Classification de questions en langage naturel par le type sémantique des réponses attendues

Théo Oriol*, Mathieu Dodard*, Kévin Cousot*, Melissa Mekaoui*, Hani Guenoune*,**, Jean Bort*, Antoine Nguyen*, Thibaud Sanchez*, Philippe Garnier*, Cédric Lopez*

*Emvista, Cap Oméga, Rond-point Benjamin Franklin, Montpellier
prenom.nom@emvista.com,
https://www.emvista.com/
**LIRMM, 161 rue Ada, Montpellier
hani.guenoune@lirmm.fr
http://www.lirmm.fr/

Résumé. Les systèmes de question-réponse (QA, *Question Answering*) sont traditionnellement constitués des trois tâches suivantes : 1) analyse de la question, 2) analyse de l'ensemble documentaire contenant les réponses, 3) recherche et extraction des réponses. Dans cette dernière décennie, les systèmes de QA à base d'apprentissage prennent la forme d'un modèle *end-to-end*. Par conséquent, les trois étapes ne sont plus explicitement représentées. Il en résulte que les systèmes de QA à base d'apprentissage les plus récents commettent de nombreuses erreurs dès lors que la réponse n'est pas dans le texte ou qu'un raisonnement est nécessaire. En particulier, le type sémantique de la réponse attendue (TSA) peut être incohérent avec le type sémantique de la réponse retournée. Dans cet article, nous nous focalisons sur la tâche d'identification du TSA. Dans un premier temps, nous proposons une taxonomie pour représenter les TSA. Dans un second temps, nous expérimentons des modèles avec CamemBERT développés à partir du corpus de questions-réponses français FQUAD. L'évaluation est réalisée sur le corpus de questions-réponses français PIAF.

1 Introduction

Les systèmes de question-réponse (QA) ont pour objectif de retourner automatiquement les réponses aux questions posées par les humains. Les systèmes de QA sont indispensables pour faciliter l'accès à une information précise noyée dans un grand volume documentaire. Les systèmes actuels sont fondés sur l'hypothèse que la réponse se trouve dans le texte. Il en découle que, dans le cas contraire, les modèles les plus avancés demeurent confrontés à un problème d'incohérence sémantique de la réponse retournée vis-à-vis de la question posée. Par exemple, la question "Quel animal mange des céréales ?" posée sur le texte "La société Gradle fournit des céréales." retourne "Gradle" avec le modèle le plus récent traitant le français, CamembertQA [1] (Martin et al., 2020). L'incohérence sémantique entre la réponse attendue de

1. https ://fquad-demo.illuin.tech/

type "animal" et la réponse retournée de type "organisation" témoigne d'une des limites des systèmes actuels.

Depuis la création du track QA à Text Retrieval Conference (TREC) (Voorhees et Tice, 1999), trois modules composent traditionnellement un système de QA : 1) un module d'analyse de la question 2) un module d'analyse de l'ensemble documentaire contenant les réponses 3) un module de recherche et d'extraction des réponses. Depuis la fin des années 90, de nombreux systèmes ont été expérimentés, à base de règles, à base d'apprentissage ou hybrides. Les systèmes de question-réponse ont connu cette dernière décennie de grandes avancées grâce notamment aux nouvelles techniques d'apprentissage. Les systèmes les plus récents sont des modèles dits "end-to-end", *i.e.* un système d'apprentissage complexe représenté par un seul modèle. Il en résulte que l'étape de classification de la question n'est plus explicite.

Cette étape largement étudiée jusqu'en 2010 a généralement été traitée de façon surfacique (utilisation de patrons morphosyntaxiques, peu de types sémantiques traités, ...) alors que la tâche peut s'avérer complexe. Par exemple, deux questions telles que "Quel est le prix du concours ?" et "Quel est le prix de l'inscription ?" sont lexicalement et syntaxiquement proches mais le type sémantique de la réponse attendue est différent (une récompense *vs.* une mesure).

Dans cet article, nous revenons sur l'existence d'un module indépendant de classification des questions par le type sémantique des réponses attendues, noté TSA dans la suite. En s'appuyant sur les taxonomies existantes, nous proposons une nouvelle taxonomie pour annoter deux récents jeux de données constitués de questions en français, FQUAD (Martin et al., 2020) et PIAF [2] (*cf.* section 3.1). Nous définissons le protocole d'annotation et nous décrivons les jeux de données annotés en section 3.2. Ensuite, nous décrivons les systèmes développés sur la base des jeux de données annotés (*cf.* section 3.3) avant de les évaluer (*cf.* section 4).

