lettre comme marqueur de début de phrase.

Q1utilisateur « J’ai des allergies (pollen, graminées, arachides…) et je voulais savoir si c’était un problème dans le cadre d’un don. »

Q2utilisateur « Je suis atteinte de la maladie de Raynaud. Cela pose-t-il un problème ? »

Dans la question Q1, les points de suspension ne marquent pas la fin d’une phrase mais une non-exhaustivité concernant l’énumération qui précède. Dans la question Q2, la majuscule ne correspond pas au commencement d’une nouvelle phrase mais au nom d’une maladie. Il est également à noter que dans le cas de la question Q1, aucune marque d’interrogation (« ? ») n’est visible. Dès lors qu’une demande d’information ou de confirmation d’un fait est formulée, on a affaire à une question.

Lorsque l’on soumet ces deux questions à TreeTagger, la ponctuation et la présence de majuscule ne pose aucun problème. En revanche, les tirets présents dans la question 2 ne sont pas segmentés, ce qui nous donne *-t-il.* Lorsqu’on soumet la structure *est-ce que* au système, la même chose se produit. Ce défaut de segmentation n’a cependant pas une grande importance, d’autant plus que TreeTagger est quand même capable de reconnaître les mots composés comme *porte-manteau*.

#### Lemmatisation.

De même que pour le processus de segmentation, le processus de lemmatisation sera effectué de manière automatique grâce aux modules cités précédemment (NLTK, Stanford CoreNLP, TreeTagger). Nous devons cependant faire face à un problème de taille dans le domaine des systèmes QR : celui de l’orthographe des segments textuels saisis par les utilisateurs. Il serait en effet souhaitable de prendre en compte les erreurs pouvant être commises dans l’*input* d’un utilisateur afin de ne pas omettre un élément important.

Qutilisateur: « je suis atteins**1** d’anémi**2**. C’est grave pour donné**3** son sang ? »

Les éléments 1 et 3 ne poseront pas de problème car un lemmatiseur est capable de traiter des entrées comportant des erreurs de conjugaison. Ainsi, si nous soumettons cette phrase à TreeTagger, nous obtenons la sortie suivante :

je PRO:PER je

suis VER:pres suivre|être

atteins VER:pres atteindre

d’anémi VER:pper <unknown>

. SENT .

C’est NOM <unknown>

grave ADJ grave

pour KON pour

donné VER:pper donner

son DET:POS son

sang NOM sang

? SENT ?

Nous pouvons observer ici que l’élément 2 n’est pas analysé correctement car il présente une erreur d’orthographe qui n’aboutit à aucun autre mot comme c’est le cas des éléments 1 et 3. Le lemmatiseur ne parvient alors pas à rétablir sa forme d’origine. (Jurafsky and Martin 2017) définissent deux types d’erreurs :

* La correction de non-mots : il s’agit de la détection et de la correction des erreurs aboutissant à des non-mots (l’omission d’une lettre par exemple, comme l’élément 2 dans l’exemple ci-dessus).
* La correction de mots : il s’agit de la détection et de la correction des erreurs aboutissant néanmoins à des mots existants (comme les éléments 1 et 3 dans l’exemple ci-dessus, qui sont des erreurs cognitives)

Pour corriger les erreurs, nous recherchons les mots erronés dans un dictionnaire. S’ils ne sont pas présents, il s’agit de non-mots ; nous utiliserons alors la distance de Levenshtein afin de trouver le mot existant le plus proche. Cette distance se base sur le nombre minimal d’opérations élémentaires de modification de caractère (substitution, addition, suppression) nécessaires à l’obtention de l’un des mots à partir de l’autre et permet, à partir d’une séquence de caractères (ou d’un non-mot), de rechercher dans un le mot qui aurait la distance de Levenshtein le plus petite avec le non-mot. Cela permettrait de fournir une meilleure entrée pour le lemmatiseur, qui n’aurait pas pu identifier le non-mot. Le schéma général de ce traitement est présenté sur la figure 2.

