*Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles*
*Lille, France, 28 juin au 2 juillet 2021*

# Modification d'un modèle de liage d'entités nommées end-to-end par l'ajout d'embeddings contextuels

Valentin Carpentier[1]
(1) CNRS, LIMSI, Université Paris-Saclay, 91400, Orsay, France
valentin.carpentier@limsi.fr

RÉSUMÉ

Cet article présente les expériences effectuées sur un système de liage d'entités nommées. Cette tâche se découpe en deux principales parties que sont la détection de mentions méritant d'être liées à la base de connaissance et la désambiguïsation qui permet de sélectionner l'entité finale à lier à chaque mention. Deux approches existent pour résoudre cette tâche. Il y a celle de désambiguïsation seule et celle *end-to-end* qui effectue les deux sous-tâches simultanément. Nous nous sommes intéressés au modèle *end-to-end* atteignant l'état de l'art. Le cœur de ces expériences était d'exploiter des embeddings contextuels afin d'améliorer les performances. Trois approches ont été testées afin d'intégrer ces embeddings et de remplacer les embeddings de mots. Les différentes versions atteignent au mieux l'état de l'art. L'article présente quelques pistes déjà étudiées expliquant les raisons pour lesquelles les expériences testées ne dépassent pas le modèle initial et ouvrent des possibilités d'amélioration.

ABSTRACT

**Modifying an end-to-end named entity linking model by adding contextual embeddings**

This paper presents the experiments performed on a named entity linking system. This task is divided into two main parts, which are the detection of mentions deserving to be linked to the knowledge base and the disambiguation which makes it possible to select the final entity to be linked to each mention. Two approaches exist to solve this task. There are disambiguation-only one and *end-to-end* one that perform both subtasks simultaneously. We were interested in the *end-to-end* model reaching the state of the art results. The aim of these experiments was to experiment the use of contextual embeddings in order to improve performance. Three approaches have been tested to integrate these embeddings and replace word embeddings. The different versions reach the state of the art at best. The article discusses some previously explored avenues of why the experiments tested did not go beyond the initial model and open upgrading possibilities.

MOTS-CLÉS : mention, entité nommée, base de connaissances, approche de bout en bout, vecteurs sémantiques.

KEYWORDS: mention, named entity, knowledge base, end-to-end, embeddings.

## Introduction

La tâche de liage d'entités nommées permet d'identifier les éléments d'intérêts dans un texte et de les relier à une entrée d'une base de connaissances. Cette tâche est primordiale pour d'autres applications telles que le résumé de texte, les systèmes de question-réponses ou l'augmentation de bases de connaissances (Shen *et al.*, 2014). C'est cependant une tâche difficile car elle doit à la fois se

charger de repérer les mentions d'intérêts dans un texte et en même temps déterminer à quelle entrée de la base de connaissance chaque mention correspond. Or il est fréquent que relier une mention à une entité soit ambigu. Plusieurs entrées peuvent correspondre et se tromper entraîne alors la génération d'un contre-sens sur le texte analysé. Les deux tâches que sont (a) trouver les mentions puis (b) les relier à une entité peuvent être disjointes, beaucoup de travaux ne concernent que l'un des deux aspects. Néanmoins, effectuer la tâche d'un seul trait, par une approche dite *end-to-end*, peut avoir l'avantage de renforcer les performances globales du modèle. À partir d'un système *end-to-end*, l'objectif de ce travail est d'en modifier l'architecture afin de tenter de l'améliorer. L'axe privilégié a été l'amélioration des embeddings de mots, initialement du Word2Vec (Mikolov *et al.*, 2013), pour les remplacer par des embeddings contextuels de type BERT (Devlin *et al.*, 2018). Le modèle BERT a été choisi car il correspond à l'état de l'art en terme de modèle d'embeddings contextuels. Le travail présenté se base sur un corpus de liage d'entités en anglais.
Cette article présentera dans un premier temps la tâche de liage d'entités nommées ainsi que les travaux récents, puis présentera les différentes expériences réalisées avec BERT avant de présenter les résultats obtenus et de les discuter.