2 Travaux antérieurs

La classification des questions selon leur TSA a été largement étudiée au début des années 2000 (Loni, 2011). La majorité des travaux définit la tâche comme une tâche de classification mono-label et reposent donc sur l'hypothèse qu'une question ne peut pas être ambiguë. Or, une question telle que "Comment se nomme le glacier ?" suffit à montrer la non validité de cette hypothèse : au moins deux TSA sont possibles (le nom de l'humain qui vend des glaces *vs.* la masse de glace). Ce problème d'ambiguïté peut être levé par l'utilisation d'une taxonomie à plusieurs niveaux de classes telle que celle que nous proposons dans cet article. En effet, dans l'exemple donné, quel que soit le type sémantique attendu, il est une chose concrète (classe *Concrete*, *cf.* section 1). Quelques rares travaux considèrent la tâche comme une tâche de classification multi-label (Li et Roth, 2002).

Les systèmes symboliques utilisent généralement des lexiques et des patrons morphosyntaxiques. Des heuristiques simples sont la plupart du temps mises en place (par exemple, si la question commence par "Qui" alors le TSA est *Human*) (Cabrio et al., 2012) (Ben Abacha, 2012). De telles heuristiques s'avèrent inefficaces lorsqu'il s'agit de traiter une question telle que "Qui est Robert Rubin" qui attend comme réponse "Secrétaire d'Etat au Trésor américain", *i.e.* une fonction. Monceaux et Robba (2002) analyse la question avec un parser syntaxique de façon à repérer le gouverneur du premier groupe nominal. Ainsi pour la question "Quel métal

2. https://piaf.etalab.studio/dataset/

a le point de fusion le plus élevé ?" le terme "métal" est retourné (mais pas la classe sémantique correspondante). Néanmoins, l'utilisation de la syntaxe peut se révéler inefficace ; par exemple, les questions "Quel est le prix du concours ?" et "Quel est le prix de l'inscription ?" sont lexicalement et syntaxiquement proches mais le TSA est différent (une récompense *vs.* une mesure). Les ressources lexicales doivent donc être très couvrantes au risque que le système ne soit pas en mesure d'identifier certains TSA.

Les systèmes à base d'apprentissage s'emparent également de la problématique, en particulier suite à la publication du jeu de données UIUC (Li et Roth, 2002) et de son utilisation récurrente à la conférence Text REtrieval Conference. Ce jeu de données propose $6\,000$ questions annotées avec le TSA ($5\,500$ pour l'entraînement et 500 pour l'évaluation). Les types sont organisés dans une taxonomie en deux couches, la première assez générale (humain, lieu), la seconde plus spécifique (groupe/individu, ville/pays). Les solutions proposées sont principalement des modèles d'apprentissage supervisé. Celles-ci visent à attribuer à une question la classe la plus vraisemblable en exploitant des traits relevant du lexique, de la syntaxe ou bien encore de la sémantique. Des systèmes hybrides sont également proposés (par exemple Silva et al. (2011)). D'abord un système de patrons est appliqué à la question et, si la reconnaissance fonctionne la question est directement classifiée. En cas d'échec, l'objet de la question, ses hyperonymes et d'autres traits sont extraits et présentés à un SVM. En guise d'exemple, l'approche développée par Silva et al. (2011) atteint $90,8\%$ d'accuracy sur les classes générales et $95,0\%$ sur les spécifiques. D'autres travaux se distinguent en abordant la tâche comme une classification multi-label hiérarchique (Li et Roth, 2002). La classification s'effectue en deux temps. Un premier classifieur calcule la probabilité conditionnelle d'appartenance de la question aux classes de la première couche puis, fort de ce résultat, un second classifieur fait de même pour la couche fine. Enfin, plus récemment, des architectures de type transformers atteignent un taux d'erreur de $1,93\%$ (Cer et al., 2018) sur la taxonomie de TREC-6 (six classes).

3 Description des travaux

Dans un premier temps, nous décrivons la taxonomie utilisée (*cf.* section 3.1) pour annoter les jeux de données (*cf.* section 3.2). Ensuite, nous décrivons les modèles expérimentés et discutons des résultats (*cf.* section 4).

3.1 Taxonomie

L'utilisation d'une taxonomie des TSA, *i.e.* une classification hiérarchisée, est pertinente dès lors que l'on souhaite inférer de nouvelles classes à partir d'une classe retournée par un système. Par exemple, dans le cas où un système identifierait qu'une question attend une réponse de type "Equipe de sport", un raisonneur déduirait que cette réponse est également du type "Organisation" [3].

