Exploration d'approches pour un système de questions-réponses

Laurence Dupont
Nu Echo
Juin 2022

Une fonctionnalité fondamentale des agents conversationnels est de pouvoir répondre aux questions fréquentes des utilisateurs formulées en langue naturelle. En effet, un système de questions-réponses performant permet d'améliorer l'expérience utilisateur et de diminuer le nombre d'appels faits au centre de contacts.

Dans un article de blog précédent, nous avions décrit les expériences réalisées pour implémenter un tel système avec Dialogflow ES [1]. Depuis, Nu Echo a exploré de nouvelles approches pour implémenter un système de questions-réponses qui pourrait éventuellement être intégré à un agent conversationnel en production. Le présent article vise à présenter de façon sommaire les étapes ayant mené au choix de la solution finale, soit un système utilisant le modèle Universal Sentence Encoder (USE) de Google combiné à un réseau de neurones.

1 Définition du problème

Il existe une multitude de façons d'aborder le problème de développement d'un système de questions-réponses basé sur l'apprentissage automatique. L'une d'entre elles est la tâche de sélection de réponse. Elle consiste, pour une question donnée, à retourner la meilleure réponse parmi un ensemble de réponses candidates qui comprend une ou plusieurs réponses correctes [2]. Si la réponse retournée fait partie des réponses correctes à la question, alors la prédiction est correcte, sinon elle est incorrecte [2].

Sachant qu'une réponse est associée à une paire question-réponse, il semble raisonnable d'étendre cette définition de tâche et considérer qu'un candidat peut être une paire question-réponse, sa question ou sa réponse. Lorsqu'un candidat sous l'une de ces formes est sélectionné, il est trivial de retourner la réponse correspondante.

La tâche de sélection de réponse est basée sur l'hypothèse qu'il existe toujours une réponse correcte pour chaque question. Cependant, ce n'est pas toujours le cas dans un système réel de questions-réponses [3]. En effet, on souhaite que le système ne fournisse pas de réponse si un utilisateur :

- pose une question qui appartient au domaine, mais qui n'est pas supportée par le système de questions-réponses ;
- pose une question hors domaine ;
- entre du texte qui n'est pas une question.

La tâche de sélection optionnelle de réponse (*answer triggering*) offre cette possibilité. Les différentes approches pour la réaliser ont donc été explorées.

2 Solutions potentielles

Pour compléter la tâche de sélection optionnelle de réponse, les approches décrites dans le tableau 1 sont possibles [4]. Ces approches visent à réaliser deux sous-tâches : déterminer s'il existe une réponse correcte et sélectionner une réponse [4].

Approche	**Description**
Conjointe	Un modèle optimise conjointement les sous-tâches de détection de l'existence d'une réponse correcte et de sélection de réponse.
Séquentielle avec rejet en amont	Un premier modèle détermine s'il existe une réponse correcte parmi les réponses candidates. Si c'est le cas, un deuxième modèle détermine la meilleure réponse et la retourne.
Séquentielle avec rejet en aval	Pour une question donnée, un modèle donne un score à chaque réponse candidate. Si le plus haut score est supérieur ou égal à un seuil prédéterminé, la réponse candidate associée est retournée.

Tableau 1 – Approches de sélection optionnelle de réponse.

L'approche séquentielle avec rejet en aval a été jugée préférable, notamment parce que le seuil peut être modifié facilement sans réentraîner le modèle.

Pour accomplir la sous-tâche de sélection de réponse, les approches listées dans le tableau 2 ont été considérées.

Approche	**Description**
Apprentissage de classement	Un modèle apprend à classer les réponses candidates à une question pour que les meilleures réponses soient en haut du classement [5].
Recherche d'information par similarité sémantique	Un modèle neuronal encode dans des documents la représentation sémantique vectorielle de questions fréquemment posées [6]. Ces documents sont ensuite stockés dans une base de données optimisée pour la recherche des plus proches voisins [6]. Lorsqu'un utilisateur pose une question, celle-ci est encodée avec le modèle neuronal et une recherche des plus proches voisins est effectuée pour trouver la question fréquemment posée qui présente le plus de similarités sémantiques [6]. La réponse associée peut ensuite être retournée par le système.
Classification d'intentions ou de texte	La classification d'intentions est un problème de classification de texte où les classes correspondent aux différentes intentions. Dans le contexte d'un système de questions-réponses, les intentions correspondent aux questions fréquemment posées. Lorsqu'un utilisateur pose une question, celle-ci est vectorisée à l'aide d'un modèle de vectorisation, puis le classifieur prédit une intention. La réponse associée est retournée à l'aide d'une table de correspondance entre intentions et réponses.

