Architecture Asymétrique pour les Modèles Neuronaux d'Appariement de Textes

Thiziri Belkacem, Taoufiq Dkaki, Jose G. Moreno, Mohand Boughanem

Architecture Asymétrique pour les Modèles Neuronaux d'Appariement de Textes

Thiziri Belkacem — Taoufiq Dkaki — José G. Moreno — Mohand Boughanem

Université de Toulouse, IRIT UMR 5505 CNRS
{thiziri.belkacem, taoufiq.dkaki, jose.moreno, mohand.boughanem}@irit.fr

RÉSUMÉ. *Dans les modèles neuronaux d'appariement de textes, les entrées subissent les mêmes transformations pour construire les représentations correspondantes. La nature de la tâche d'appariement est défini à partir du type des entrées du modèle et de la relation entre elles. Nous distinguons deux types d'appariement : (1) l'appariement symétrique fait référence aux tâches d'appariement à des entrées de même nature, telles que l'identification des paraphrases et la classification de documents. (2) l'appariement asymétrique concerne des tâches à des entrées de natures différentes, telle que l'appariement document-requête ou question-réponse. Généralement, les représentations des entrées sont construites indépendamment de leurs nature à partir des vecteurs de leurs mots. Nous proposons une approche permettant de prendre en compte la nature de la tâche, (1) ou (2), et de mieux traiter les entrées. Nous utilisons un modèle d'attention pour étendre des modèles de l'état de l'art. Les résultats expérimentaux montrent que l'adaptation de l'architecture du modèle au type de la tâche permet d'améliorer les performances de plusieurs modèles neuronaux bien connus.*

ABSTRACT. *In neural text matching models, the inputs undergo the same transformations to construct the corresponding representations. The nature of the task is defined w.r.t the relationship between the inputs and their types. We distinguish two types of matching: (1) Symmetric matching refers to matching tasks with similar inputs nature, such as paraphrase identification and document classification. (2) asymmetric matching refers to tasks with inputs of different nature, such as document-query or question-answer matching. Generally, the input representations are constructed, independently of their nature, from the vectors of their words. We propose an approach to take into account the type of the task, (1) or (2), and to better process the inputs. We use an attention model to extend state of the art models. Experimental results show that adapting the model architecture to the task type improves the performances of several neural models.*

MOTS-CLÉS : *Asymétrie$_1$, Modèles d'attention$_2$, Appariement de textes$_3$.*

KEYWORDS: *Asymmetry$_1$, Attention Models$_2$, Text Matching$_3$.*

1. Introduction

Les réseaux de neurones profonds ont été utilisés dans plusieurs modèles de traitement automatique de la langue et de recherche d'information (RI). Nous définissons la nature d'une tâche d'appariement en fonction du type des entrées du modèle ainsi que la relation entre elles. Nous distinguons deux catégories : *l'appariement symétrique* consiste à identifier si deux textes sont sémantiquement similaires. Elle concerne des entrées de même type, tel que l'achèvement des phrases (Zweig *et al.*, 2012 ; Wan *et al.*, 2016), la classification des documents (Pang *et al.*, 2016 ; Kim, 2014), l'identification des paraphrases (Yin et Schütze, 2015 ; Tomar *et al.*, 2017) et l'identification des paires de questions (Abishek *et al.*, 2019). La seconde catégorie est *l'appariement asymétrique*. Elle vise à savoir si un texte apporte l'information recherchée dans un autre texte. Le type des textes d'entrée est alors différent et leur similarité est déterminée non seulement par les liens sémantiques et lexicaux mais aussi par leur complémentarité : l'une des entrées contient la description de l'information recherchée, mais pas l'information elle même et qui est fournit par l'autre entrée. Cette catégorie inclut principalement l'appariement question-réponse (Yang *et al.*, 2016a ; Peng et Liu, 2018) et l'appariement document-requête (Shen *et al.*, 2014 ; Pang *et al.*, 2017). Dans la plupart des modèles existants, les entrées sont d'abord représentées par un ensemble de vecteurs de mots, puis elles subissent le même traitement afin de construire leurs représentations. Ce mécanisme est adopté dans la plupart des modèles d'appariement de texte existants, indépendemment de la nature de la tâche considérée. Pour tenir compte de l'aspect asymétrique de certaine tâches d'appariement de texte, nous croyons que le modèle d'appariement doit se concentrer davantage sur les mots les plus importants des entrées pour mieux aborder la tâche d'appariement. Les modèles d'attention (Bahdanau *et al.*, 2014 ; Yin *et al.*, 2016) peuvent fournir le moyen de le faire, il apprennent des coefficients permettant aux processus qui suivent de se concentrer sur les mots les plus importants des entrées.

Dans cet article, nous proposons une architecture basée sur l'attention et permettant de mieux gérer l'aspect asymétrique de certaines tâches d'appariement, en faisant plus attention aux mots les plus importants dans les entrées correspondantes. Nos principales contributions sont les suivantes :

1) Nous proposons une architecture d'appariement qui tient compte de l'aspect *asymétrique* de certaines tâches d'appariement de textes.

2) Nous étendons plusieurs modèles de l'état de l'art dans l'architecture proposée.

3) Nous menons une étude expérimentale comparative de quelques modèles existants, par rapport à leurs extensions correspondantes, dans le traitement des différents types de tâches d'appariement, symétrique ou asymétrique.

2. Travaux Liés

Les modèles neuronaux d'appariement de textes sont basées sur une architecture siamoise (Bromley *et al.*, 1994), où les deux entrées subissent le même type de traitement, impliquant un traitement indifférent des tâches d'appariement symétriques et asymétriques. Dans cet article, nous distinguons deux familles de modèles neuronaux : Les modèles *sans attentions* et qui sont centrés sur la réalisation de la tâche d'appa-

riement. Les modèles d'*attention* et qui sont centrés sur le traitement des éléments des entrées en se basant sur leur importance pour mieux réaliser la tâche abordée.

Dans le cadre des tâches d'appariement symétrique, les modèles sans attention traitent les différentes entrées de manière symétrique. Dans (Hu *et al.*, 2014), les auteurs proposent deux architectures, ARC-I et ARC-II, basées sur des réseaux à convolution pour l'appariement des séquences de texte de même nature. ARC-I construit les représentations des phrases d'entrées en utilisant des séquences de couches à convolution et de *max-pooling*, puis calcule le score d'appariement. ARC-II applique une série de couches à convolution et de *max-pooling* à la matrice de similarité entre les vecteurs des mots des deux entrées, pour calculer le score final de similarité. Pang et al (Pang *et al.*, 2016) ont proposé le modèle MatchPyramid pour l'identification des paraphrase et l'appariement des citations. Dans MatchPyramid, les entrées sont d'abord représentées à l'aide du plongement lexical des mots, puis une matrice d'appariement entre les mots est calculée et fournit à une séquence de couches à convolution et de *max-pooling*, afin d'extraire des signaux d'interaction de plus haut niveau. Dans (Nicosia et Moschitti, 2017), le modèle apprend d'abord les représentations des entrées par un réseau siamois utilisant des encodeurs à des paramètres partagés. Puis, compare ses représentations en utilisant la mesure de distance euclidienne. Enfin, une couche entièrement connectée combine ces représentations et calcule la probabilité d'appariement des deux entrées. Dans (Liu *et al.*, 2018), les auteurs proposent un modèle d'appariement de documents composé d'un graphe d'interaction de concepts pour calculer les représentations des documents d'entrée, dont les noeuds représentent des concepts et les arcs représentent les interactions entre ces concepts. Un réseau siamois est ensuite utilisé pour encoder apprendre les représentations de chacun des noeuds. Les caractéristiques des noeuds sont ensuite agrégées pour calculer le score d'appariement des deux documents d'entrées.