La représentation de la sémantique des réponses attendues ne fait pas l'unanimité. La tâche de classification des questions selon leur TSA nécessite une taxonomie qui soit en mesure de représenter la sémantique de tout type de mention dans un texte. Il s'agit donc d'être en

3. Par exemple, l'application d'un raisonnement avec un algorithme tel que Fact++ serait tout à fait approprié.

mesure de représenter les termes du vocabulaire ainsi que les entités nommées. Concernant les termes du vocabulaire, la communauté travaillant sur la tâche de *Word Sense Disambiguisation* (WSD) a largement adopté la taxonomie proposée dans le cadre de VerbNet (Schuler, 2005). Cette taxonomie "WSD" a été reprise ici avec quelques modifications apportées empiriquement suite à la confrontation avec des données réelles (par exemple, dans notre taxonomie, "time" est une sous-classe de "abstract", pas dans la taxonomie de VerbNet). Une telle taxonomie ne couvre pas les entités nommées ; par exemple, un nom de produit n'est pas explicitement représenté. La tâche de reconnaissance d'entités nommées a de son côté proposé de nombreuses taxonomies contenant de quelques dizaines (Rizzo et Troncy, 2012) à quelques centaines de types (Sekine et al., 2002). Nous nous sommes appropriés l'ontologie NERD de Rizzo et Troncy (2012) en plaçant chacun de ses types sous l'un des types de la taxonomie "WSD". Par exemple, la classe NERD "Product" a été séparée en deux classes : une classe *Abstract>Product* pour représenter les produits abstraits (*e.g.* une chanson) et une classe *Concrete>Product* pour représenter les produits concrets (*e.g.* une voiture).

Par ailleurs, les typologies traditionnellement utilisées pour le développement de systèmes de questions-réponses représentent si la réponse attendue est une définition, un terme, une réponse binaire, ou encore une alternative (par exemple une traduction) (Laurent et Séguéla, 2005) (Ben Abacha, 2012). Les classes correspondantes ont été ajoutées dans la classe *TextualElement*.

Ainsi, la taxonomie que nous proposons peut être vue comme une jointure d'une taxonomie de WSD pour la partie "haute" (*i.e.* proche de la racine), d'une taxonomie d'entités nommées pour la partie "basse" (*i.e.* proche des feuilles) et d'une partie spécifique à la tâche de classification de questions par le TSA. Un aperçu de la taxonomie utilisée dans le cadre de cet article est donné en Figure 1. Celle-ci est constituée de 52 classes [4] : deux classes de niveau 1 (*Abstract* et *Concrete*), vingt-et-une de niveau 2, seize de niveau 3, dix de niveau 4, trois de niveau 5. Cette taxonomie se distingue notamment des autres taxonomies utilisées dans le cadre de la classification de questions qui sont généralement limitées à deux niveaux y compris dans TREC (Loni, 2011).

3.2 Annotation des jeux de données

Deux jeux de données ont été utilisés dans le cadre de nos expérimentations : FQUAD (Martin et al., 2020) et PIAF [5]. Chaque jeu de données est issu d'une initiative différente mais s'est inspiré de son équivalent anglais SQUAD (Rajpurkar et al., 2016). Ainsi, les deux jeux de données sont formatés de la même manière ce qui facilite leur interopérabilité.

Dans le cadre de notre expérience, une équipe de dix annotateurs a été formée pour annoter la totalité des corpus : 20 729 questions pour FQUAD et 7 569 pour PIAF. Les annotateurs ont tous un niveau Bac+5 à Bac+8 spécialisés dans le domaine du traitement automatique des langues.

Une première étape a consisté à demander à chaque membre de l'équipe d'annoter manuellement 50 questions sans se concerter (chaque membre a reçu les mêmes questions que les autres). Les 500 annotations récoltées ont été utilisées pour calculer l'accord inter-annotateurs