Tableau 2 – Approches de sélection de réponse.

Parmi ces approches, seules les deux dernières ont été retenues. L'apprentissage de classement a été considéré comme moins intéressant. En effet, pour entraîner un modèle d'apprentissage de classement, il faut posséder un ensemble de données dans lequel chaque exemple est associé à un score/degré de pertinence [7], ce qui n'est pas simple à produire. Par ailleurs, comme nos agents conversationnels affichent seulement la meilleure réponse, il n'est pas vraiment nécessaire d'optimiser l'ordre des autres réponses candidates.

3 Définition du système

Pour accomplir la tâche de sélection optionnelle de réponse en utilisant la classification d'intentions ou de texte, le système représenté dans la figure 1 peut être utilisé.

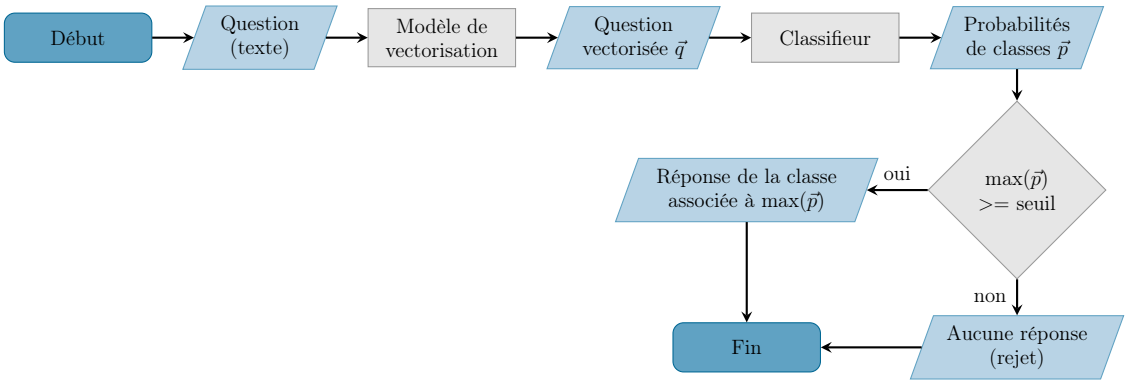


FIGURE 1 – Pipeline du système de questions-réponses.

Dans ce système, le texte de la question d'un utilisateur est d'abord vectorisé par un modèle de vectorisation. Puis, ce vecteur est fourni en entrée à un classifieur dont les classes correspondent aux questions fréquemment posées. Ce classifieur émet ensuite une probabilité pour chaque classe, et la probabilité la plus élevée est ensuite comparée à un seuil. Si cette probabilité est supérieure ou égale au seuil, le système retourne la réponse de la classe correspondant à cette probabilité. Sinon, il ne fournit pas de réponse.

Ce système peut également être utilisé pour l'approche de sélection de réponse par similarité sémantique. Plutôt que d'utiliser une base de données optimisée pour la recherche des plus proches voisins, on peut utiliser un classifieur des k plus proches voisins.

4 Ensemble de données

Un ensemble de données adéquat pour le développement de ce système de questions-réponses a ensuite été sélectionné. L'ensemble de données bancaires BANKING77 [8], créé par l'entreprise de solutions conversationnelles PolyAI, a été choisi. Bien qu'il soit prévu pour la tâche de classification d'intentions et non de sélection de réponse, il contient assez d'exemples formulés sous forme de question et suffisamment d'intentions (77) pour être représentatif de la taille d'une FAQ.

Une particularité intéressante de cet ensemble de données est qu'il possède trois variantes : *10*, *30* et *full*. Les variantes *10* et *30* ont un ensemble d'entraînement qui contient seulement 10 et 30 exemples par intention, soit un sous-ensemble des exemples de la variante *full*. Toutes les variantes ont le même ensemble de test, qui contient 40 exemples par intention.

5 Méthodologie

Le système de questions-réponses présenté précédemment contenait deux modèles : un de vectorisation et un de classification. Des expériences ont donc été réalisées dans le but de déterminer la meilleure combinaison de modèles.