Par exemple, les mots se rapprochant le plus de l’élément 2 pourraient être *anémone*, *anémie*, *ainé*, *année*. Nous obtenons alors les distances suivantes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mot erroné | Mot candidat | Correction | Erreur | Modification | Distance |
| anémi | anémone | -one | -i | Substitution et addition | 3 |
| anémi | anémie | -e | \_ | addition | 1 |
| anémi | ainé | -i | -mi | addition et suppression | 3 |
| anémi | année | -n, -e | -mi | substitution, addition, suppression | 3 |

Dans cet exemple, il est clair que le mot signifié par l’utilisateur est alors le mot *anémie*.

Recherche des mots les plus proche dans un dictionnaire

Extraction

Sélection de la plus petite distance

Calcul de la distance de Levenshtein

Mot mal orthographié

Question

Modification du mot

*Figure 2. Schéma de modification d’un mot* *mésorthographié grâce à la distance de Levenshtein*

L’autre solution serait d’utiliser l’inférence bayésienne, une méthode permettant de calculer la probabilité d’un évènement, et plus précisément le *noisy channel* *model* développé par(Shannon 1948) car celui-ci peut être appliqué à la correction orthographique. La figure 3 présente le schéma de se traitement.

Mot signifié par l’utilisateur

Mot entré par l’utilisateur

Source de bruit (faute d’orthographe)

*Figure 3. Schéma de correction orthographique avec application du modèle de Shannon*

Cependant, le contexte dans lequel cette correction orthographique s’applique est selon nous trop restreint pour que l’utilisation de ce modèle soit pertinente.

Une fois les segments textuels lemmatisés, nous pourrons procéder à leur étiquetage morpho-syntaxique. Ce dernier est envisagé via différents modules : les bibliothèques NLTK et Stanford CoreNLP, et le logiciel TreeTagger. L’étiquetage morpho-syntaxique nous permettra d’éliminer les éléments dits ‘vides’ (qui servent uniquement d’outils grammaticaux) des éléments dits ‘pleins’ (chargés d’une fonction sémantique) selon la théorie de (Tesnière 1959). C’est sur ces derniers que nous nous appuieront afin d’extraire les passages contenant la réponse potentielle à une question à partir du corpus textuel.

Ainsi, une fois que les différents processus cités précédemment auront été appliqués à la question de l’utilisateur, nous pourrons passer à l’étape suivante, c’est-à-dire à l’analyse de la question.

#### Analyse des questions

Afin que la machine puisse comprendre ce qui lui est demandé (c’est-à-dire interroger le corpus pour répondre à la question posée par l’utilisateur), il va falloir modéliser la question, c’est-à-dire la faire passer de la langue naturelle à un métalangage compréhensible par le système. En partant de notre hypothèse de filtrage par motif, nous allons tenter en prenant l’exemple spécifique de la question de poids dans le cadre du don du sang, de déterminer les différents motifs possibles.

Q1utilisateur « Je pèse X kilos. Est-ce que je peux donner mon sang ? »

Q2utilisateur « Est-ce que je peux donner mon sang si je pèse X kg ? »

Q3utilisateur « Je peux donner mon sang si je pèse X kg ? »

Q4utilisateur « Si je pèse X kg, je peux donner mon sang ? »

Q5utilisateur « Y a-t-il un poids minimal/maximal pour donner son sang ? »

Q6utilisateur « Est-ce qu’il y a un poids minimal/maximal pour donner son sang ? »

Q7utilisateur « Combien faut-il peser pour donner son sang ? »

Q8utilisateur « Quel est le poids minimal/maximal pour faire le don ? »

Les six premières questions, qui sont des questions fermées, attendent une réponse de type booléen. Les deux dernières, qui sont des questions ouvertes puisqu’elles commencent respectivement par une entité nommée et par un pronom interrogatif, attendent une réponse de type poids.

