évitez les accents dans les noms de fichiers !

MULTILINGUISME

Projet

15

Evaluation de la traduction

Charlotte Dugrain

Alexandra Guérin

Année 2019-2020
M2 - Linguistique Informatique

Introduction :

La pertinence de la traduction automatique dépend de sa rapidité et de ses performances. Il est évident qu'elle est toujours moins digne de confiance que celle effectuée par un traducteur humain, cependant, dans certaines situations, comme par exemple lorsque la masse de texte à traduire est très importante ou que les délais sont serrés, il peut être judicieux d'utiliser ces systèmes et d'accepter les erreurs de traduction qu'elle peuvent engendrer. Tout est une question d'équilibre entre qualité de la traduction, temps passé à traduire et coût de la traduction. Le temps passé et le coût sont facilement mesurables, mais qu'en est-il de la qualité de la traduction ?

La solution la plus fiable pour le savoir, consiste à demander à des personnes bilingues d'évaluer les traductions réalisées, on peut alors déterminer un score et chiffrer la qualité de la traduction. Cependant, avec cette méthode, on tombe dans un paradoxe puisque le temps et l'argent gagné à utiliser des systèmes automatiques pour l'étape de traduction est de nouveau perdu lors de son évaluation. Il faut donc trouver un système capable d'évaluer la traduction de façon automatique.

Beaucoup de solutions existent déjà pour scorer une traduction, on peut utiliser des modèles de langue, des probabilités d'alignement (IBM-1), des métriques de similarité (score bleu, meteor..) ou encore réaliser des calculs à l'aide d'embeddings.

Dans ce projet, nous allons examiner ces différentes solutions et regarder dans quelle mesure elles sont ou ne sont pas pertinentes pour réaliser la tâche qui nous intéresse ici, et en fonction de ces résultats, nous proposerons une métrique qui nous semble la plus efficace.

Présentation de la tâche

La tâche à réaliser

La tâche que nous voulons réaliser ici est conceptuellement assez simple : mettre au point un système qui permet de donner un score, sur une échelle de 0 à 1, à une phrase dans une langue B candidate pour la traduction d'une autre phrase dans une langue A, en ayant la possibilité de se servir d'une traduction de référence dans la langue cible .

Il faut que ce score prenne en compte à la fois l'aspect syntaxique, c'est à dire que la phrase doit être correctement construite (fluency) , et l'aspect lexical, c'est à dire que le sens de la phrase source doit être conservé dans la traduction (adequacy). Globalement, ces deux aspects sont la plupart du temps étroitement liés et il n'est pas très pertinent de les différencier au cours du calcul.

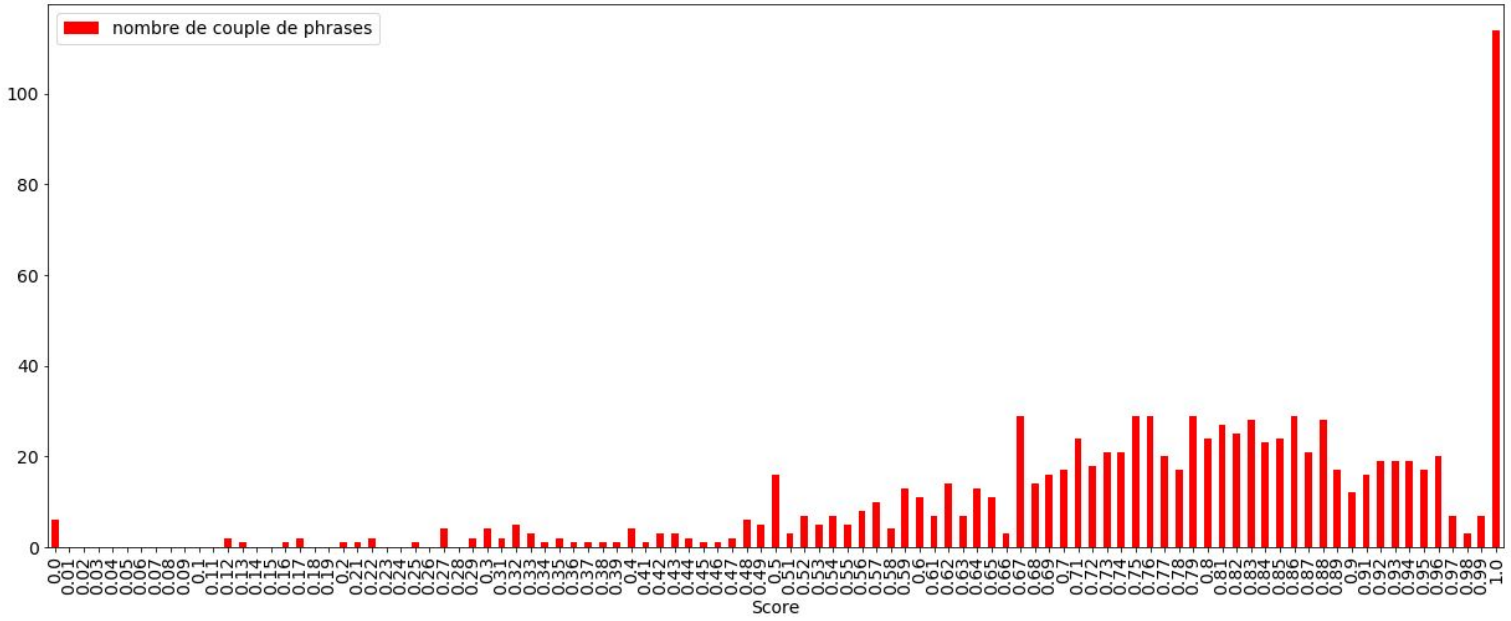
La principale difficulté qui se pose à nous est dans un premier temps d'explorer les pistes qui existent déjà pour déterminer dans quelle mesure il est pertinent ou non de les utiliser. Il faudra ensuite élaborer une métrique personnalisée permettant d'attribuer des scores qui se rapprochent le plus possible des scores humains donnés.