Pour chaque variante de l'ensemble de données et chaque combinaison de modèle de vectorisation et de classifieur, la procédure suivante a été effectuée :

- Les données d'entraînement ont d'abord été vectorisées avec un modèle de vectorisation.
- Un classifieur a été entraîné avec une procédure de recherche d'hyperparamètres par validation croisée.
- Les données de test vectorisées ont été fournies au classifieur pour qu'il puisse générer des prédictions à partir de celles-ci.
- Enfin, ces prédictions ont été évaluées avec la métrique d'exactitude (*accuracy*), qui calcule le pourcentage de prédictions correctes.

6 Modèles de vectorisation

Les modèles de vectorisation traditionnels et neuronaux contenus dans le tableau 3 ont été évalués.

Modèle	**Description**
bag-of-words	Représente un texte par le nombre d'occurrences des mots contenus dans celui-ci sans tenir compte de leur position [9].
TF-IDF	Comme bag-of-words, mais considère également le nombre d'occurrences des mots dans l'ensemble des textes pour accorder moins de poids aux mots se retrouvant dans beaucoup de textes (ex : déterminants) [9].
BERT (*average embeddings*)	Un modèle de langue BERT préentraîné qui accepte en entrée un texte et produit en sortie une représentation vectorielle dont on fait la moyenne pour obtenir un plongement de phrase (*sentence embedding*) [10].
Sentence-BERT	Un modèle BERT préentraîné et optimisé sur plusieurs tâches de nature sémantique qui accepte en entrée un texte et produit en sortie un plongement de phrase [10].
Universal Sentence Encoder (USE)	Un réseau de neurones préentraîné simultanément sur plusieurs tâches de nature sémantique qui accepte en entrée un texte et produit en sortie un plongement de phrase [11]. Dans la variante de base, le réseau de neurones est un Deep Averaging Network [12]. Dans la variante large, c'est l'encodeur d'un Transformer [13].

Tableau 3 – Modèles de vectorisation.

Les modèles traditionnels bag-of-words et TF-IDF ont été utilisés pour fins de référence (*baselines*) pour les expériences. Ils sont basés uniquement sur l'ensemble d'entraînement du jeu de données utilisé pour une expérience donnée. Les modèles neuronaux sont quant à eux préentraînés sur des ensembles de données externes.

Pour chacun de ces modèles de vectorisation, des expériences de sélection de réponse ont été réalisées avec différents classifieurs.

7 Expériences avec classifieur KNN

L'approche de recherche d'information par similarité sémantique décrite plus tôt encode une représentation vectorielle des questions fréquentes dans une base de données optimisée pour la recherche des plus proches voisins. Pour évaluer le potentiel de cette approche, des expériences ont été réalisées avec un classifieur des k plus proches voisins, aussi appelé KNN (KNeighborsClassifier [9]).

7.1 Distance cosinus

Dans un premier temps, des expériences ont été réalisées avec un KNN avec distance cosinus. Pour chacune des variantes de l'ensemble de données, le modèle de vectorisation USE large est celui qui a le mieux performé. Il a aussi été observé que le choix du modèle de vectorisation avait un impact considérable sur la performance. Cependant, même pour le meilleur modèle de vectorisation, l'exactitude laissait encore place à l'amélioration. En effet, pour la variante *10* de l'ensemble de données, l'exactitude était inférieure à 80% (78.44%).

7.2 Distance apprise

Dans un deuxième temps, d'autres expériences ont été réalisées avec une fonction de distance apprise dans le but d'améliorer la performance. L'apprentissage de fonction de distance (*metric learning*) a pour but d'apprendre une fonction de distance propre à une tâche spécifique et est bénéfique lorsque combinée à un modèle des plus proches voisins.

L'algorithme d'apprentissage supervisé Local Fisher Discriminant Analysis (LFDA [14] [15]) a été utilisé. Il vise à rapprocher les exemples d'une même classe et éloigner les exemples appartenant à des classes différentes. Avec cette fonction de distance, c'est le modèle USE large qui a obtenu

la meilleure performance pour les variantes *30* et *full*. Toutefois, cela représente une amélioration relative très légère par rapport à la performance obtenue avec la fonction de distance cosinus.

Dans un article de blog de PolyAI paru après la réalisation des expériences décrites ici, de meilleurs résultats ont été rapportés avec une méthode d'apprentissage de fonction de distance utilisant un réseau de neurones [16]. L'avantage principal du réseau de neurones par rapport à l'algorithme LFDA est qu'il peut apprendre une fonction de transformation non linéaire. Les expériences de PolyAI sur différents ensembles de données montraient toutefois qu'un réseau de neurones performait un peu mieux qu'un KNN avec apprentissage profond de fonction de distance [16]. Il n'est donc pas clair que les bénéfices du KNN avec distance apprise soient suffisamment grands pour justifier la complexité d'implémentation supplémentaire.