4. La taxonomie est consultable ici : `https://www.emvista.com/publications/qa-taxonomy.owl`

5. `https://piaf.etalab.studio/dataset/`

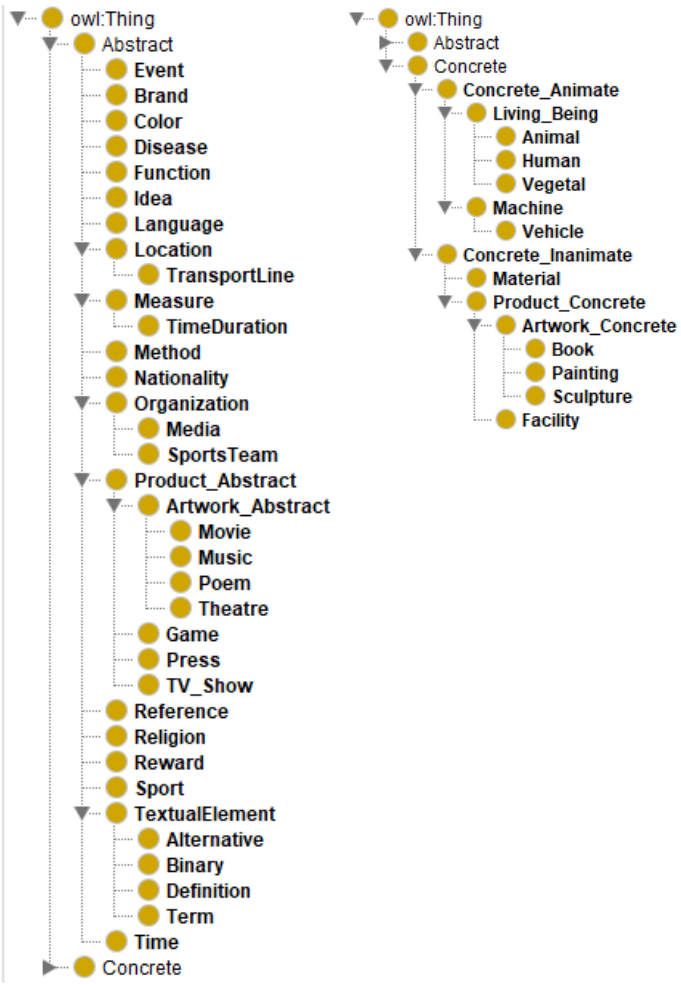

FIG. 1 – *Aperçu de la taxonomie des types sémantiques des réponses attendues (TSA)*

en utilisant les deux métriques suivantes : 1) le Kappa de Cohen, qui mesure le degré de concordance entre deux annotateurs, 2) le Kappa de Fleiss qui mesure le degré de concordance pour un nombre d'annotateurs qui peut être supérieur à deux. Le Kappa de Cohen calculé pour chaque couple d'annotateurs est situé entre 0,56 et 0,80 ce qui est généralement interprété comme un accord fort. Le Kappa de Fleiss calculé pour l'ensemble des annotateurs est de 0,62 ce qui désigne une concordance importante. À ce stade, la tâche d'annotation est donc considérée comme correctement définie et les annotations sont considérées comme étant de bonne qualité.

Lors de la seconde étape, les annotateurs ont contribué sur la totalité des corpus FQUAD et PIAF. Chaque question est annotée avec un unique TSA : dans le cas où deux TSA sont envisagés, la super-classe commune aux deux TSA est retenue.

Une particularité de notre travail repose sur le fait que les questions ont été annotées selon leur TSA sans avoir pris connaissance des réponses présentes dans les jeux de données. Ainsi, la question "A l'aide de quoi Höss est réveillé si il s'endort ?" est annoté "Thing" car, sans connaître "Höss", il est connu qu'un réveil peut être provoqué par une chose abstraite "Abstract" (un cauchemard) ou concrète "Concrete" (un réveil).

Les tableaux 1 et 2 indiquent le nombre d'occurrences de chaque TSA dans les corpus FQUAD et PIAF. Il s'avère que certaines classes ont très peu d'exemples (par exemple 9 TSA "Poem" sont présents dans FQUAD). Nous avons enrichi manuellement les classes jusqu'à atteindre un minimum de 100 questions par classe, soit une extension (notée EXT dans la suite) de 1560 questions qui s'avèrera utile pour enrichir le jeu d'apprentissage dans le cadre de nos expérimentations.

Au total, 29 858 questions ont été manuellement annotées avec leur TSA.