Les données disponibles

Les données dont nous disposons pour effectuer cette tâche sont : une liste de 999 phrases sources, avec leur hypothèse de traduction et leur traduction de référence, accompagnées par un score qui a été donné manuellement , qui peut être considéré comme un score de référence duquel se rapprocher avec notre métrique.

Il est toujours intéressant d'explorer les données avant de les manipuler. Nous allons particulièrement nous concentrer sur le score puisque c'est dans notre cas la valeur que notre métrique devra refléter. Nous allons donc étudier la répartition des scores sur nos 999 phrases, en arrondissant les scores au centième et observer le nombre de couple de phrases par score.

est-ce la représentation graphique la plus adaptée --> votre score est continu, il me parait plus judicieux d'utiliser un « density plot » (cf la fonction distplot de Seaborn)



Nous calculons également la moyenne et la médiane qui sont respectivement de 0,768 et de 0,795. De plus, 75% des scores sont supérieurs à 0,681. Ces indicateurs ainsi que le graphe les représentant nous montre donc que les bons scores sont extrêmement majoritaires dans ce corpus, preuve à priori d'une certaine qualité des traductions fournies.

Les solutions existantes

commencez toujours les sections par une mini-intro pour expliquer / rappeler au lecteur où vous en êtes dans votre exposé

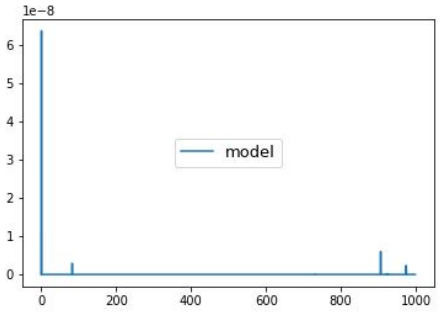
Modèle de langue

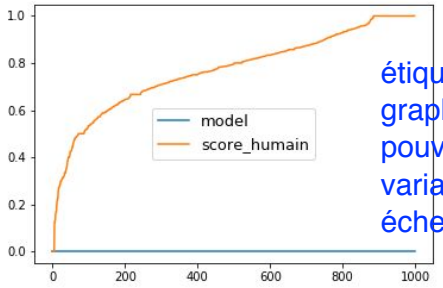
Un modèle de langue permet d'estimer la probabilité d'une séquence de mots dans une langue donnée. Cette probabilité nous permet donc d'estimer la vraisemblance de l'apparition d'une phrase dans la langue considérée. Il paraît alors pertinent de prendre cet aspect en compte lors de l'attribution d'un score pour une traduction.

Dans notre cas nous souhaitons attribuer à chacune de nos hypothèses de traduction une probabilité. En effet nous aimerions estimer si la syntaxe a bien été adaptée au français, qui est dans ce cas la langue cible de notre traduction.

Nous allons entraîner notre modèle à l'aide de KenLM qui utilise le lissage de Kneser–Ney. Nous choisirons d'obtenir un modèle 5-grams. Afin d'entraîner correctement un modèle de langue il est nécessaire d'avoir un corpus conséquent dans la langue ciblée, nous allons donc utiliser le corpus Europarl, corpus extrait des sessions du parlement européen. Le fichier utilisé est composé de 2 007 723 phrases en français.