# 1 Travaux récents

La tâche de **Liage d'Entités Nommées** (LEN) consiste à repérer les mentions d'intérêts dans un document permettant sa compréhension et sa mise en contexte en les reliant à une base de connaissances. Elle est traditionnellement composée de deux sous-tâche (Shen *et al.*, 2014).
La première est **La Reconnaissance d'Entités Nommées** (REN) qui consiste à trouver dans un texte les mots ou groupes de mots significatifs qui doivent être mis en relation avec la base de connaissances, dont les éléments sont appelés *entités*. Ces mots ou groupes de mots sont appelés des *mentions* car ils mentionnent des éléments de la base de connaissances.
La seconde tâche est **La Désambiguïsation d'Entités Nommées** (DEN) qui consiste à relier à chaque mention la bonne entité dans la base de connaissances. C'est une tâche de désambiguïsation car plusieurs entités peuvent initialement correspondre à une même mention. Par exemple, si dans un texte on relève la mention "*le président français*", plusieurs entités issues d'une base de connaissances peuvent correspondre en premier lieu car il y a eu plusieurs présidents français (même si on se limite aux présidents de la République). De même, un texte comportant un nom comme "*Obama*" pourra correspondre à plusieurs entités car plusieurs personnes peuvent porter ce patronyme. Il faut donc déterminer duquel on parle. Cette tâche est souvent découpée en 2 temps (Shen *et al.*, 2014). On commence par générer l'ensemble des entités qui pourraient correspondre à la mention donnée (*Candidate Entity Generation*), puis on les classe de la plus pertinente à la moins pertinente pour prendre la décision d'association finale (*Candidate Entity Ranking*), la pertinence pouvant correspondre à la fréquence (moyen le plus naïf).
Ainsi, un système LEN a besoin d'une **base de connaissances** qui servira de référence pour associer les mentions de tous les textes. *Wikipedia* [1] est souvent retenu pour ce rôle. Chaque page correspond à une entité unique, et il n'existe pas deux pages référençant la même personne, institution, concept ou autre (on exclut les cas des pages traduites). Une bonne base de connaissances n'est pas qu'un simple dictionnaire et possède des liens entre les entités qui permettent de les situer les unes par rapport aux autres. Dans Wikipedia de tels liens peuvent s'exprimer par les *liens hypertextes* présents dans une page et permettant d'atteindre d'autres pages. Ces liens hypertextes étant associés à des mots ou groupes de mot, ils permettent aussi d'avoir des exemples de mentions devant être reliés à ces entités. Ces liens peuvent permettre aussi de définir des relations qui peuvent s'avérer utiles lors