TSA	**Occurrences**	**TSA**	**Occurrences**	**TSA**	**Occurrences**
Abstract	3659	Human	3957	Product	142
Alternative	198	Idea	62 (100)	Reference	53 (100)
Animal	204	Language	36 (100)	Religion	24 (100)
Artwork	140	Location	1735	Reward	50 (100)
Binary	50 (100)	Machine	13 (100)	Sport	14 (100)
Book	122	Material	55 (100)	SportsTeam	142
Brand	37 (100)	Measure	2038	Term	65 (100)
Color	72 (100)	Media	12 (100)	Theatre	22 (100)
Concrete	92 (100)	Method	60 (100)	Thing	3134
Definition	199	Movie	69 (100)	Time	2428
Disease	20 (100)	Music	66 (100)	TimeDuration	177
Event	397	Nationality	104	TransportLine	40 (100)
Facility	167	Organization	448	TV_Show	3 (100)
Function	212	Poem	9 (100)	Vegetal	19 (100)
Game	17 (100)	Press	51 (100)	Vehicle	112

TAB. 1 – *Nombre d'occurrences de chaque type sémantique attendu (TSA) dans le jeu de données FQUAD ; entre parenthèses, le nombre d'occurrences incluant EXT.*

TSA	Occurrences	TSA	Occurrences	TSA	Occurrences
Abstract	671	Human	1326	Product	78
Alternative	71	Idea	10	Reference	4
Animal	27	Language	45	Religion	15
Artwork	35	Location	712	Reward	10
Binary	248	Machine	4	Sport	13
Book	33	Material	15	SportsTeam	46
Brand	7	Measure	654	Term	66
Color	19	Media	12	Theatre	2
Concrete	77	Method	27	Thing	1635
Definition	149	Movie	3	Time	839
Disease	8	Music	23	TimeDuration	69
Event	164	Nationality	51	TransportLine	5
Facility	38	Organization	201	TV_Show	5
Function	104	Poem	2	Vegetal	7
Game	3	Press	12	Vehicle	23

TAB. 2 – *Nombre d'occurrences de chaque type sémantique attendu (TSA) dans le jeu de données PIAF*

3.3 Description des modèles

L'objectif des modèles développés est de classifier chaque question selon le type sémantique de la réponse attendue. Les modèles doivent faire face au fait qu'une question peut être formulée avec une forte variation lexicale et syntaxique (Loni, 2011). Par exemple, les types attendus pour les questions "Quel est le prix du concours de Valve et The National ?" et "Quel est le prix du ticket rechargeable ?" sont respectivement *Reward* et *Measure*. Par ailleurs, les modèles doivent être en mesure de s'abstraire des formulations non standards "**Combientième** Johnny Herbert a-t-il finit ?", des problèmes de typographie, "Quel **joueurmarque** le plus de points sur la saison ?", ou encore des questions difficilement compréhensibles telles que "Pour les soins reçus par les chevaux sont-ils controversés ?".

Dans le cadre de nos expériences, quatre modèles ont été entraînés à partir du modèle de langue CamemBERT (Martin et al., 2020), une version française de BERT (Devlin et al., 2018) :

— "CamemBERT A/C", entraîné uniquement sur les classes Abstract et Concrete qui constituent le premier niveau de la taxonomie. Dans les jeux de données, ces classes contiennent également les exemples des sous-classes respectives.
— "CamemBERT A/I", entraîné uniquement sur les classes Animate et Inanimate, deuxième niveau de la taxonomie. Ces classes contiennent également les exemples des sous-classes respectives.
— "CamemBERT 7C" a été entraîné sur sept classes de niveau intermédiaire : *Artwork*, *Human*, *Location*, *Measure*, *Organization*, *Time* et *TextualElement*.
— "CamemBERT 52C" est entraîné sur les 52 classes de la taxonomie, sans prise en compte de la relation de subsomption.

3.4 Inférences

Le moteur d'inférences utilise la taxonomie décrite en section 3.1 et le TSA retourné par le système. Par exemple, pour la question "A quel autre roman lie-t-il cet ouvrage ?", le système retourne le TSA *Book* ; le moteur d'inférences infère les classes *Artwork_Concrete*, *Product_Concrete*, *Concrete_Inanimate*, *Concrete*, *Thing*. De telles inférences peuvent être pertinentes lorsqu'il s'agit de trouver une réponse de type "produit" *Product* alors que le texte mentionne la réponse sous la forme d'un titre de livre *Book*.

4 Evaluation

L'objectif de l'expérimentation est d'évaluer les performances des modèles en fonction du nombre de classes. Les modèles ont appris sur le jeu de données FQUAD et EXT puis évalués sur le jeu de données PIAF. La précision a été calculée pour chaque modèle. Les modèles ayant donné lieu aux résultats présentés dans la suite sont entrainés sur cinq épochs avec un learning rate de 2e-5 et pour optimizer adamW. Ces modèles sont entrainés avec une taille de batch de 32. Pour chaque classe de la taxonomie, l'apprentissage a été réalisé sur les exemples de ladite classe et de ses sous-classes.