Dans le cadre des tâches asymétriques, les modèles proposés traitent aussi les entrées de manière symétrique. Dans (Huang *et al.*, 2013) les auteurs proposent le modèle DSSM pour l'appariement document-requête. Ce modèle utilise une architecture profonde afin de projeter un vecteur de haute dimension dans un espace latent de plus petite dimension. La même séquence de transformations est appliquée pour les différentes entrées, pour construire les représentations vectorielles correspondantes. Les vecteurs de sortie sont ensuite comparés par une fonction cosinus pour calculer le score de similarité final. Dans (Shen *et al.*, 2014), les auteurs proposent le modèle CDSSM, qui étend le modèle DSSM (Huang *et al.*, 2013) avec une couche à convolution pour l'appariement document-requête. Dans CDSSM, une couche de convolution est appliquée pour les entrées afin d'en extraire des interactions contextuelles, ensuite la même succession de couches que dans le DSSM est utilisée pour calculer les représentations des entrées ainsi que leur score de similarité. Wan et al (Wan *et al.*, 2016) ont proposé le modèle MVLSTM basé sur la position, pour l'appariement question-réponse. Dans le MVLSTM, la question aussi bien que la réponse est traitée par la même couche de réseau à mémoire à long terme et à court terme (LSTM) bidirectionnelle (bi-LSTM) pour construire la représentation correspondante basée sur les différentes positions de chacun de ses mots. Des matrices d'interaction entre les entrées

sont ensuite calculées et passées à travers une couche de *max-pooling* et des couches entièrement connectées pour calculer le score de similarité. Dans (Mitra *et al.*, 2017), les auteurs ont proposé le modèle DUET ayant une architecture en duo pour l'appariement document-requête. L'architecture de DUET est composée de deux modèles parallèles, qui effectuent chacun un traitement symétrique pour les deux entrées : le modèle *local* utilisant la matrice d'interaction entre les mots de la requête et ceux du document, le modèle *distribué* apprend des représentations vectorielles pour la requête et le document avant de les comparer. Le score final est calculé par le DUET en combinant les scores calculés par les deux modèles *local* et *distribué*.
Dans l'ensemble des modèles susmentionnés, les deux entrées du modèle neuronal sont traitées de manières symétrique malgré l'aspect asymétrique des tâches abordées.

Concernant la famille des modèles d'attention, ils sont utilisés en traduction automatique (Bahdanau *et al.*, 2014), la classification des sentiments (Yang *et al.*, 2016b ; Parikh *et al.*, 2016) et l'identification des paraphrases (Yin *et al.*, 2016). Ces modèles permettent de déterminer le noyau de l'information véhiculée par une séquence donnée et de se focaliser sur les éléments discriminants. Formellement, étant donnée une séquence de mots S, le modèle d'attention apprend un vecteur de coefficients α qui détermine combien d'attention doit être accordée à chaque élément de S, en fonction de la tâche à accomplir. Dans l'ensemble de ces modèles, aussi bien que la famille des modèles sans attention, les deux entrées sont traitées de la même manière malgré l'aspect asymétrique de certaines des tâches abordées.

Dans (Yang *et al.*, 2016b), les auteurs proposent un réseau d'attention hiérarchique pour la classification des documents. Le modèle combine les scores d'attention du niveau de mot et de phrase, dans une architecture de réseau de neurones récurrent. Le texte en entrée est d'abord représenté en utilisant les vecteurs de ses mots, puis une première couche GRU[1] bidirectionnelle (bi-GRU) calcule la représentation vectorielle du texte d'entrée. Une représentation intermédiaire est des coefficients d'attention sont ensuite calculés pour chaque mots, puis transmis pour une autre couche bi-GRU pour calculer une représentation finale en combinant les représentations de chaque phrase avec son poids d'attention. La softmax de cette représentation est ensuite utilisée pour la classification des documents d'entrée. Dans (Yang *et al.*, 2016a), les auteurs proposent le modèle d'attention aNMM pour l'appariement question-réponse. aNMM utilise un réseau de neurones avec un schéma de pondération à valeur partagée et un réseau portique (*gating network*) en fonction des mots de la question. Le schéma de pondération à valeur partagée consiste à partager les mêmes poids de connexion par certains éléments des couches MLP, en fonction de l'intensité du signal d'appariement au niveau de ses éléments. La question et la réponse sont représentées par leurs vecteurs de mots, puis une matrice d'interaction est calculée en utilisant une similarité cosinus. Dans cette matrice, les éléments de la même ligne ayant une valeur du même intervalle sont agrégés. Par conséquent, ces éléments partagent les mêmes poids de connexion. Les vecteurs de mots de la question d'entrée sont utilisés pour calculer les

1. Est un type particulier de réseaux LSTM qui utilise un mécanisme de contrôle pour suivre l'état d'une séquence donnée sans utiliser de cellules mémoire séparées (Bahdanau *et al.*, 2014)

scores d'attention pour les poids partagés, ainsi, déterminer les signaux d'interaction les plus forts à prendre en compte dans le processus d'appariement.
Dans (Peng et Liu, 2018), les auteurs proposent un modèle à convolution basé sur l'attention pour l'appariement question-réponse représentées par les vecteurs de leurs mots. Le modèle est composé de deux modules parallèles : un pour le niveau mot et un autre pour le niveau phrase. Premièrement, les matrices d'appariement au niveau des mots et des phrases sont calculées en utilisant la fonction cosinus, puis une fonction *max-pooling* est appliquée aux deux matrices d'appariement, afin de sélectionner les signaux d'interaction les plus importants à chaque niveau. Ensuite, les poids d'attention sont calculés à la fois au niveau des mots et des phrases. Enfin, les vecteurs de score des deux niveaux sont concaténés et transmis à une couche entièrement connectée pour calculer la probabilité de similarité finale.

Tous les modèles présentés dans cette section traitent de manière symétrique les différentes entrées, quelle que soit la nature de la tâche abordée. Le traitement neuronal asymétrique des entrées a déjà été discuté dans (Bordes *et al.*, 2011), où le modèle proposé apprend plusieurs relations asymétriques en même temps dans une base de connaissances. Cependant, dans l'appariement de texte, aucun des travaux précédents n'a fourni le moyen de prendre en compte l'aspect asymétrique de certaines tâches.

3. Appariement Asymétrique de Textes

Dans cette section, nous présentons d'abord la motivation principale de ce travail, puis l'approche d'appariement que nous proposons afin de tenir compte de l'aspect asymétrique de certaines tâches d'appariement de texte.