L'entraînement via KenLM nous permet d'obtenir un fichier arpa, que l'on transformera ensuite en fichier binary afin d'améliorer la vitesse d'exécution. Une fois ce modèle chargé nous pouvons attribuer les scores voulus à chacune de nos hypothèses.





probabilités données par le modèle de langue

probabilités données par le modèle de langue à l'échelle des scores humains

étiquetez vos axes —> là votre graphe ne sert à rien + vous ne pouvez pas représenter des variables n'utilisant pas la même échelle sur un même axe !

est-ce qu'un critère quantitatif (typiquement corrélation) ne serait pas préférable pour une conclusion aussi forte ? —> ici une corrélation de Spearman peut même prendre en compte les différences d'échelles entre les deux valeurs que vous comparez

le soulignement est une aberration typographique

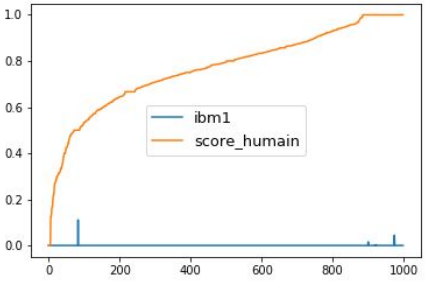
On voit donc que nos résultats ne sont pas du tout concluants tous très proches de zéro et n'observant pas une quelconque croissance bien que les exemples aient été rangés par score humain croissant.

Outre nos résultats principal problème de cette métrique est dû au fait que, comme nous l'avons expliqué précédemment, elle se base uniquement sur les hypothèses de traduction, et n'est à aucun moment mise en relation ou comparée à la source ou la référence. On mesure donc ici uniquement la fluency et pas du tout l'adequacy de la traduction. L'hypothèse pourrait avoir une probabilité de 1 mais être un total contresens de la source et la référence.

Cette probabilité bien que pouvant faire partie des indicateurs de la qualité d'une traduction n'est donc absolument pas suffisante à elle seule, elle doit forcément, si elle est utilisée, être associée à une mesure mettant l'hypothèse en liant avec la source ou la référence.

IBM1

L'algorithme d'IBM1 est un système qui assigne une probabilité d'alignement entre une hypothèse de traduction et une phrase source. Ce n'est pas le type de données d'entrée sur lequel nous souhaitons nous baser mais cela pourrait potentiellement constituer une source d'information supplémentaire à prendre en compte.



probabilités d'alignement ibm1 par rapport au score humain

On peut voir que les probabilités d'alignement ne correspondent pas du tout aux scores qui ont été donnés manuellement. Ils sont d'une part beaucoup plus petits, mais cela aurait pourrait être une question d'ordre de grandeur.
Cependant, quand on classe les phrases selon leur score manuel, on peut voir que cela ne correspond pas du tout aux probabilités d'alignement IBM-1.

sur quel corpus estimez-vous vos données ?

Cela vient sûrement du fait que cette probabilité est utilisée dans IBM-1 pour comparer des scores de phrases “hypothèses” entre elles et ainsi prendre la meilleure. Donc, pour comparer des phrases qui ont approximativement le même nombre de mot et des mots semblables à quelques petits détails près. Ainsi on compare des probabilités “comparables”, du même ordre de grandeur.

Ici on voudrait l’utiliser comme un score absolu quelque soit la longueur de la phrase mais forcément si on multiplie des probas entre elles, le score va être dépendant de la longueur de la phrase et il ne peut donc pas être utilisé comme métrique pour donner un score absolu à une traduction.

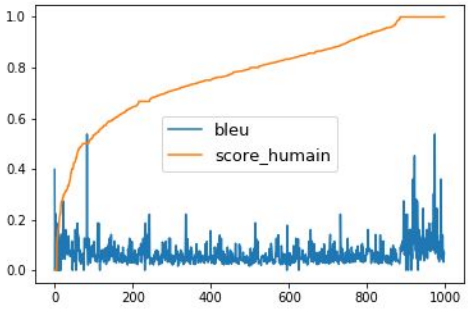
Scores bleu et meteor

Il existe déjà des métriques de similarité utilisées pour donner des scores aux traductions, notamment le score BLEU et le score METEOR. Ils permettent de monitorer l’évolution d’un système de traduction ou encore de comparer l’efficacité de plusieurs systèmes de traduction différents.