Le modèle "CamemBERT A/C" obtient une précision de 92.2% et 89.4% respectivement sur les classes Abstract et Concrete. La prise en compte de EXT dans l'apprentissage permet d'obtenir 94.3% et 88.0% de précision. Parmi les erreurs commises, certaines sont acceptables ; par exemple le modèle prédit Abstract pour "Que trouve-t-on en face de l'abbaye Saint Victor", au lieu de Concrete tel qu'annoté par l'humain. En revanche, le modèle prédit Abstract pour la question "Quel artiste inspira particulièrement Doris Day ?" alors que la réponse ne peut être que Human, donc Concrete. Au global, la macro precision du modèle est de 87.6% avec un apprentissage sur FQUAD et de 91.1% avec FQUAD+EXT.

Le modèle "CamemBERT A/I" permet de comparer deux classes d'un niveau de granularité inférieur au précédent : Concrete_Animate vs. Concrete_Inanimate. Le modèle obtient une précision de 99.1% et 58.1%. Il existe de nombreuses questions telles que "Quels livres Pascal a-t-il écrit ?" qui reçoivent par erreur une prédiction de type Animate. Au contraire, quasiment toutes les prédictions de type Animate sont correctes. Au global, la macro precision est de 78.6% avec un apprentissage sur FQUAD et de 82.1% avec FQUAD+EXT.

Le modèle "CamemBERT 7C" obtient de très bons résultats pour les classes Human (95.3%), Time (94.4%), Measure (93.4%) ou encore Location (86.1%). Au contraire, la classe Artwork obtient uniquement 5.7% probablement à cause d'un nombre d'exemples trop faible (140 exemples) ; Organization obtient une précision de 49.8%. La principale confusion (*cf.* Fig. 3) existe entre les classes Organization et Human, par exemple pour les questions suivantes : "Entre qui éclatent maintes guerres ?" ou "Contre qui l'Inter a perdu ?" ou encore "Quelle association joue un rôle important dans l'exploitation des fossiles ?". Au global, la macro precision est de 69.5% avec un apprentissage sur FQUAD et de 71.8% avec FQUAD+EXT.

Le modèle "CamemBERT 52C" obtient une macro précision de 39.7% avec un apprentissage sur FQUAD et de 52.5% avec FQUAD+EXT. La matrice de confusion du modèle "CamemBERT 52C" est présentée en Fig. 4. La confusion la plus forte existe entre les classes Disease et Theatre (50%) ce qui est difficilement explicable si ce n'est par le faible nombre d'exemples dans Disease, alors qu'une confusion importante existe également entre Trans-

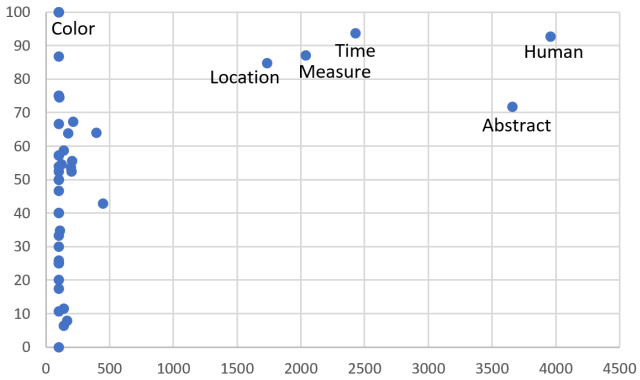

FIG. 2 – *Corrélation entre le nombre d'exemples dans les classes d'apprentissage (abscisses) et la précision obtenue (ordonnées).*

portLine et Location, ce qui est recevable, TransportLine étant une sous-classe de Location. Les scores atteints par ce modèle témoignent de la difficulté de cette tâche de classification lorsqu'un nombre élevé de classes existe.

La figure 2 met en évidence les corrélations qui existent entre le nombre d'exemples dans les classes d'apprentissage (FQUAD + EXT) et la précision obtenue par le modèle "CamemBERT 52C". Il est par exemple intéressant de mettre en évidence que la classe Human obtient un résultat plus faible que la classe Time alors que Human contient 40% d'exemples en plus par rapport à Time. La classe Abstract est aussi largement au-dessous (21%) des résultats de Human avec un nombre d'exemples pourtant proche. Ces résultats illustrent probablement la difficulté pour un tel modèle à prédire certaines classes à cause de la quantité importante de façons d'exprimer des questions pour un TSA donné. Au contraire, certaines classes comme Color nécessitent un nombre moins important de questions pour obtenir une précision parfaite (100%) ; en effet, les questions attendant une réponse de type Color s'expriment presque toujours avec le terme "couleur" et avec des syntaxes proches.