3.1. *Motivation*

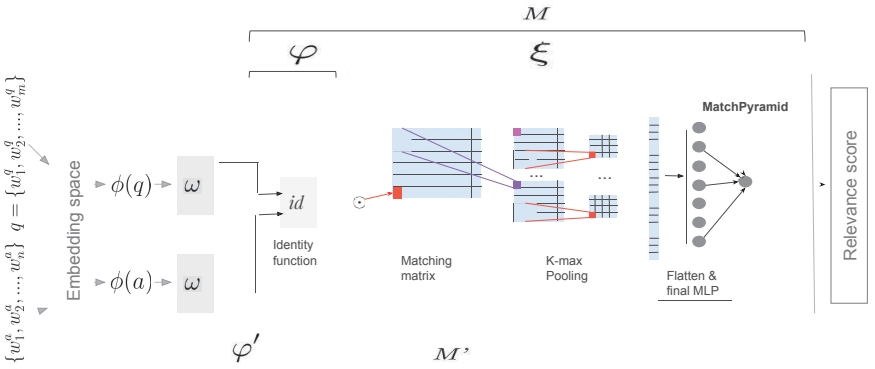
Afin de mettre en valeur notre intuition pour l'approche que nous proposons, nous considérons la tâche d'appariement asymétrique *question-réponse*. Cette dernière peut être considérée comme un puzzle où les pièces manquantes doivent être trouvées et assemblées, de façon logique, pour résoudre le problème. Chaque vide dans ce puzzle décrit la partie manquante. Il en va de même pour l'appariement question-réponse, en effet : un vide représente l'information manquante et l'ensemble des vides peut être rempli par l'ensemble des informations apportées par une ou plusieurs réponses. Toutes les réponses ne peuvent être appropriées, seule la réponse qui remplit correctement le vide pouvant être considérée. Afin d'illustrer cette idée, la figure 1 montre l'exemple d'une question de la collection WikiQA, avec deux réponses candidates. Dans cet exemple, pour répondre correctement à la question *Q*, nous devons nous concentrer sur chacun des mots soulignés décrivant chacune des parties manquantes dans le puzzle correspondant. Toutes les deux réponses *A1* et *A2* contiennent plusieurs traits (les mots soulignés) correspondant à la pièce (l'information) manquante. Cependant, afin de distinguer entre les deux réponses *A1* et *A2*, le mots clés *"married to"* de la question *Q* nécessite plus d'attention que les autres mots soulignés de *Q*. En effet, le mot *"who"* fait référence à la personne qui a épousé (*"married to"*) *Mary Matalin* mais pas à *Mary Matalin* elle même. À cet effet, la solution 2 résout mieux le problème par rapport à la solution 1 et représente la pièce manquante de ce puzzle.
L'exemple précédent met en évidence l'asymétrie de la tâche d'appariement question-

Figure 1 – Dans les réponses *A1* et *A2*, les mots soulignés représentent les liens sémantiques et/ou lexicaux avec la question *Q*, les mots en caractères **gras** représentent les mots clés (description de la pièce manquante) de l'information apportée dans *A1* et *A2*. La solution 2 représente la bonne réponse car elle remplit tout les vides du puzzle.

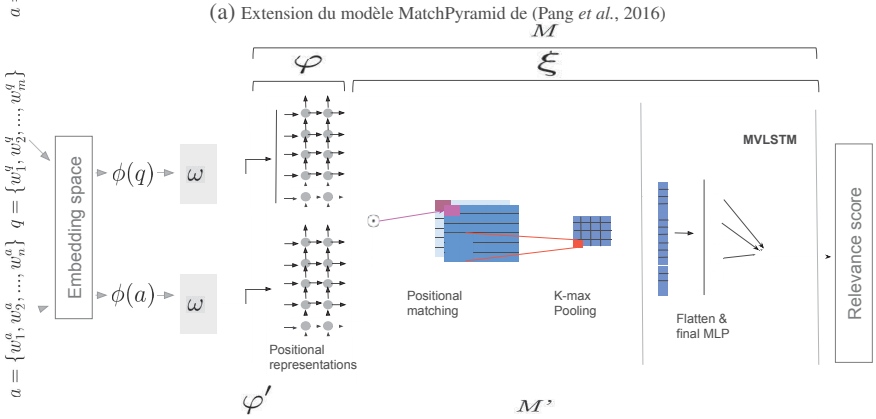
réponse. Il montre aussi l'intérêt de la prise en compte de l'importance de chacun des mots clés de la question, au delà des liens sémantiques et lexicaux entre la question et le réponse, afin de désigner la réponse correcte. Dans ce travail, nous proposons une approche pour tenir compte de ces aspects. Plus précisément, nous utilisons des couches d'attention, pour permettre au modèle d'appariement de se focaliser sur les mots les plus importants de ses différentes entrées selon la nature de la tâche abordée.

3.2. *Présentation du Modèle*

Soient une question $q = \{w_1^q, w_2^q, ..., w_m^q\}$, une réponse $a = \{w_1^a, w_2^a, ..., w_n^a\}$, un modèle d'appariement neuronal *M* et une fonction de représentation ϕ. La plupart des modèles neuronaux d'appariement de textes peuvent être résumés comme suit : premièrement, calculer les représentations $\phi(q)$ et $\phi(a)$ correspondantes à q et à a, respectivement. Puis, une fonction φ, pouvant être un bloc de couches neuronales, est utilisée pour calculer les représentations internes au réseau pour les entrées simultanément. Enfin, une fonction ξ, est utilisée pour calculer le score d'appariement final s. Nous définissons le modèle *M'* qui étend le modèle *M* avec une couche ω appliquée pour les entrées du modèle. Ce processus est illustré dans la figure 2 avec deux exemples de modèles neuronaux de l'état de l'art. Dans cet exemple, pour le modèle MatchPyramid, la fonction φ peut être vu comme une fonction identité ($id(x) = x$), où les entrées ne subissent aucune transformation (Pang *et al.*, 2016). Cependant, dans l'exemple du modèle MVLSTM, la fonction φ correspondra à un bloc de couches bi-LSTM utilisées pour construire les représentations des entrées (Wan *et al.*, 2016). Pour prendre en compte l'aspect asymétrique de certaines tâches d'appariement et ainsi de traiter les entrées selon leur types, nous définissons la fonction φ' du modèle *M'* et qui représente la fonction de transformation des entrées, comme montré dans la figure 2. La définition formelle de φ' est donnée dans l'équation 1. Concernant le bloc ξ, il représente la partie d'interaction du modèle, où l'objectif est de calculer des caractéristiques et des résultats intermédiaires jusqu'à la sortie finale.
Nous définissons la fonction φ' utilisée dans le modèle *M'* comme suit : étant donnée une séquence de mots $S = \{w_1, w_2, ..., w_k\}$ de taille k. Nous définissons un paramètre $e_S \in \{0, 1\}$ utilisé pour configurer le modèle *M'*, selon la nature de la tâche abordée. Tel que : si $e_S = 0$ pour les deux entrée, alors *M'* est similaire à *M* et si $e_S = 1$, alors la couche ω est activée pour l'entrée correspondante, tel montré dans



(a) Extension du modèle MatchPyramid de (Pang *et al.*, 2016)



(b) Extension du modèle de MVLSTM (Wan *et al.*, 2016)

Figure 2 – L'architecture d'appariement asymétrique *M'*, par extension du modèle original *M*, illustrée en utilisant deux exemples de modèles de l'état de l'art. φ' manie l'asymétrie grâce à la couches d'attention ω qui est activée selon la configuration de l'équation 1. ξ est une séquence de couches de traitement utilisées pour calculer le score correspondant s.

la figure 2, avec $S \in \{q, a\}$. Si la tâche d'appariement abordée est symétrique, alors $e_S = 1$ pour les deux entrées simultanément.