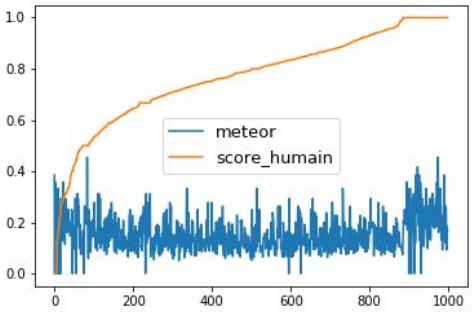
Au premier abord, ces scores ne paraissent pas très pertinents pour la tâche que nous voulons réaliser. En effet, le principal problème du score bleu est qu’il est fait pour être calculé sur un corpus entier et non sur une phrase individuelle, ce qui n’est pas en adéquation avec la tâche que nous voulons réaliser. Pour ce qui est du score meteor, il est difficile de vouloir se baser sur un score qui nécessite des ressources externes comme une table de paraphrases par exemple, car celle ci n’est pas toujours disponible.

on peut utiliser cette métrique uniquement avec des matchs exacts

Cependant, nous pouvons tout de même calculer ces scores pour voir s’ils peuvent être pertinents pour notre tâche. Voici leur répartition par rapport aux scores humains :



score bleu des hypothèses de traduction par rapport au score humain



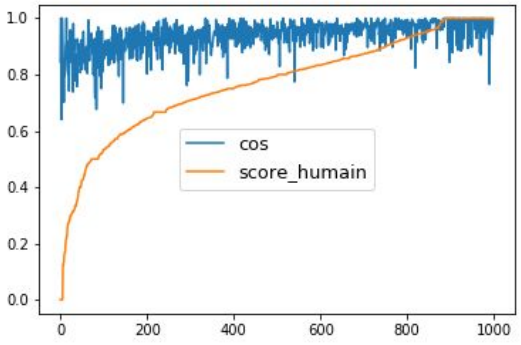
scores meteor des hypothèses de traduction par rapport aux score humains

On peut voir que ces scores ne correspondent pas du tout aux scores qui sont donnés manuellement aux traductions. Ils ne sont déjà pas du même ordre de grandeur et, de plus, ils ne semblent même pas qu'ils aient quelque cohérence avec les scores manuels.

Embeddings pré-entrainés

Une autre piste à explorer afin de comparer nos hypothèses et nos références est d'utiliser des embeddings pour les représenter. Nous allons plus précisément utiliser des embeddings contextualisés pré-entrainés du modèle camemBERT. CamemBERT est un midèle de langue pour le français préentrainé sur le sous-corpus OSCAR ~~du français~~, corpus de 46 milliards de mots. Nous décidons de représenter la phrase comme la moyenne des embeddings de ses tokens, afin d'obtenir des vecteurs de même taille facilitant la comparaison entre phrase. Nous choisissons d'utiliser l'avant dernière couche pour récupérer ces embeddings. Une fois les vecteurs des deux phrases récupérés nous comparons les phrases en calculant la cosinus similarité entre les 2 vecteurs. Nous obtenons de cette manière une corrélation de Spearman de 60% avec le score humain.

pourquoi changer de méthode d'évaluation tout d'un coup ?



cosinus similarité entre les embeddings de l'hypothèse et de la référence par rapport au score humain