5 Conclusion

Dans cet article, nous avons décrit une expérience concernant le développement de modèles utilisant CamemBERT pour la tâche de classification de questions selon le type sémantique de la réponse attendue. Après avoir annoté les deux récents corpus français FQUAD et PIAF avec une taxonomie permettant de prendre en compte les aspects sémantiques de haut niveau (*i.e.* choses abstraites, choses concrètes) et de bas niveau (*i.e.* métiers, maladies), plusieurs modèles ont été développés. Les résultats indiquent que des classes telles que Abstract ou Human sont plus difficiles à prédire que d'autres, ce qui est probablement lié à une variété lexicale et syntaxique plus prononcée pour certaines classes que pour d'autres. Ces observations donnent une indication pour la poursuite des travaux, notamment sur l'enrichissement du jeu d'apprentissage. En effet, l'extension à FQUAD créée de toute pièce par les annotateurs avec seulement 100 questions maximum par classe a permis d'améliorer les résultats entre 2% et 13% selon

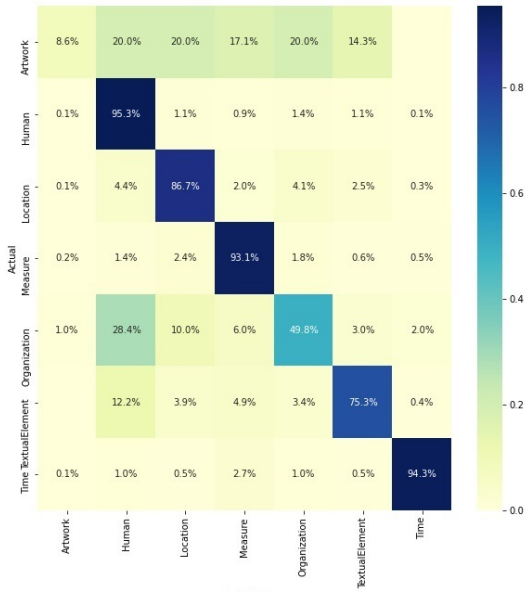

FIG. 3 – *Matrice de confusion de CamemBERT 7C.*

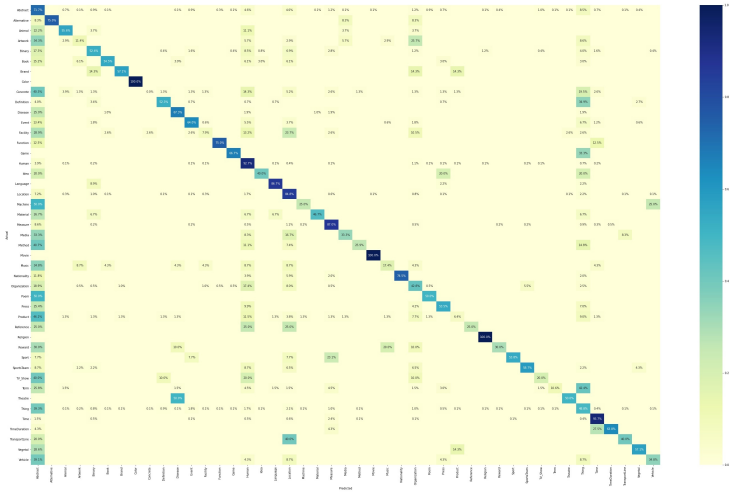

FIG. 4 – *Matrice de confusion de CamemBERT 52C.*

TSA	Précision	TSA	Précision	TSA	Précision
Abstract	68.5 (71.7)	Human	93.4 (92.7)	Press	50.0 (75.0)
Alternative	47.9 (53.5)	Idea	0 (20)	Product	5.1 (6.4)
Animal	48.1 (55.6)	Language	71.1 (86.7)	Reference	0 (25.0)
Artwork	5.7 (11.4)	Location	85.0 (84.8)	Reward	10.0 (30.0)
Binary	1.6 (52.4)	Machine	0 (25)	Sport	0 (53.8)
Book	54.5 (54.5)	Material	40.0 (46.7)	SportsTeam	60.9 (58.7)
Brand	57.1 (57.1)	Measure	85.6 (87.0)	Term	1.5 (10.6)
Concrete	0 (0)	Media	0 (33.3)	Theatre	50.0 (50.0)
Definition	44.3 (52.3)	Method	11.1 (25.9)	Thing	49.1 (48.8)
Disease	62.5 (75.0)	Movie	100 (100.0)	Time	94.2 (93.7)
Event	65.2 (64.0)	Music	17.4 (17.4)	TimeDuration	62.3 (63.8)
Facility	13.2 (7.9)	Nationality	68.6 (74.5)	TransportLine	40.0 (40.0)
Function	68.3 (67.3)	Organization	43.8 (42.8)	TV_Show	0 (40.0)
Game	0 (66.7)	Poem	0 (50.0)	Vegetal	0 (57.1)
				Vehicle	30.4 (34.8)