$$\varphi'(\phi(S)) = \begin{cases} \varphi \circ \omega(\phi(S)) & si \;\; e_S = 1 \\ \varphi(\phi(S)) & sinon \end{cases} \quad [1]$$

où ω est la couche d'extension basée sur l'attention telle montrée dans la figure 2. Nous définissons ω en utilisant une fonction portique (*gating*) (Yang *et al.*, 2016a ; Guo *et al.*, 2016) définie dans l'équation 2.

$$\omega(\phi(S)) = [\alpha_1 w_1, \alpha_2 w_2, ..., \alpha_t w_t, ..., \alpha_k w_k] \quad [2]$$

$$\alpha_t = \frac{exp(\boldsymbol{v}^T.w_t)}{\sum_{j=1}^{l_S} exp(\boldsymbol{v}^T.w_j)} \quad [3]$$

où $\boldsymbol{v}$ est un paramètre de la couche d'attention, c'est un vecteur de coefficients d'attention correspondants au mot w_t de la séquence S à la position t. Ainsi, les paramètres de *M'* sont $\theta_{M'} = \{\theta_M, \theta_\omega\}$ avec θ_M et θ_ω sont les paramètres du modèle original M et les paramètres de la couche d'attention ω, respectivement. Le score de pertinence d'une réponse a par rapport à une question q est alors donnée par *M'* $(a, q; \theta_{M'})$.

4. Expérimentation

Dans cette section, nous présentons l'expérimentation menée pour évaluer l'efficacité de notre approche, dans deux collections de deux types de tâches d'appariement.

4.1. *Cadre Expérimental*

Nous présentons d'abord notre cadre expérimental, en terme des collections de données, les modèles de référence et les mesures d'évaluation utilisée dant ce travail.

4.1.1. *Méthodologie d'évaluation*

La mise en œuvre des modèles neuronaux et leur processus expérimental sont réalisés[2] à l'aide du framework MatchZoo (Fan et al., 2017) pour l'appariement de textes. Comme notre contribution est applicable à des modèles existants, nous avons choisi MatchZoo pour le grand nombre de modèles de l'état de l'art qui y sont déjà implémentés. Notez qu'il existe d'autres méthodes[3] très performantes pour cette tâche, comme (Rao *et al.*, 2016) et (Tymoshenko et Moschitti, 2018), mais nous n'avons pas implémenté une architecture asymétrique pour ces méthodes du fait qu'elles ne font pas partie du framework MatchZoo.

Afin de valider nos hypothèses, nous avons évalué différentes architectures de plusieurs modèles de l'état de l'art implémentés dans ce framework. Nous utilisons un ensemble de labels pour faire référence à chacune de ces architectures pour de les distinguer les unes des autres, comme suit : l'appariement symétrique inclut le modèle de base (Original) et l'application de la couche d'attention ω (voir équation 1) pour les deux entrées du modèle simultanément (Q+A). L'appariement asymétriques inclut l'application de la couche ω pour l'une des entrées à la fois, la question uniquement (Q) ou la réponse uniquement (A). Concernant la configuration des différents modèles évalués, quelque soit le type d'appariement, symétrique ou asymétrique, nous avons opté pour la configuration des paramètres recommandée, soit dans les papiers correspondants aux différents modèles ou dans MatchZoo[4]. Concernant les hyperparamètres de MatchZoo, le tableau 1 montre les descriptions et les valeurs associées. Pour le plongement lexical des mots, tout les modèles évalués utilisent des vecteurs à 300 dimensions pré-entraînés de GloVe[5]. Pour la tâche d'appariement asymétrique dans la collection WikiQA, nous avons considéré la précision (P) et la mesure du gain cumulatif normalisé (nDCG) à différent ordres, ainsi que la précision moyenne (MAP) et le rang réciproque moyen (MRR). Pour la tâche symétrique dans la collection QuoraQP, nous avons utilisé la mesure d'exactitude[6].

4.1.2. *Collections*

Nous avons utilisé deux collections : la collection WikiQA (Yang *et al.*, 2015) est utilisée pour l'évaluation de de notre approche avec une tâche asymétrique. Cette

2. Le code est disponible sur une branche de MatchZoo accessible à l'adresse https ://github.com/thiziri/lates_MZ_and_data_pre/ .

3. https ://aclweb.org/aclwiki/Question_Answering_(State_of_the_art)

4. https ://github.com/NTMC-Community/MatchZoo/tree/1.0/examples

5. http ://nlp.stanford.edu/data/glove.840B.300d.zip

6. Nous utilisons *exactitude* pour *accuracy* afin de la distinguer de la mesure de précision classique (*Precision*). Certains auteurs préfèrent utiliser aussi le terme *précision* pour *accuracy*

collection est composée de 2477 questions et 21629 réponses. Les questions sont collectées du moteur de recherche Bing. Les réponses candidates sont des phrases de la section *résumé* des pages Wikipedia contenant l'information recherchée. Nous avons adopté la répartition des questions suggérée par MatchZoo[7], avec 10% pour les tests, 10% pour la validation et 80% l'entraînement. La deuxième collection est QuoraQP, utilisée pour l'évaluation avec une tâche symétrique, elle est composée d'un ensemble de paires de questions pouvant être similaires ou pas et provenant du forum Quora[8]. Elle contient 404351 paires de questions, dont 149306 positives et 255045 négatives. Nous avons utilisé également la répartition pré-défini par MatchZoo[9].

4.1.3. *Modèles de Référence*

Nous avons utilisé deux classes de modèles de références, comme suit :

4.1.3.1. Modèles Classiques

L'évaluation des modèles[10] suivants, ainsi que la construction de l'index correspondant, ont été réalisés en utilisant le système de recherche INDRI[11].

– *BM25* (Robertson et Walker, 1994) est un modèle de RI classique pour le tri des documents. Nous avons adopté la configuration des paramètres la plus utilisée dans l'état de l'art, avec $k1 = 1.5$ et $b = 0.75$.

– *ML* avec lissage de Jelinek Mercer (Jelinek, 1980) est un probabiliste de la RI classique, avec $\lambda_{collection} = 0.4$ et $\lambda_{document} = 0.0$.

4.1.3.2. Modèles Neuronaux

Notre contribution est applicable à des modèles neuronaux existants pour l'appariement de texte. Nous avons choisi MatchZoo pour le grande nombre de modèles de l'état de l'art déjà implémentés pour l'appariement de textes, indépendamment de la tâche ou la collection utilisée. Afin d'évaluer l'appariement symétriques et asymétrique, nous avons considéré les modèles suivants implémentés dans MatchZoo.

– *ARC-II* (Hu *et al.*, 2014) est une architecture pour l'adaptation de texte basée sur un réseau de neurones à convolution.

– *CDSSM* (Shen *et al.*, 2014) fait référence à la version étendue du modèle DSSM (Huang *et al.*, 2013) avec une couche convolutionnelle.

– *DUET* (Mitra *et al.*, 2017) est un modèle d'appariement composé de deux parties parallèles. Il combine l'appariement lexical local et l'appariement sémantique global.

– *MatchPyramid* (Pang *et al.*, 2016) est un modèle d'appariement de textesen adaptant le processus du traitement d'images.

7. https ://github.com/NTMC-Community/MatchZoo/tree/1.0/data/WikiQA

8. https ://www.kaggle.com/c/quora-question-pairs/data

9. https ://github.com/NTMC-Community/MatchZoo/tree/1.0/data/QuoraQP

10. Nous avons utilisé la configuration par défaut proposée par INDRI. Notre objectif n'étant pas d'optimiser les modèles classiques mais est de comparer leurs performances par rapport aux modèles neuronaux.