8 Expériences avec classifieur SVM linéaire et classifieur perceptron multicouches (MLP)

Suite à la lecture d'un article ayant conclu que pour une tâche de classification de texte, le classifieur SVM avait mieux performé qu'un KNN avec distance apprise [17], des expériences ont été réalisées avec un SVM linéaire (LinearSVC [9]). Pour toutes les variantes de l'ensemble de données et tous les modèles de vectorisation, le SVM linéaire a effectivement mieux performé que le KNN (avec distance cosinus et distance apprise). Quant aux modèles de vectorisation, c'est encore une fois USE large qui a le mieux performé.

Malgré la très bonne performance du modèle SVM sur la tâche de sélection de réponse, ce n'est pas le modèle le plus approprié pour la tâche de sélection optionnelle de réponse. Comme le SVM ne retourne pas directement des probabilités, un module de calibration doit être utilisé. Cependant, celles-ci ne sont pas très bien calibrées [9]. Pour que le système de questions-réponses soit en mesure de bien rejeter les questions hors domaine, il était donc préférable d'utiliser un modèle d'apprentissage automatique retournant directement des probabilités, comme un réseau de neurones.

D'autres expériences ont ainsi été réalisées avec un perceptron multicouches (MLPClassifier [9]). Cela a permis d'obtenir un modèle avec une performance légèrement supérieure à celle du SVM, mais qui est mieux adapté à la tâche de sélection optionnelle de réponse, car il retourne des probabilités.

9 Évaluation sur la tâche de sélection de réponse

Les expériences de sélection de réponse décrites précédemment ont permis d'établir que le meilleur modèle était celui combinant le Universal Sentence Encoder (USE) large avec un MLP. Pour évaluer ce modèle, une comparaison a été effectuée avec les modèles de classification d'intentions des engins NLU de Dialogflow ES et Rasa 1.10.12.

Pour Rasa, le pipeline NLU qui a été utilisé est listé ci-dessous. Il est basé sur la configuration qui était recommandée pour l'anglais.

```
language: "en"
pipeline:
- name: ConveRTTokenizer
- name: ConveRTFeaturizer
- name: CountVectorsFeaturizer
- name: CountVectorsFeaturizer
  analyzer: "char_wb"
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  entity_recognition: False
  epochs: 100
  random_seed: 42
```

Les ensembles d'entraînement et de test de chaque variante de l'ensemble de données BANKING77 ont d'abord été transformés dans le format attendu par Dialogflow et Rasa. Chaque modèle NLU a ensuite été entraîné sur l'ensemble d'entraînement et évalué sur l'ensemble de test avec la métrique d'exactitude. Les résultats obtenus sont listés dans le tableau 4.

	Variante		
Modèle	**10**	**30**	**full**
USE (large) + MLP	86.33	90.03	92.69
Dialogflow ES	74.77	83.86	90.26
Rasa	69.03	86.33	91.17

TABLEAU 4 – Résultats de l'évaluation sur l'ensemble de données BANKING77.

On peut observer que pour toutes les variantes de l'ensemble de données, le modèle USE combiné à un MLP a mieux performé que Dialogflow ES et Rasa. Il faut toutefois préciser que la performance du modèle basé sur Rasa pourrait sûrement être améliorée en faisant une optimisation des hyperparamètres.

10 Évaluation sur la tâche de sélection optionnelle de réponse

L'évaluation précédente de la tâche de sélection de réponse avait pour objectif de vérifier la capacité d'un modèle à fournir la réponse correcte à une question supportée par le système. Pour la tâche de sélection optionnelle de réponse, un objectif supplémentaire est d'évaluer la capacité d'un modèle à rejeter une question hors domaine.

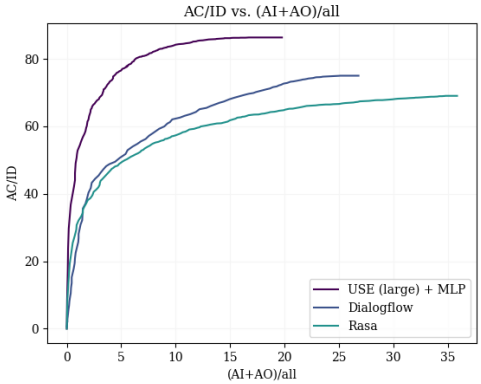
Pour ce faire, une nouvelle intention "out_of_scope" contenant 234 exemples hors domaine a été ajoutée à l'ensemble de test des trois variantes de l'ensemble de données BANKING77. Plus spécifiquement, ces exemples sont des questions en lien avec la COVID-19 tirées de cet ensemble de données [18].