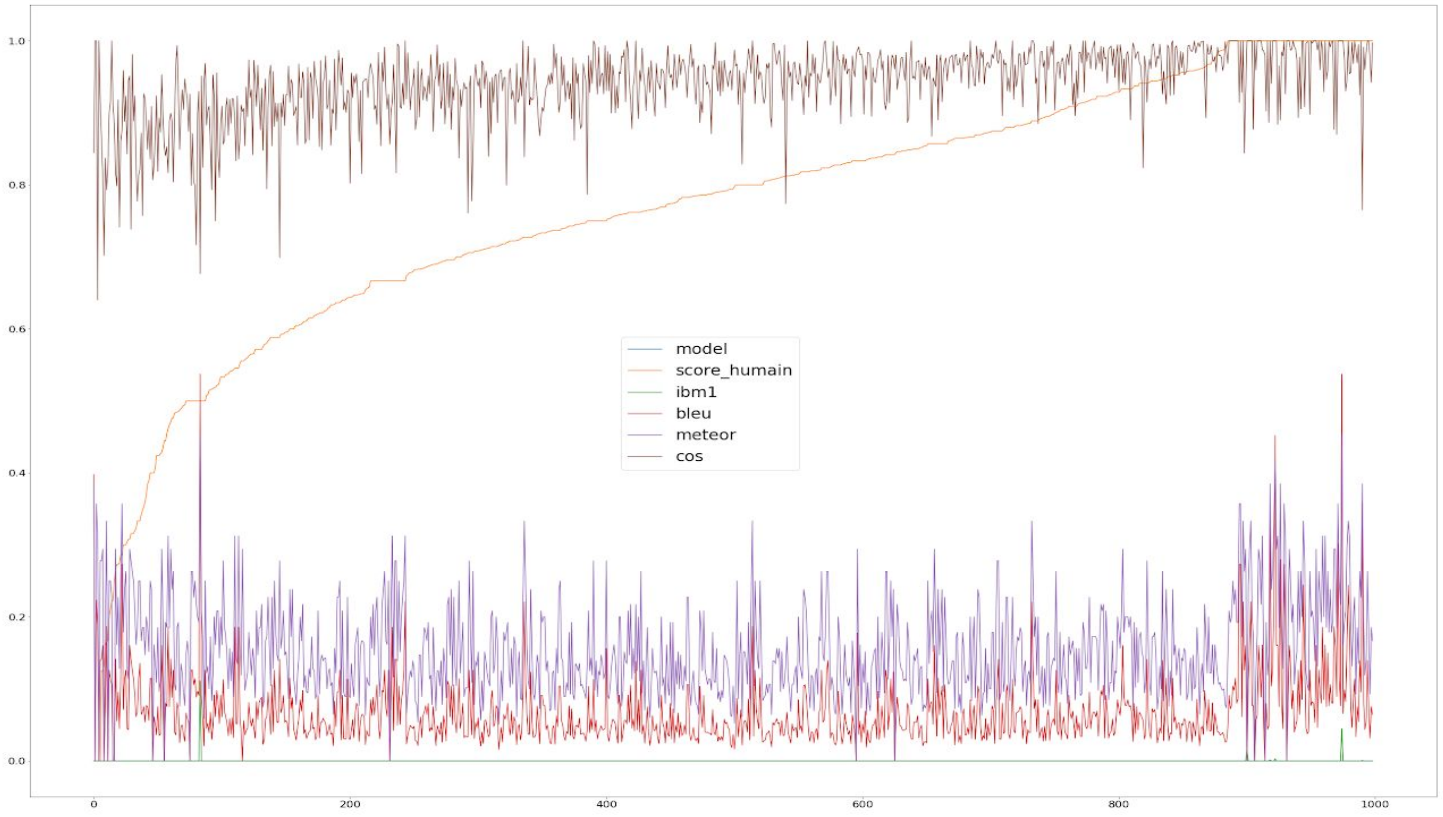
Nous aurions peut être pu plutôt utiliser les embeddings des tokens composant la phrase en ne les moyennant pas directement mais camemBERT utilisant un tokenizer Wordpiece il n'est pas toujours évident d'accéder aux tokens exacts de la phrase. Et cela aurait également demandé une phase d'alignement pour comparer les tokens correspondant d'une phrase à l'autre, ce qui aurait pu fausser les résultats.

et pourquoi ne pas travailler uniquement sur des données tokenisées ?

La solution proposée et sa démarche d'élaboration

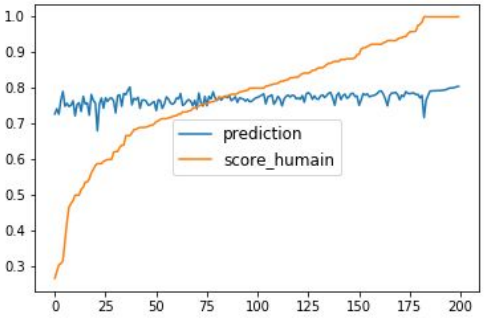
Fonctionnement

Il est évident aux vues des précédents résultats qu'aucune mesure existante ne peut suffire à exprimer le score que nous voulons déterminer à elle seule. Le graphique nous aide bien à nous rendre compte à quel point ils sont décorrélés des scores humains.



représentation graphique des différentes métriques en fonction du score humain

Cependant, peut être que la combinaison de ces multiples informations peut être significative et permettre à un système d'apprentissage, en l'occurrence une régression, de prédire un score. Nous allons donc mettre tous ces scores les uns à la suite des autres dans un vecteur et utiliser cela comme une donnée d'apprentissage.



scores prédits par le système en prenant en compte le métriques de base, model de langue, ibm1, bleu, meteor et cosinus similarité d'embeddings

Par cette méthode, on obtient une corrélation de 0,61 mais graphiquement on voit clairement que cette prédiction n'est pas de bonne qualité.

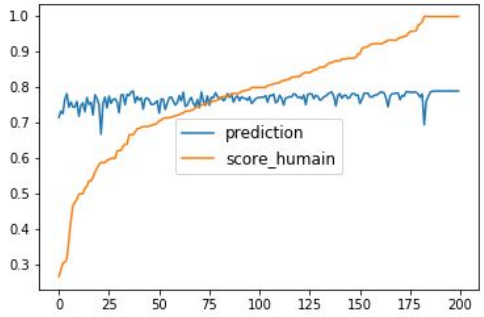
pourquoi ? comment l'interprétez vous ?