11. https ://www.lemurproject.org/indri.php

Tableau 1 – Description et configuration des hyper-paramètres de MatchZoo.

Paramètre	Valeur		Description
	WikiQA	QuoraQP	
num_iters	400	500	Époques d'entraînement
test_weights_iters	400	500	Époque de test
text1_maxlen	20	20	Longueur maximale du texte de la première entrée
text2_maxlen	40	20	Longueur maximale du texte de la deuxième entrée
batch_size	100	1024	Taille du batch d'entraînement et de test
losses	rank_hinge_loss	categorical_crossentropy	Fonction d'optimisation de l'erreur d'entraînement

– *MVLSTM* (Wan *et al.*, 2016) est un modèle basé sur la position pour la représentations des phrases.

Nous avons aussi considéré deux autres modèles de l'état de l'art qui ne sont pas implémentés dans MatchZoo.

– $SUM_{BASE;PTK}$ (Tymoshenko et Moschitti, 2018) est un modèle basé sur les Kernels, combinant les similarités intra et croisées des paires de textes.

– $PairwiseRank + SentLevel$ (Rao *et al.*, 2016) est une approche d'apprentissage contraste pour la sélection de réponses et qui utilise des modèles de profonds existants comme composantes de base.

4.2. *Résultats et Analyses*

Dans cette section, nous analysons les résultats expérimentaux de façon à répondre à un ensemble de questions de recherche. Comme suit :

QR1 : Quel est l'apport du type d'architecture, symétrique ou asymétrique, vis-à-vis de la tâche d'appariement de textes de natures différentes ?

Le tableau 2 montre les performances, en termes de MRR, P@1 et P@3, nDCG@1 et nDCG@3, des différents modèles neuronaux évalués dans la collection WikiQA, en utilisant l'architecture originale de chacun des modèles ainsi que les architectures étendues notéespar les labels (Q), (A) et (Q+A) tels décrits dans la section 4.1.1. Les valeurs en caractères **gras** montrent la meilleure performance pour chacune des mesures d'évaluation utilisées. Les symboles ▲ et ▼ représentent respectivement la significativité[12] Nous remarquons que, pour tous les modèles et métriques, au moins une des architectures asymétriques, soit (Q) ou (A), surpasse ses homologues symétriques, notamment (Q+A) et la version originale, les améliorations sont aussi significatives pour le modèle MVLSTM. La prise en compte de la couche d'attention dans (Q) et (A) permet d'obtenir une amélioration importante dans les performances des différents modèles, notamment en termes de MRR et de P@1 où : l'architecture (Q) des modèles ARC-I, MVLSTM et MatchPyramid surpasse l'architecture originale respective à chaque modèle. L'architecture (A) des modèles DUET et CDSSM surpasse considérablement leur architecture originale. Les améliorations restent non significatives pour la majorité des modèles évalués.

Afin d'analyser plus finement les résultats montrés dans le tableau 2, nous voulons vérifier les améliorations au niveau des différentes questions de la collection WikiQA. Le tableau 3 montre les performances en termes de MAP, des différents modèles neu-

12. T-test avec $p = 0.05$.

Tableau 2 – Comparaison des performances, en termes de MRR, nDCG@3, P@1 et P@3, des différents modèles de référence, dans la collection WikiQA, en utilisant les différentes architectures. Les valeurs en caractères **gras** représentent les performances maximales. Les symboles ▲ et ▼ représentent respectivement la significativité des améliorations et détérioration des performances.

Modèle	MRR	P@1	P@3	nDCG@1	nDCG@3
ARC-II	0.5708	0.3840	0.2489	0.3840	0.5410
ARC-II.(Q)	0.5748	0.4177	0.2419	0.4177	0.5357
ARC-II.(A)	0.5528	0.3544	0.2531	0.3544	0.5327
ARC-II.(Q+A)	0.5814	0.3924	0.2503	0.3924	0.5485
CDSSM	0.5586	0.3671	0.2475	0.3671	0.5285
CDSSM.(Q)	0.5222	0.3333	0.2335	0.3333	0.4973
CDSSM.(A)	0.5886	0.4135	0.2461	0.4135	0.5490
CDSSM.(Q+A)	0.5622	0.3924	0.2405	0.3924	0.5266
DUET	0.6259	0.4599	0.2714	0.4599	0.6016
DUET.(Q)	0.6314	0.4641	0.2742	0.4641	0.6116
DUET.(A)	0.6383	0.4852	0.2714	0.4852	0.6112
DUET.(Q+A)	0.5982	0.4262	0.2531▼	0.4262	0.5589▼
MatchPyramid	0.6529	0.4726	0.2869	0.4726	0.6448
MatchPyramid.(Q)	**0.6715**	**0.5063**	**0.2981**	**0.5063**	**0.6649**
MatchPyramid.(A)	0.5575▼	0.3544▼	0.2531▼	0.3544▼	0.5381▼
MatchPyramid.(Q+A)	0.4698▼	0.2574▼	0.2110▼	0.2574▼	0.4211▼
MVLSTM	0.6215	0.4388	0.2813	0.4388	0.6101
MVLSTM.(Q)	0.6691▲	0.5021▲	0.2897	0.5021▲	0.6539▲
MVLSTM.(A)	0.6174	0.4388	0.2714	0.4388	0.5984
MVLSTM.(Q+A)	0.5904	0.4093	0.2517▼	0.4093	0.5562▼

ronaux évalués, en utilisant les différentes architectures correspondantes. En comparaison avec les modèles classiques BM25 et ML. Nous montrons aussi le nombre de questions ayant été Améliorées, ou Détériorées ou qui sont restées Inchangées, A/D/I respectivement, par l'architecture du modèle correspondant en comparaison au modèle original. Nous constatons que la MAP est meilleure dans l'une des architectures asymétriques pour tous les modèles neuronaux, permettant des meilleurs performances par rapport aux modèles classiques. Concernant les statistiques des questions, nous constatons que le nombre de questions améliorées par les architectures (Q) et (A) est plus important qu'avec l'architecture (Q+A). Notez que ces résultats appuient fortement notre hypothèse concernant l'impact des architectures asymétriques sur la tâches d'appariement asymétrique question-réponse. De plus, les performances obtenues avec le modèle asymétrique (Q)-MatchPyramid.ω sont les meilleures pour cet collection[13], tandis que le nombre de requêtes améliorés est plus grand pour le modèle CDSSM avec l'architecture (Q+A). Dans ce tableau, nous remarquons que lorsque les nombres des questions améliorées et détériorées sont très proches, la différence entre les performances du modèle original et son architecture étendue correspondante, notamment (Q), (A) ou (Q+A) n'est pas significative.

13. (Q)-MatchPyramid.ω surpasse également toutes les méthodes rapportées par MatchZoo dans https ://github.com/NTMC-Community/MatchZoo .

Tableau 3 – Comparaison des performances en termes de MAP, de l'appariement *Symétrique* et *Asymétrique* des différents modèles neuronaux de référence en comparaison aux modèles classiques, dans la collection WikiQA. L'extension *".ω"* fait référence à l'application de la couche d'attention ω avec le modèle correspondant, comme décrit dans les exemples de la figure 2. A/D/I représentent le nombre de questions Améliorées, Détériorées et Inchangées respectivement, par le modèle correspondant avec l'architecture respective, en comparaison à son architecture (Original). Les symboles ▲ et ▼ représentent respectivement la significativité des améliorations et détérioration des performances.