Ainsi, nous obtenons les motifs suivants pour les quatre premières questions citées ci-dessus :

[conditions\_utilisateur]. ?[conditions\_don]=true/false

?[conditions\_don]si[conditions\_utilisateur]

Si[conditions\_utilisateurs] alors ?[conditions\_don]

Dans le cas de certaines questions, l’utilisateurs d’opérateurs logiques (∧,∨,¬,⊕,→ etc) est fortement envisagée.

En ce qui concerne les questions ouvertes, nous pourrions considérer un modèle d’appariement question-réponse, notamment proposé par (Athenikos, Han, and Brooks 2008). Ces derniers ont construit des patrons question-réponse en créant des triplets sémantiques ayant la forme sujet-prédicat-objet. Cette forme logique pourrait être représentée de la manière suivante :

« Est-ce qu’il y a un poids minimal pour donner son sang ? »→ don\_poids(minimal, ?x).

En revanche, cette technique ne se révèlerait pas particulièrement adaptée dans le cas des questions fermées, qui attendent une réponse booléenne et qui adopteraient plutôt les motifs cités ci-dessus.

Bibliographie – elle doit etre avant les annexes !

# Annexes

Annexes

**Annexe 1. Taxonomie de (Li & Roth, 2004)**

|  |  |  |  |
| --- | --- | --- | --- |
| Classe | Définition | Classe | Définition |
| **ABBREVIATION** | abréviation | **description** | description de qqch |
| **abb** | abréviation | **manière** | manière dont une action est effectuée |
| **exp** | expression abrégée | **raison** | raisons |
| **ENTITE** | entités | **HUMAIN** | êtres humains |
| **animal** | animaux | **groupe** | groupe ou organisation de personnes |
| **corps** | organes du corps humain | **per** | une personne |
| **couleur** | couleurs | **titre** | le titre d'une personne |
| **création** | inventions, livres, œuvres d'art etc. | **description** | description d'une personne |
| **devise** | monnaies | **ENDROIT** | endroits |
| **mal.med.** | maladies et médicaments | **ville** | villes |
| **évènement** | évènement | **pays** | pays |
| **nourriture** | nourriture | **montagne** | montagnes |
| **instrument** | instrument de musique | **autre** | autres endroits |
| **lang** | langues | **région** | régions |
| **lettre** | lettre de l'alphabet | **NUMERIQUE** | valeurs numériques |
| **autre** | autres entités | **code** | code postal |
| **plante** | plantes | **compte** | quantité de qqch |
| **produit** | produits | **date** | dates |
| **religion** | religions | **distance** | mesures linéaires |
| **sport** | sports | **argent** | prix |
| **substance** | éléments et substances | **ordre** | classement |
| **symbole** | signes et symboles | **autre** | autre nombre |
| **technique** | techniques et méthodes | **période** | durée de qqch |
| **terme** | termes équivalents | **pourcentage** | pourcentage ou fraction |
| **véhicule** | véhicules | **vitesse** | vitesse |
| **mot** | mots avec une spécificité | **temp** | température |
| **DESCRIPTION** | description et concepts abstraits | **taille** | taille, aire et volume |
| **définition** | définition de qqch | **poids** | poids |

**Annexe 2. Liste non-exhaustive de sites internet permettant la construction du corpus de réponses.**

<https://www.service-public.fr/particuliers/vosdroits/F2376>

<https://www.april.fr/informations/don-de-sang-articles>

<https://www.toutsurlatransfusion.com/dondusang/gestion-post-don/information-post-don.php>

[www.donnersonsang.com/site/page-80-les-differents-dons-sang-plasma-plaquette.html](http://www.donnersonsang.com/site/page-80-les-differents-dons-sang-plasma-plaquette.html)

<http://www.dondusang-doubs.org/comment-se-passe-le-don-de-sang>

https://transfusion.be/fr/le-don-de-sang/qui-peut-donner-du-sang/

Bibliographie

Abacha, Asma Ben, and Demner-Fushman Dina. 2016. “Recognizing Question Entailment for Medical Question Answering.” *AMIA ... Annual Symposium Proceedings. AMIA Symposium* 2016 (May): 310–18.