Le paramètre du vecteur qui semble être le plus utile dans cette prédiction est la similarité cosinus des embeddings. En effet, si l'on enlève du vecteur de données les autres scores, la corrélation reste autour de 0,6 mais si on enlève la similarité cosinus, la corrélation chute

brusquement à 0.09. Cela était assez facilement déductible en raison de la forte corrélation qu'elle présente déjà à elle seule avec les scores humains.

Nous décidons donc de continuer sur cette voie en gardant dans le vecteur de données d'apprentissage uniquement la similarité cosinus, mais en ajoutant d'autres caractéristiques.

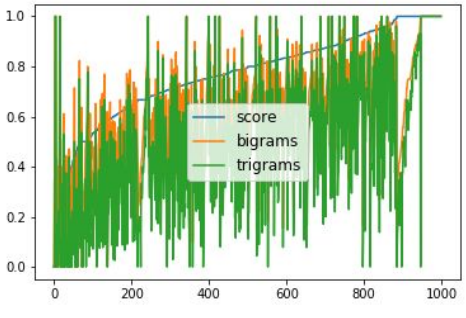
1. Mesure de la différence de longueur de l'hypothèse et de la référence



scores prédits par le système en prenant en compte la similarité cosinus et la différence de longueur des phrases

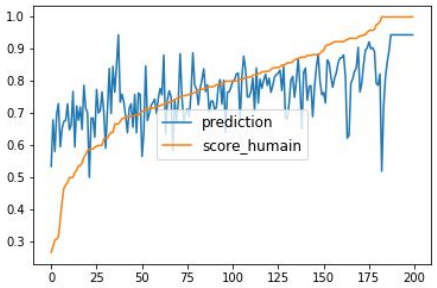
La première caractéristique que nous rajoutons est un critère basique mais assez représentatif, la différence de longueur entre l'hypothèse et la référence. Nous ajoutons cette caractéristique au cosinus dans le vecteur de donnée d'entrée et les résultats ne sont pas beaucoup plus concluant. La corrélation reste autour de 0,61 ce qui ne change pas grand chose.

2. Proportion de bigrams et/ou de trigrams communs



rapport entre les bigrams et trigrams communs avec le score humain

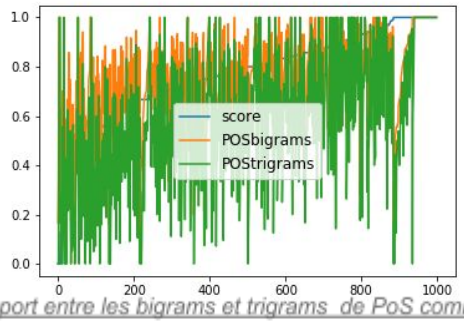
La seconde caractéristique que nous voulons prendre en compte est la proportion de bigrams communs entre l'hypothèse et la traduction ou de trigrams communs. On peut voir une tendance qui se dessine un peu entre le score humain et le nombre de bigrams communs. Mais la correspondance n'est toujours pas évidente



scores prédits par le système en prenant en compte le cosinus et la différence de longueur et les bigrams et trigrams communs

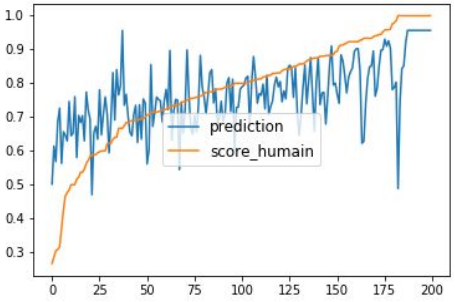
Finalement la corrélation est meilleure avec ces bigrams et trigrams communs (0,64) et on peut commencer à voir un semblant de tendance similaire à la courbe des scores qui se dessine.

3. Proportion de bigrams et de trigrams de PoS communs à la référence.



rapport entre les bigrams et trigrams de PoS communs avec le score humain

Afin d'essayer de prendre l'aspect syntaxique en compte nous décidons d'ajouter la proportion de bigrams ou trigrams communs entre l'hypothèse et la traduction mais cette fois au niveau de leurs parties du discours. Nous essayons en effet de comparer la similarité entre la structure des 2 phrases.



scores prédits par le système en prenant en compte la cosinus similarité et la différence de longueur des phrases, les bigrams, trigrams de mots et de pos communs

La corrélation (0,64) ainsi que la courbe reste très similaires comparées à notre tentative précédente. Cette solution ne semble pas être révolutionnaire.