Classe	Modèle			MAP	A/D/I
Modèles classiques	BM25			0.5762	-/-/-
	ML			0.5932	-/-/-
Modèles Neuronaux	Symétrique	Original	ARC-II	0.5606	-/-/-
			CDSSM	0.5473	-/-/-
			DUET	0.6113	-/-/-
			MatchPyramid	0.6436	-/-/-
			MVLSTM	0.6046	-/-/-
		(Q+A)	ARC-II.ω	0.5648	66/66/105
			CDSSM.ω	0.5523	85/76/76
			DUET.ω	0.5801▼	64/76/97
			MatchPyramid.ω	0.4697▼	56/129/52
			MVLSTM.ω	0.5562	64/83/90
	Asymétrique	(Q)	ARC-II.ω	0.5595	42/72/123
			CDSSM.ω	0.5134	76/91/70
			DUET.ω	0.6158	69/58/110
			MatchPyramid.ω	**0.6591**	47/42/148
			MVLSTM.ω	0.6507▲	70/41/126
		(A)	ARC-II.ω	0.5439	60/81/96
			CDSSM.ω	0.5779	78/75/84
			DUET.ω	0.6251	68/53/116
			MatchPyramid.ω	0.5502▼	43/97/97
			MVLSTM.ω	0.6165	68/67/102

QR2 : Quel est l'impact de la couche d'attention sur le comportement des différents modèles neuronaux ?

Pour répondre à cette question, nous analysons tout d'abord le comportement des différents modèles neuronaux, sans et avec la couche d'attention, par rapport au tri des résultats d'une question. Par la suite, nous analysons comment cette couche agit sur le traitement de chaque mot de la question ou de la réponse correspondante.
Considérons la question *Q* suivante extraite de la collection WikiQA. Dans cette collection, il y a 10 réponses candidates, dont 4 seulement sont pertinentes.
***Q** : "What happened to George O'malley on grey's anatomy ?"*
Dans la figure 3, nous représentons la tâche de tri des 10 réponses candidates, par une liste de 10 carrés. Chaque carré représente l'une des réponses. Les réponses pertinentes sont représentées par des carrés opaques. L'objectif est de pousser les réponses pertinentes vers la gauche de la liste afin de les retourner en premier, ce qui correspond au résultat "Ideal rank" de la figure 3. Dans cette figure, nous comparons chacune des architectures asymétriques par rapport au modèle original correspondant. Nous remarquons que les architectures où, la couche d'attention appliquée pour la question (Q), dans les modèles MatchPyramid et MVLSTM, et pour la réponse (A) dans les modèles CDSSM, ARC-II et DUET, permet de pousser les résultats pertinents vers la gauche

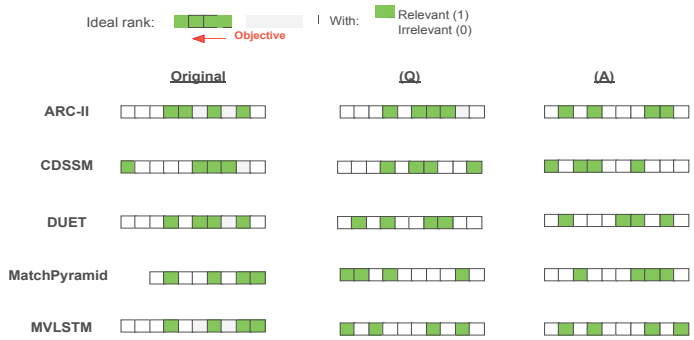
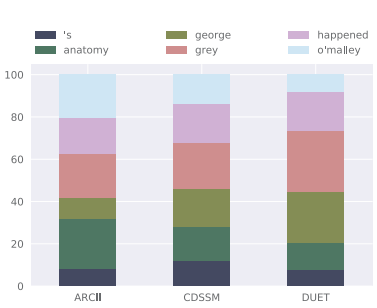


Figure 3 – Tri des différentes réponses par chacun des modèles, sans et avec la couche d'attention. L'objectif est d'avoir toutes les réponses pertinentes poussées vers la gauche de la liste.

de la liste, résultant en un meilleur tri par rapport aux modèles originaux.
Nous reprenons la question *Q* précédente avec l'une des réponses pertinentes *A* qui est retournée par tous les modèles neuronaux du tableau 3. Pour *Q* et *A*, nous donnons la liste correspondante des mots-clés indexés par MatchZoo :
Q *: "What happened to George O'malley on grey's anatomy ?"*
Key words *: happened, george, o'malley, grey, 's, anatomy.*
A *: "In 2007, Knight's co-star Isaiah Washington (Preston Burke) insulted him with a homophobic slur, which resulted in the termination of Washington's Grey's Anatomy contract."*
Key words *: knight, 's (1), isaiah washington (1), preston, burke, resulted, termination, washingthon (2), 's (2), grey, 's (3), anatomy, contact.*
Pour chacun des modèles neuronaux évalués, nous retenons l'architecture asymétrique avec laquelle le modèle a donné de meilleurs résultats dans les analyses précédentes, puis nous analysons les poids d'importance calculés dans chaque modèle, pour les mots clés de la question ou de la réponse. La figure 4 montre les différentes valeurs calculées. L'objectif est d'analyser l'impact de la couche d'attention au niveau de chaque mot selon son importance réelle dans la question ou la réponse.
Dans la figure 4a, les mots clés *happened* et *george* sont majoritairement pondérés dans la question *Q* par les trois modèles représentés, ce qui est cohérent avec l'objet de *Q*. En effet, la question vise à savoir ce qui est arrivé à *george*. Dans la figure 4b, les mots clés *contract*, *resulted* et *termination* ont acquis des poids majeurs. Ces mots représentent le noyau de l'information apportée par la réponse *A*. Ces résultats montrent l'intérêt de l'utilisation de la couche d'attention dans le processus de pondération des mots de la question/réponse selon leur importance informative.

QR3 : Quel est l'impact de l'architecture symétrique dans l'appariement de textes de même nature ?

La figure 5 montre les performances des différents modèles neuronaux, avec les différentes architectures, en terme d'exactitude (*Accuracy*) dans la tâche d'appariement des paires de questions dans la collection QuoraQP. Nous constatons que la majorité des modèles ne bénéficient pas de l'architecture asymétrique comme prévu, étant donné l'aspect symétrique de cette tâche. À l'exception du modèle CDSSM où les ar-



(a) Poids des mots clés de la question *Q* calculés par la couche d'attention de chacun des modèles avec l'architecture asymétrique (Q).

(b) Poids des mots clés de la réponse *A* calculés par la couche d'attention de chacun des modèles avec l'architecture asymétrique (A).

Figure 4 – Poids d'importance calculés par la couche d'attention dans les différents modèles dans l'architecture asymétrique correspondante.