1. http ://www.wikipedia.org/

du classement des entités comme leur fréquence. Les autres bases de connaissances fréquemment utilisées sont YAGO (Fabian *et al.*, 2007), DBPedia (Auer *et al.*, 2007) et Freebase (Bollacker *et al.*, 2008). Le travail sur la base de connaissances est fait en amont, le modèle l'utilise seulement.
Un système LEN a ensuite besoin d'un **corpus de textes** sur lequel le modèle devra extraire et lier les mentions vers les entités de la base de connaissances. Chaque texte du corpus (ou document) est composé de plusieurs phrases et potentiellement de plusieurs mentions afin de les mettre en relation, de les contextualiser et de faciliter ainsi le travail de désambiguïsation. Un corpus de textes peut être un recueil d'articles de presse. Les deux principaux corpus pour le liage d'entités nommées sont CoLNN (Hoffart *et al.*, 2011) et TAC [2] (Getman *et al.*, 2018; McNamee & Dang, 2009).
Certains modèles découplent les parties REN et DEN. La partie DEN est la plus complexe et il existe des systèmes permettant de réaliser la tâches REN en amont (tels que StandfordNER [3] ou OpenNLP [4]). Cependant, depuis quelques années, les modèles tentent davantage une approche *end-to-end* (Shen *et al.*, 2014). L'idée est de rendre la tâche REN plus robuste en utilisant les résultats de la désambiguïsation pour aider à mieux capturer les mentions. Par exemple, un système découplé aura tendance à mal étiqueter *The New York Times* pour le tronquer en *New York Times* ou simplement *New York* (l'article *the* étant le plus souvent oublié). Or, un bon étiquetage de mention peut permettre de faciliter sa correspondance avec une référence de la base de connaissance. DeepType (Raiman & Raiman, 2018) est un exemple de système de *désambiguïsation seule*. Il est la référence de l'état de l'art en DEN. End-to-End Neural Entity Linking (Kolitsas *et al.*, 2018) est un exemple de système *end-to-end*. Il est la référence de l'état de l'art pour cette approche.
Certains systèmes utilisent BERT (Devlin *et al.*, 2018) pour obtenir des *embeddings* contextuels car il peut être spécialisé pour réaliser une tâche de liage d'entités nommées (Broscheit, 2019). D'autres approches se concentrent sur une adaptation de BERT dans le cadre de *datasets* plus spécifiques auxquels BERT n'a pas été confronté, comme le fait PEL-BERT (Li *et al.*, 2020). BERT n'est pas le seul modèle d'*embeddings* contextuels (comme ELMo (Peters *et al.*, 2018) ou GPT (Radford *et al.*, 2019)), mais il constitue la référence.
Nous avons choisi de nous appuyer sur un modèle NEL pré-existant : le *End-to-End Neural Entity Linking* de Kolitsas (Kolitsas *et al.*, 2018). Il effectue la tâche d'*Entity Linking* (EL) de la *détection de mention* (REN) jusqu'à la *désambiguïsation* (DEN). Ce choix a été motivé car le modèle de Kolitsas est à l'état de l'art actuel en tant que système *end-to-end*. Des travaux comme ceux de Broscheit (Broscheit, 2019) l'utilise comme point de comparaison. Son architecture est, de plus, aisée à modifier, contrairement à des modèles comme DeepType qui repose principalement sur son système de types. Ainsi, il était plus envisageable d'explorer l'impact de nouveaux modules en les utilisant sur le modèle de Kolitsas. Enfin, les modules utilisés par ce système sont des outils éprouvés tels que Word2Vec (Mikolov *et al.*, 2013) et des *Bidirectional Long Short-Term Memory* (Bi-LSTM). Il était donc intéressant de tester des améliorations en intégrant des approches plus récentes ayant prouvé leur robustesse (notamment BERT).

# 2 Modèle Initial

Cette section a pour but d'expliquer le fonctionnement du modèle *End-to-End Neural Entity Linking* de Kolitsas (Kolitsas *et al.*, 2018), illustré dans la Figure 1 afin de comprendre par la suite les modifications qui y ont été apportées. Le modèle part d'un document, soit un texte cohérent, de quelques phrases. Le modèle y détecte les mentions, les relie aux entités correspondantes dans la base de connaissances et renvoie une liste des couples Mention-Entités ainsi prédits dans le document.

2. https ://tac.nist.gov/2010/RTE/
3. https ://nlp.stanford.edu/ner/
4. http ://opennlp.apache.org/