Alliot, J, T Schiex, P Brisset, and F Garcia. 2002. “Intelligence Artificielle et Informatique Théorique (2 Ed.).” *Recherche*.

Antonio, G. A., A. Cisternino, F. Formica, M. Simi, and R. Tommasi. 2001. “PiQASso: Pisa Question Answering System.” *Proceedings of the Text REtrieval Conference*, no. January: 599–607.

Athenikos, Sofia J., Hyoil Han, and Ari D. Brooks. 2008. “Semantic Analysis and Classification of Medical Questions for a Logic-Based Medical Question-Answering System.” *Proceedings - 2008 IEEE International Conference on Bioinformatics and Biomedicine Workshops, BIBMW*, no. December: 111–12.

Clark, Alexander, Chris Fox, and Shalom Lappin. 2010. *The Handbook of Computational Linguistics and Natural Language Processing*. Edited by Wiley-Blackwell.

Cruchet, Sarah, Arnaud Gaudinat, and Célia Boyer. 2008. “Supervised Approach to Recognize Question Type in a QA System for Health.” *Studies in Health Technology and Informatics* 136 (February): 407–12.

Green, Bert, Alice Wolf, Carol Chomsky, and Kenneth Laughery. 1961. “Baseball: An Automatic Question Answerer.” *Proceedings of Western Computing Conference*, 219–24.

Jurafsky, Daniel, and James H Martin. 2017. “Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.”

Langlet, Caroline, and Chloé Clavel. 2015. “Modélisation Des Questions de l’agent Pour l’analyse Des Affects , Jugements et Appréciations de l’utilisateur Dans Les Interactions Humain-Agent,” no. July.

Laurent, Dominique, and Patrick Séguéla. 2004. “QRISTAL, Système de Questions-Réponses.” *TALN 2005*.

Li, Xin, and Dan Roth. 2006. “Learning Question Classifiers: The Role of Semantic Information.” *Natural Language Engineering* 12 (3): 229–49.

Ligozat, Anne-laure. 2004. “Système de Question Réponse : Apport de l’analyse Syntaxique Lors de l’extraction de La Réponse.” *RECITAL 2004*.

Mendes, Sara, and Véronique Moriceau. 2004. “L ’ Analyse Des Questions : Intérêts Pour La Génération Des Réponses.” *TALN Workshop Question-Réponse, Fès, 22 Avril 2004*.

Moriceau, Véronique, and Xavier Tannier. 2009. “Apport de La Syntaxe Dans Un Système de Question-Réponse : Étude Du Système FIDJI.” *Architecture*, no. January 2009: 24–26.

Niu, Yun, Graeme Hirst, Gregory McArthur, and Patricia Rodriguez-Gianolli. 2003. “Answering Clinical Questions with Role Identification.” *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine - Volume 13*, no. July: 73–80.

Omrane, Nouha, Adeline Nazarenko, and Sylvie Szulman. 2009. “Les Entités Nommées : Éléments Pour La Conceptualisation,” no. May: 1–6.

Rinaldi, Fabio, James Dowdall, Gerold Schneider, and Andreas Persidis. 2004. “Answering Questions in the Genomics Domain.” *ACL 2004 Workshop on Question Answering in Restricted Domains*, no. July: 46–53.

Shannon, Claude E. 1948. “A Mathematical Theory of Communication.” *The Bell System Technical Journal* 27 (July 1928): 379–423.

Tesnière, Lucien. 1959. *Eléments de Syntaxe Structurale*.

1. https://www.hon.ch/ [↑](#footnote-ref-1)
2. https://www.nltk.org/ [↑](#footnote-ref-2)
3. https://stanfordnlp.github.io/CoreNLP/ [↑](#footnote-ref-3)
4. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ [↑](#footnote-ref-4)
5. https://dondesang.efs.sante.fr/ [↑](#footnote-ref-5)