Pour comparer la performance des modèles sur l'ensemble de test, une courbe ROC (*Receiver Operating Characteristic*) a été utilisée. Ce graphique d'une métrique en fonction d'une autre (ici *AC/ID* en fonction de *(AI+AO)/all*) est généré en faisant varier un seuil. Les définitions contenues dans le tableau 5 ont été utilisées.

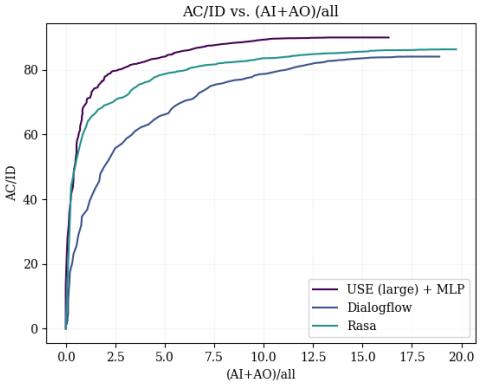
	Définition
AC	Nombre de prédictions correctes dont le score de confiance est supérieur ou égal au seuil
AI	Nombre de prédictions incorrectes dont le score de confiance est supérieur ou égal au seuil
AO	Nombre de prédictions hors domaine dont le score de confiance est supérieur ou égal au seuil
ID	Nombre d'exemples qui appartiennent au domaine
all	Nombre d'exemples total

TABLEAU 5 – Définitions utilisées pour le calcul des métriques.

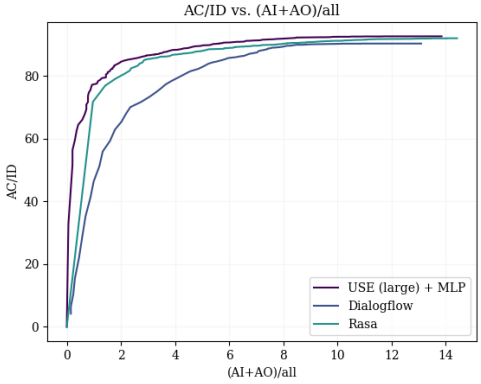
Les courbes ROC obtenues sont représentées dans la figure 2. Un modèle dont la courbe est au-dessus d'une autre est meilleur. Un résultat parfait serait d'atteindre le coin gauche supérieur du graphe, mais cela ne se produit pas en pratique.



(a) Variante *10*



(b) Variante *30*



(c) Variante *full*

FIGURE 2 – Courbes ROC pour l'ensemble de données BANKING77 modifié avec données hors distribution.

On peut constater que, comme dans l'évaluation précédente, le modèle USE combiné à un MLP a obtenu la meilleure performance, surtout pour la variante *10*. On peut également constater que si les modèles avaient une performance similaire sur la tâche de sélection de réponse pour la variante *full*, ce n'est pas le cas pour la tâche de sélection optionnelle de réponse. Ceci démontre l'importance de cette tâche pour comparer différents modèles.

Pour tous les modèles, la tâche de rejeter des exemples hors domaine s'est avérée difficile, et les scores de confiance étaient parfois plus élevés qu'attendus. Par exemple, le modèle de Rasa a retourné l'intention "age_limit" avec un score de confiance de plus de 95% pour les exemples hors domaine suivants :

- "Will educational childcare centres provide meals and snacks ?"
- "Should an employee who is 70 years of age continue to work ?"
- "Can children accompany their parents in grocery stores, drugstores and other public spaces ?"

On peut expliquer ceci par le fait que l'intention "age_limit" était la seule à posséder des exemples en lien avec l'âge et les enfants. Une solution potentielle à ce genre de problème serait de rajouter des exemples similaires à une intention "out_of_scope" dans l'ensemble d'entraînement [19].

11 Conclusion

En conclusion, ces expériences ont permis de livrer une preuve de concept d'un système de questions-réponses performant basé sur un modèle d'apprentissage automatique. Un juste équilibre entre la performance et l'efficacité a pu être atteint grâce à l'utilisation du modèle de vectorisation

préentraîné USE large combiné à un MLP.