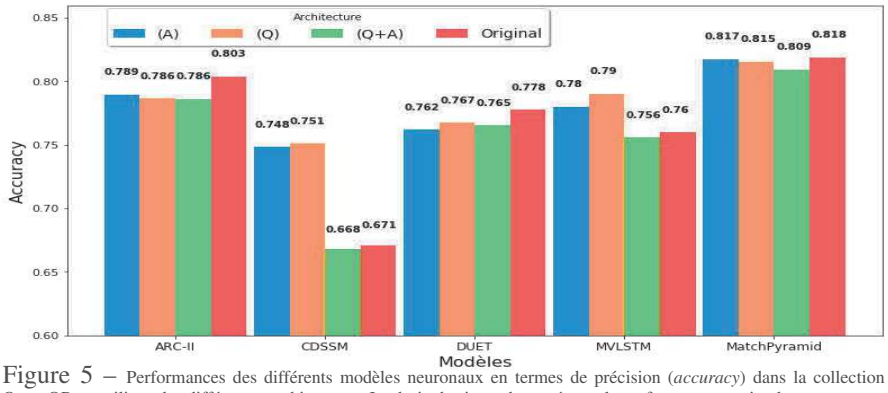


Figure 5 – Performances des différents modèles neuronaux en termes de précision (*accuracy*) dans la collection QuoraQP, en utilisant les différentes architectures. La droite horizontale représente la performance maximale.

chitectures asymétriques (Q) et (A) surpassent considérablement le modèle original. En effet, le modèle CDSSM utilise une couche à convolution permettant de prendre en compte la notion du contexte (Shen *et al.*, 2014) et les poids d'attention calculés exclusivement pour l'une des entrées à la fois permettent d'informer le modèle sur les mots les plus importants de chaque contexte, ainsi d'améliorer ses performances. De même pour le modèle MVLSTM, où les mots sont considérés dépendamment de leurs positions (Wan *et al.*, 2016), les poids d'attention permettent d'informer le modèle sur l'importance du mot à chaque position. Par ailleurs, pour le reste des modèles, l'architecture asymétrique n'a pas eu d'apport important sur leurs résultats.

À cet effet, nous voulons analyser le comportement de l'appariement asymétrique dans une tâche symétrique. Considérons les questions suivantes de la collection QuoraQP :

Q1 : "Are we living in a simulation ?"

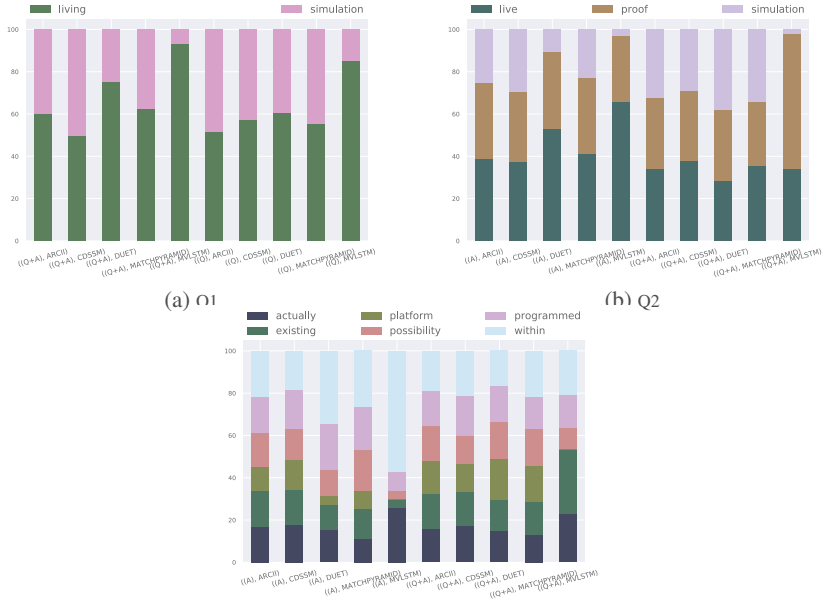
Q2 : "Do we have any proof that we live in a simulation ?"

Q3 : "Is there a possibility that we actually are existing in a programmed platform ?"

Q1 est similaire à *Q2* (Label = 1), mais pas similaire à *Q3* (Label = 0). Dans le tableau 4 nous présentons la variation des différents scores de similarité calculés par les différents modèles par rapport à chacune des architectures. Dans le tableau 4, les cellules

Tableau 4 – Les scores de similarité attribués à une paire de questions similaires (*Q1*, *Q2*) et une paire de questions non-similaires (*Q1*, *Q3*) par les différents modèles neuronaux avec les architectures symétriques et asymétriques.

Paire de question		Label	Architecture	Score de similarité prédit				
				ARCI-II	CDSSM	DUET	MatchPyramid	MVLSTM
Q1	Q2	1	Original	0.0464	0.3694	0.7067	0.1002	0.5658
	Q3	0		0.9641	0.3694	0.1416	0.0131	0.1447
	Q2	1	(Q+A)	0.3621	0.2092	0.6211	0.9084	0.6783
	Q3	0		0.3464	0.2917	0.0840	0.0456	0.0150
	Q2	1	(Q)	0.2638	0.7184	0.6946	0.7736	0.7060
	Q3	0		0.0024	0.2727	0.1576	0.1612	0.0162
	Q2	1	(A)	0.1530	0.6922	0.8057	0.6116	0.6600
	Q3	0		0.0462	0.2208	0.4703	0.1055	0.2225



(a) Q1 (b) Q2

(c) Q3

Figure 6 – Poids d'importance calculés par la couche d'attention dans les différents modèles avec différentes architectures, pour les mots de 3 questions de la collection QuoraQP.

opaques font référence à l'optimisation du score de similarité correspondant à la paire correcte (*Q1*, *Q2*) avec l'architecture correspondante par rapport à l'architecture originale du modèle. Nous remarquons que le score de la question *Q2* est optimisé dans les architectures asymétriques (Q) et (A) aussi bien que dans l'architecture symétrique (Q+A). En particulier, concernant les modèles ARC-II et CDSSM, l'architecture originale attribuent des scores inappropriés aux différentes paires (*Q1*, *Q2*) et (*Q1*, *Q3*), la couche d'attention a permis de calculer des scores de similarité qui séparent correctement les deux paires. Pour vérifier ces résultats, nous avons calculé les poids d'attention attribués par la couche ω appliquée pour *Q1* dans les architectures (Q) et (Q+A) et pour les deux autres question, *Q2* et *Q3*, dans les architectures (A) et (Q+A). La figure 6 montre les poids d'attention attribués pour chacun des termes de ces questions, par rapport aux différents modèles. Dans la figure 6, nous remarquons que pour chaque modèle, que ce soit avec une architecture symétrique (Q+A) ou asymétrique

Tableau 5 – Comparaison des résultats de notre approche, utilisant des modèles *Internes* à MatchZoo, par rapport aux modèles *Externes*. Les valeurs en caractères **gras** montrent les meilleurs performances pour chacune des mesures.

Modèles		MAP	MRR
Internes (nos modèles)	Asymétrique-ARC-II$_{(Q)}$	0.5595	0.5748
	Asymétrique-CDSSM$_{(A)}$	0.5779	0.5886
	Asymétrique-DUET$_{(A)}$	0.6251	0.6383
	Asymétrique-MatchPyramid$_{(Q)}$	0.6591	0.6715
	Asymétrique-MVLSTM$_{(Q)}$	0.6507	0.6691
Externes	$SUM_{BASE;PTK}$ (Tymoshenko et Moschitti, 2018)	**0.7559**	**0.7700**
	$PairwiseRank + SentLevel$ (Rao *et al.*, 2016)	0.7010	0.7180

(Q) et (A), la couche d'attention a le même impact sur le calcul des poids d'importance des différents mots des trois questions *Q1*, *Q2* et *Q3*. De plus, pour un modèle qui calcule des scores de similarité erronés, notamment ARC-II et CDSSM, la couche d'attention permet au modèle de se focaliser sur les mots clés de l'entrée.