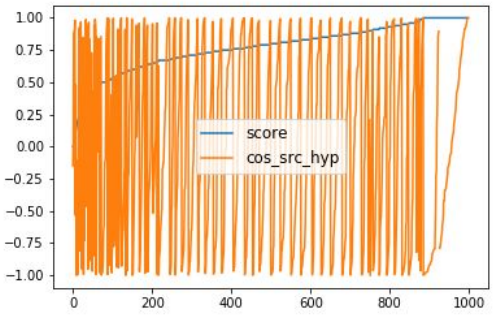
conclusion un peu forte —> la corrélation est loin d'être mauvaise ! Plutôt qu'un jugement aussi catégorique il peut être plus intéressant de s'interroger sur les raisons qui expliquent pourquoi une caractéristique aussi simple est autant corrélées.

4. Comparaison des embeddings de la source et de l'hypothèse.

La dimension que nous ne semblons pas arriver à capter est le fait que le sens ait été conservé entre la source et l'hypothèse. Pour cela, nous voudrions pouvoir comparer les embeddings des deux langues source et hypothèse. Le TSNE de sklearn pourrait nous aider à visualiser ces embeddings dans une même dimension et ainsi pouvoir faire cette comparaison.

Pour chaque couple de phrase (source,hypothèse), nous récupérons les embeddings des mots les composants, réduisons ces embeddings grâce au TSNE en 2 dimensions et effectuons ensuite une moyenne des embeddings réduits pour avoir un vecteur à 2 dimensions pour chaque phrase que nous voulons comparer.

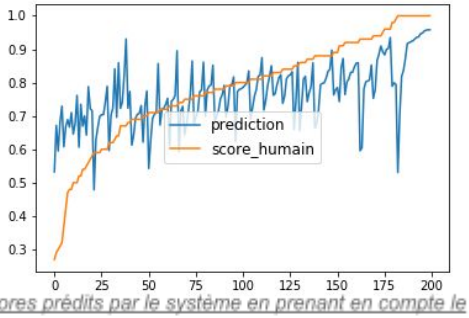
votre représentation graphique ne me parait pas très éclairante



cosinus entre les vecteurs réduits de source et d'hypothèse de traduction par rapport au score humain

Voici les scores de similarité cosinus entre les embeddings de la source et de l'hypothèse par rapport aux scores humains donnés, on a du mal à voir quelque corrélation visuellement.

et quantitativement ?



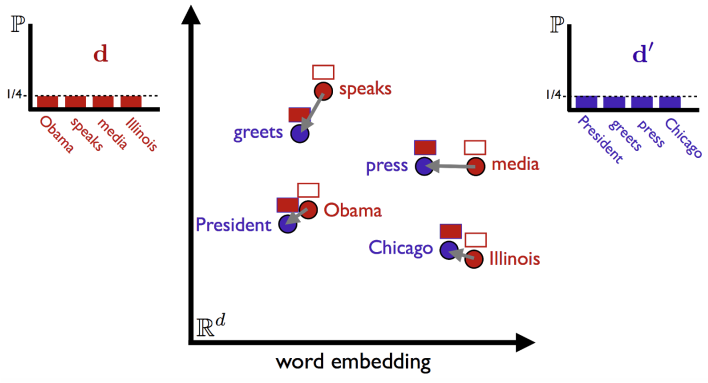
scores prédits par le système en prenant en compte le cosinus et la différence de longueur des phrases

Les performances obtenues en utilisant ces informations confirment qu'elles n'apportent rien de supplémentaire à ce que nous avions déjà puisque nous ne voyons rien de plus se dessiner visuellement, de plus la corrélation qui en résulte a même baissé depuis la précédente amélioration, elle est maintenant à 0,63.

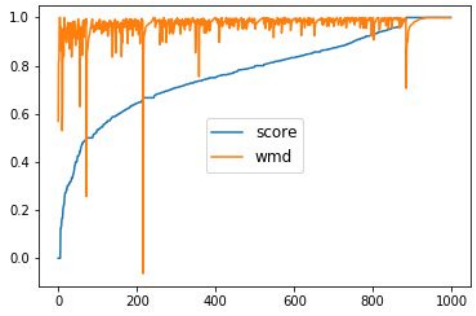
5. Word mover's distance similarity

Word mover's distance est une mesure servant à indiquer la distance entre 2 phrases ou documents, qu'ils aient des mots en commun ou non. Cette métrique utilise les embeddings des mots et son but est de calculer la somme des distances des mots du textes A pour correspondre aux mots du text B est de prendre la somme minimale parmi les possibilités.

On peut visualiser celà de la façon suivante pour les phrases : "Obama speaks to the media in Illinois" et "The president greets the press in Chicago"



Nous calculerons la similarité liée à cette distance en ajoutant le module wmd à la pipeline spacy. Cela pourrait normalement couvrir le cas de l'utilisation de synonymes par exemple.