TAB. 3 – *Précision de chaque type sémantique attendu (TSA) avec le modèle CamemBERT 52C. Les valeurs entre parenthèses sont les résultats obtenus par le modèle ayant appris sur FQUAD et EXT.*

le modèle. Des travaux doivent également être menés pour étendre la taxonomie, de sorte à représenter la cardinalité des réponses attendues, les rôles sémantiques attendus (Agent, Patient, Source, Destination, ...) ou encore l'exclusion de réponses candidates déjà inclues dans la question (par exemple "Hormis le pavillon de Cruzcampo, quel autre pavillon echappe à la démolition ?").

Références

Ben Abacha, A. (2012). *Recherche de réponses précises à des questions médicales : le système de questions-réponses MEANS*. Theses, Université Paris Sud - Paris XI.

Cabrio, E., J. Cojan, A. Palmero Aprosio, B. Magnini, A. Lavelli, et F. Gandon (2012). QAKiS : an Open Domain QA System based on Relational Patterns. International Semantic Web Conference, ISWC 2012. Poster.

Cer, D., Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, et R. Kurzweil (2018). Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, pp. 169–174. Association for Computational Linguistics.

Devlin, J., M. Chang, K. Lee, et K. Toutanova (2018). BERT : pre-training of deep bidirectional transformers for language understanding. *CoRR*.

Laurent, D. et P. Séguéla (2005). Qristal, système de questions-réponses. In *Actes de la 12ème conférence sur le Traitement Automatique des Langues Naturelles*, pp. 53–62. Association pour le Traitement Automatique des Langues.

Li, X. et D. Roth (2002). Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pp. 1–7. Association for Computational Linguistics.

Loni, B. (2011). A survey of state-of-the-art methods on question classification.

Martin, d., V. Maxime, B. Wacim, et B. Tom (2020). FQuAD : French Question Answering Dataset. *arXiv e-prints*, arXiv :2002.06071.

Martin, L., B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, et B. Sagot (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7203–7219. Association for Computational Linguistics.

Monceaux, L. et I. Robba (2002). Les analyseurs syntaxiques : atouts pour une analyse des questions dans un système de question-réponse. *Actes de TALN 2003*, 195–204.

Rajpurkar, P., J. Zhang, K. Lopyrev, et P. Liang (2016). Squad : 100, 000+ questions for machine comprehension of text. *CoRR*.

Rizzo, G. et R. Troncy (2012). Nerd : a framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 73–76.

Schuler, K. K. (2005). *Verbnet : A Broad-Coverage, Comprehensive Verb Lexicon*. Ph. D. thesis.

Sekine, S., K. Sudo, et C. Nobata (2002). Extended named entity hierarchy. In *LREC*.

Silva, J., L. Coheur, A. C. Mendes, et A. Wichert (2011). From symbolic to sub-symbolic information in question classification. *Artificial Intelligence Review* (2).

Voorhees, E. M. et D. M. Tice (1999). The TREC-8 question answering track report. In *Proceedings of the 8th Text Retrieval Conference*, pp. 77–82.

Summary

Question answering systems (QA) are traditionally made up of the following three tasks: 1) Analysis of the question, 2) Analysis of the textual set containing the answers, and 3) Search and extraction of the answers. In the last decade, learning-based QA systems have taken the form of an end-to-end model. Therefore, the three stages are no longer explicitly represented. As a result, the most recent QA systems make many mistakes when the answer is not in the text or when reasoning is required. In particular, the semantic type of the expected answer (TSA) may be inconsistent with the semantic type of the returned answer. In this article, we focus on the task of identifying TSA. First, we propose a taxonomy to represent TSAs. Secondly, we experiment models developed with CamemBERT from FQUAD, a French dataset consisting of questions and related answers. The evaluation is carried out on PIAF, another French dataset consisting of questions and related answers.