QR4 : Quelle est l'efficacité de notre approche par rapport aux modèles de l'état de l'art qui ne sont pas implémentés dans MatchZoo ?

Le tableau 5 montre les performances de notre approche en termes de MAP et MRR. Les modèles dits *Internes* sont les modèles implémentés dans MatchZoo, pour chaque modèle, nous considérons l'architecture asymétrique ayant donné les meilleurs résultats, (Q) ou (A). Les modèles dits *Externes* représentent des modèles de l'état de l'art qui ne sont pas implémentés dans MatchZoo. Nous remarquons que le modèle externe $SUM_{BASE;PTK}$ donne de meilleurs résultats par rapport aux différents autres modèles. En outre, le modèles $PairwiseRank + SentLevel$ donne de meilleurs résultats par rapport au différents modèles *Internes* avec les architectures correspondantes. Le travail à venir, va porter sur l'évaluation de notre approche asymétrique avec ces modèles afin d'obtenir de nouveaux résultats de l'état de l'art. De plus, l'intégration de ces modèles dans MatchZoo, nous permettra de comparer les résultats dans le même cadre expérimental.

5. Conclusion

Dans cet article, nous avons proposé une approche d'appariement permettant de prendre en compte l'aspect asymétrique de certaines tâches d'appariement de texte. Nous avons étendu plusieurs modèles de l'état de l'art, avec des couches d'attention pour construire les architectures asymétriques correspondantes. Les expérimentations, dans deux différentes collections du domaine du question-réponse, ont montré des résultats prometteurs de la prise en compte de l'asymétrie de certaines tâches d'appariement par la couche d'attention. Dans le cas d'une tâche symétriques, nous avons vu que l'architecture asymétrique n'a pas eu d'impact considérable sur les performance de certain modèles. Cependant, la couche d'attention a permis de corriger les scores de similarité calculés par les modèles basés sur des réseaux MLP comme le modèle CDSSM et sur la position des mots comme le modèle MVLSTM, où les mots ne sont pas considérés par rapport à leur importance. Le travail à venir va concentrer notre réflection sur un modèle qui tient compte de l'aspect symétrique/asymétrique de la tâche abordée, dans la même architecture, ainsi que l'évaluation de notre approche dans d'autres collections du domaine du question-réponse.

6. Bibliographie

Abishek K., Hariharan B. R., Valliyammai C., « An Enhanced Deep Learning Model for Duplicate Question Pairs Recognition », *Soft Computing in Data Analytics*, Springer, p. 769-777, 2019.

Bahdanau D., Cho K., Bengio Y., « Neural machine translation by jointly learning to align and translate », *arXiv preprint arXiv :1409.0473*, 2014.

Bordes A., Weston J., Collobert R., Bengio Y., « Learning Structured Embeddings of Knowledge Bases », *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, AAAI Press, p. 301-306, 2011.

Bromley J., Guyon I., LeCun Y., Säckinger E., Shah R., « Signature verification using a" siamese" time delay neural network », *Advances in neural information processing systems*, p. 737-744, 1994.

Guo J., Fan Y., Ai Q., Croft W. B., « A deep relevance matching model for ad-hoc retrieval », *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, ACM, p. 55-64, 2016.

Hu B., Lu Z., Li H., Chen Q., « Convolutional Neural Network Architectures for Matching Natural Language Sentences », *in* Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (eds), *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., p. 2042-2050, 2014.

Huang P.-S., He X., Gao J., Deng L., Acero A., Heck L., « Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data », *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, ACM, New York, NY, USA, p. 2333-2338, 2013.

Jelinek F., « Interpolated estimation of Markov source parameters from sparse data », *Proc. Workshop on Pattern Recognition in Practice, 1980*, 1980.

Kim Y., « Convolutional Neural Networks for Sentence Classification », p. 1746-1751, 2014.

Liu B., Zhang T., Niu D., Lin J., Lai K., Xu Y., « Matching Long Text Documents via Graph Convolutional Networks », *arXiv preprint arXiv :1802.07459*, 2018.

Mitra B., Diaz F., Craswell N., « Learning to Match Using Local and Distributed Representations of Text for Web Search », *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, p. 1291-1299, 2017.

Nicosia M., Moschitti A., « Accurate Sentence Matching with Hybrid Siamese Networks », *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, ACM, New York, NY, USA, p. 2235-2238, 2017.

Pang L., Lan Y., Guo J., Xu J., Wan S., Cheng X., « Text Matching as Image Recognition. », *AAAI*, p. 2793-2799, 2016.

Pang L., Lan Y., Guo J., Xu J., Xu J., Cheng X., « DeepRank : A New Deep Architecture for Relevance Ranking in Information Retrieval », *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, ACM, New York, NY, USA, p. 257-266, 2017.

Parikh A., Täckström O., Das D., Uszkoreit J., « A Decomposable Attention Model for Natural Language Inference », p. 2249-2255, 2016.

Peng Y., Liu B., « Attention-based Neural Network for Short-text Question Answering », *Proceedings of the 2018 2Nd International Conference on Deep Learning Technologies*, ICDLT '18, ACM, New York, NY, USA, p. 21-26, 2018.

Rao J., He H., Lin J., « Noise-Contrastive Estimation for Answer Selection with Deep Neural Networks », *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, ACM, New York, NY, USA, p. 1913-1916, 2016.

Robertson S. E., Walker S., « Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval », *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, Springer-Verlag New York, Inc., p. 232-241, 1994.

Shen Y., He X., Gao J., Deng L., Mesnil G., « Learning Semantic Representations Using Convolutional Neural Networks for Web Search », *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, ACM, New York, NY, USA, p. 373-374, 2014.

Tomar G. S., Duque T., Täckström O., Uszkoreit J., Das D., « Neural Paraphrase Identification of Questions with Noisy Pretraining », p. 142-147, 2017.

Tymoshenko K., Moschitti A., « Cross-Pair Text Representations for Answer Sentence Selection », *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 2162-2173, 2018.

Wan S., Lan Y., Guo J., Xu J., Pang L., Cheng X., « A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. », *AAAI*, vol. 16, p. 2835-2841, 2016.

Yang L., Ai Q., Guo J., Croft W. B., « aNMM : Ranking short answer texts with attention-based neural matching model », *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ACM, p. 287-296, 2016a.

Yang Y., Yih W.-t., Meek C., « WikiQA : A Challenge Dataset for Open-Domain Question Answering », *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Lisbon, Portugal, September, 2015.

Yang Z., Yang D., Dyer C., He X., Smola A., Hovy E., « Hierarchical attention networks for document classification », *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1480-1489, 2016b.

Yin W., Schütze H., « Convolutional neural network for paraphrase identification », *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 901-911, 2015.

Yin W., Schütze H., Xiang B., Zhou B., « ABCNN : Attention-Based Convolutional Neural Network for Modeling Sentence Pairs », *Transactions of the Association of Computational Linguistics*, vol. 4, nº 1, p. 259-272, 2016.

Zweig G., Platt J. C., Meek C., Burges C. J. C., Yessenalina A., Liu Q., « Computational Approaches to Sentence Completion », *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers - Volume 1*, ACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 601-610, 2012.