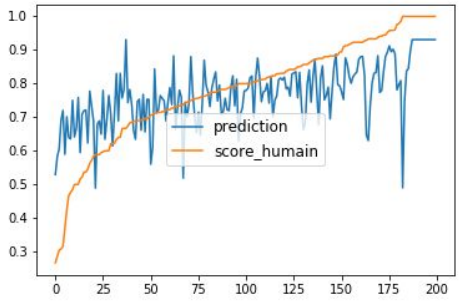


similarité WMD et score humain par score humain croissant

L'observation du graphe de cette similarité nous laisse à penser qu'elle ne semble pas suivre la courbe des scores humains, on a tout de même une corrélation de Spearmann de 53,6%

le coefficient de Spearmann n'est pas un %

un peut avant vous trouviez qu'une corrélation de .6 n'était pas terrible !



scores prédits par le système en prenant en compte la cosinus similarité et la différence de longueur des phrases, les bigrams, trigrams de mots, bigrams de PoS communs et similarité WMD

L'ajout de la similarité WMD n'améliore pas le modèle nous restons en effet à une corrélation de 64,7%. Cela est à priori dû à la ressemblance avec la mesure de cosinus similarité en effet on se rend compte que leur corrélation est de 75%

Il aurait peut être été intéressant d'obtenir cette similarité avec les embeddings camemBERT si cela était possible afin de voir si on continuait à avoir une grande ressemblance avec la cosinus similarité.
Nous ne garderons pas ce paramètre étant donné qu'il n'apporte pas d'amélioration visible.

Pourquoi ne pas chercher à combiner toutes ces caractéristiques ?

6. Pistes de recherche

Ce vaste sujet ne nous a malheureusement pas laissé le temps d'explorer l'intégralité des pistes auxquelles nous aurions voulu nous intéresser. En voici quelques unes qui nous semblent pertinentes :

Nous aurions aimé pouvoir réaliser les calculs d'embeddings mot à mot en utilisant les alignements ibm1. Le module AlignedSent de nltk nous aurait fourni les alignements entre la source et l'hypothèse et nous aurions pu ensuite comparer des vecteurs réduits par tsne. Cela aurait surement été plus significatif qu'une moyenne de vecteurs réduits.

Nous aurions voulu trouver un moyen de pénaliser les grosses erreurs syntaxiques, chose que le modèle de langue aurait du pouvoir capturer mais qui n'a pas été fructueux avec cette métrique.

Nous aurions également voulu ajouter la similarité de Wu-Palemet calculé à l'aide des synsets de Wordnet et de la distance entre les mots dans les arbres d'hyperonymes.

On aurait pu aussi, grâce au pos tagger et parser de spacy, essayer de vérifier les accords entre les noms et les adjectifs, ou d'autres relations de ce type, afin de pénaliser si les accords étaient incorrects.

L'ajout de caractéristiques ainsi qu'un nombre de données plus conséquentes nous auraient peut être permis d'obtenir des résultats plus satisfaisants.

Conclusion

Nous pouvons dire que les solutions que nous avons proposées d'apporter à la tâche ne fonctionnent pas et ne répondent pas suffisamment au problème, la corrélation avec le score humain atteignant au maximum, tout juste sous les 65%. Cette tâche nécessite de capturer automatiquement des nuances de sens entre les mots et d'ajustements syntaxiques qui sont très difficiles à modéliser.

On peut cependant voir que bien que minimes nous avons pu observer des améliorations après l'ajout de certaines caractéristiques, preuve de l'intérêt et de l'utilité de la combinaison de plusieurs facteurs pour imiter la métrique du score humain.

Ces résultats peuvent être la preuve de la difficulté à reproduire une métrique humaine de manière automatique. En effet, les locuteurs d'une langue sont capables de juger de l'agrammaticalité d'une phrase donnée ou bien de la similarité entre différentes phrases très facilement et rapidement bien qu'utilisant de nombreux critères, et ce de manière quasi inconsciente. Il n'est pas aisé de reproduire la multitude de caractéristiques prises en compte ainsi que la finesse avec laquelle un locuteur est capable d'effectuer l'évaluation de ces différentes caractéristiques. Et il faudrait en tout cas ajouter beaucoup plus de paramètres à notre modèle pour espérer l'améliorer et être représentatif de la pluralité des informations utilisées par l'humain pour son évaluation.

Sources

Modèle de langue :

https://kheafield.com/code/kenlm/estimation/

https://github.com/kmario23/KenLM-training

Embeddings :

https://camembert-model.fr/

Word mover's distance

https://github.com/src-d/wmd-relax

From Word Embeddings To Document Distances, Matt Kusner, Yu Sun, Nicholas Kolkin and Kilian Weinberger.