Remerciements

Cet article est basé sur les travaux réalisés dans le cadre de mon stage de maîtrise, effectué en 2020. Je tiens à remercier Yves Normandin (Nu Echo) et Sherjil Ozair (Mila) pour la supervision de ce stage. Je tiens aussi à remercier Guillaume Voisine (Nu Echo) pour l'édition de ce texte.

Références

[1] Y. Normandin. "Question answering experiments with the Dialogflow FAQ Knowledge Connectors." (2020), adresse : https://www.nuecho.com/news-events/question-answering-experiments-with-the-dialogflow-faq-knowledge-connectors/.

[2] M. Feng, B. Xiang, M. R. Glass, L. Wang et B. Zhou, "Applying deep learning to answer selection : A study and an open task," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, p. 813-820. DOI : 10.1109/ASRU.2015.7404872.

[3] Y. Yang, W.-t. Yih et C. Meek, "WikiQA : A Challenge Dataset for Open-Domain Question Answering," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2015, p. 2013-2018. DOI : 10.18653/v1/D15-1237.

[4] J. Zhao, Y. Su, Z. Guan et H. Sun, "An End-to-End Deep Framework for Answer Triggering with a Novel Group-Level Objective," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2017, p. 1276-1282. DOI : 10.18653/v1/D17-1131.

[5] T. M. Lai, T. Bui et S. Li, "A Review on Deep Learning Techniques Applied to Answer Selection," in *Proceedings of the 27th International Conference on Computational Linguistics*, Association for Computational Linguistics, 2018, p. 2132-2144.

[6] Y. Yang et A. Ahmad. "Multilingual Universal Sentence Encoder for Semantic Retrieval." (2019), adresse : https://ai.googleblog.com/2019/07/multilingual-universal-sentence-encoder.html.

[7] H. Li, *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers, 2011, p. vi, 4. DOI : 10.2200/S00348ED1V01Y201104HLT012.

[8] I. Casanueva, T. Temcinas, D. Gerz, M. Henderson et I. Vulic, "Efficient Intent Detection with Dual Sentence Encoders," in *Proceedings of the 2nd Workshop on NLP for Conversational AI*, Association for Computational Linguistics, 2020, p. 38-45. DOI : 10.18653/v1/2020.nlp4convai-1.5.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn : Machine Learning in Python," *Journal of Machine Learning Research*, t. 12, p. 2825-2830, 2011, documentation et guide utilisateur des modèles (CountVectorizer, TfidfVectorizer, KNeighborsClassifier, LinearSVC, MLPClassifier), documentation des modules ("Probability calibration").

[10] N. Reimers et I. Gurevych, "Sentence-BERT : Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019, p. 3982-3992. DOI : 10.18653/v1/D19-1410.

[11] D. Cer, Y. Yang, S. Kong et al., "Universal Sentence Encoder," *CoRR*, 2018. arXiv : 1803.11175.

[12] M. Iyyer, V. Manjunatha, J. Boyd-Graber et H. Daumé III, "Deep Unordered Composition Rivals Syntactic Methods for Text Classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, Association for Computational Linguistics, 2015, p. 1681-1691. DOI : 10.3115/v1/P15-1162.

[13] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc., 2017, p. 6000-6010.

[14] M. Sugiyama, "Dimensionality Reduction of Multimodal Labeled Data by Local Fisher Discriminant Analysis," *Journal of Machine Learning Research*, t. 8, n° 37, p. 1027-1061, 2007.
[15] W. de Vazelhes, C. Carey, Y. Tang, N. Vauquier et A. Bellet, "metric-learn : Metric Learning Algorithms in Python," *Journal of Machine Learning Research*, t. 21, n° 138, p. 1-6, 2020.
[16] E. Liberis. "Intent Classification with Geometrically-Friendly Embeddings." (2020), adresse : https://www.polyai.com/intent-classification-geometrically-friendly-embeddings/.
[17] K. Q. Weinberger et L. K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, t. 10, p. 207-244, 2009.
[18] M. Bronzi, J. Pinto, J. Ghosn et al. "A Question Answering System in Response to the COVID-19 Crisis." (2020), adresse : https://drive.google.com/file/d/1chJyp6mEhieL13EzbX_Xpf8m3MenDuDE/view.
[19] S. Larson, A. Mahendran, J. J. Peper et al., "An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, p. 1311-1316. doi : 10.18653/v1/D19-1131.