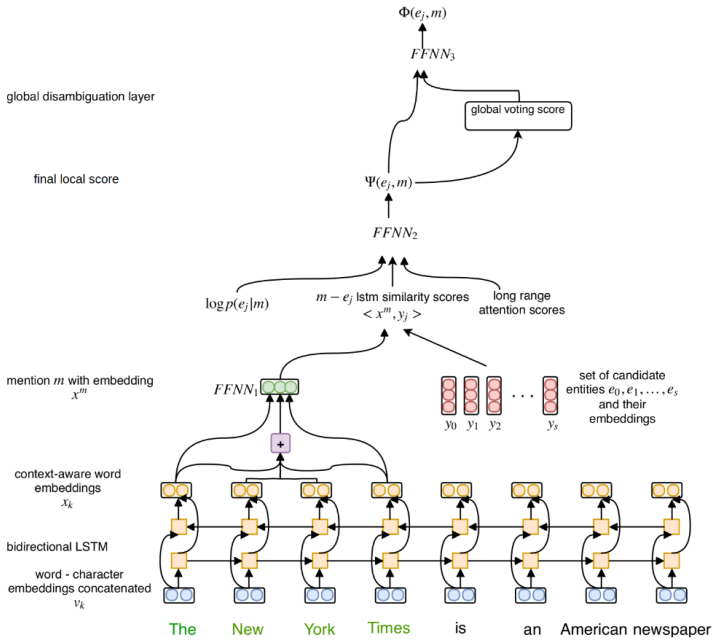


FIGURE 1 – Architecture du modèle End-to-End Neural Entity Linking (Kolitsas *et al.*, 2018)

Le principe général est de construire un **embedding de mention** à partir des embeddings de mots. Cet embedding de mention pourra être comparé à des embeddings d'entités pré-calculés, permettant d'assigner un score de similarité à chacun. Enfin, on choisit la meilleure entité candidate grâce au score de similarité des embeddings et de la cohérence globale en utilisant les autres entités présentes dans le document. L'entrée est constituée des **embeddings de mots *Word2Vec* pré-entraînés**. Les embeddings de mots finaux $\{\nu_k\}_{k\in 1,n}^{-}$ sont obtenus par concaténation des embeddings de mots *Word2Vec* et des embeddings de caractères de chaque mot. Ces embeddings de caractères sont appris grâce à un bi-LSTM appliqué sur les caractères du mot (étape absente sur la figure 1). Ces embeddings de mots sont ensuite transformés en **embeddings de contexte** $\{x_k\}_{k\in 1,n}^{-}$ (appelé *context-aware word embedding* dans la figure 1) grâce à un bi-LSTM. Une fois les embeddings de contexte fixés, ceux pertinents dans la construction d'une mention (c'est-à-dire les embeddings de contexte correspondant à la mention $m = w_q, ..., w_r$) sont combinés grâce à un réseau Feed Foward (appelé ***FFNN de Mention*** ou $FFNN_1$ dans la figure 1) pour transformer l'ensemble en **Embedding de Mention**. Le modèle ne prédéfinit pas les mentions qui seront construites. Il construit et teste toutes les mentions possibles. Compte tenu des notations précédentes, l'Embedding de Mention est obtenu ainsi :

$$x^m = FFNN_1(g^m) \text{où } g^m = [x_q; x_r; \hat{x}^m] \text{ et } \hat{x}^m = \sum_{k=q}^{r} \alpha_k^m \nu_k.$$

Les coefficients $\alpha_k^m$ sont obtenus ainsi :

$$\alpha_k^m = \frac{exp(\alpha)}{\sum_{t=q}^{r} exp(\alpha_k)} \text{ et } \alpha_k = \langle w_\alpha, x_k \rangle$$

Chaque **Embedding de Mention** est ensuite comparé à une liste de candidats $((e_j)_{\geq 1})$ issus des **Embeddings d'Entités** pré-entraînés (Ganea & Hofmann, 2017) $(y_e)_{\in wikipedia}$ et sélectionnés par des probabilités **pré-calculées** (ou *prior* $(p(e_j, m))$ à partir des liens hypertextes de Wikipedia. Ces probabilités ont été établies lors de l'apprentissage des embeddings d'entités et sont considérées comme acquises par le modèle. Chaque couple Entité-Mention reçoit ensuite un score de similarité $\Psi$ obtenu par le réseau Feed Foward (appelé ***FFNN de Score*** ou $FFNN_2$ dans la figure 1) :

$$\Psi(e_j, m) = FFNN_2([logp(e_j, m); \langle x^m; y_i \rangle])$$

Seules les entités avec un score final suffisamment haut sont testées (supérieur à un paramètre $\gamma'$). Ceci permet de filtrer les mentions construites par le modèle qui ne correspondent à aucune réelle entité. On obtient donc l'ensemble des couples (mention, candidat) sérieux :

$$V_G = \{(m, e) \in M, e \in (e_j)_{\geq 1}, \Psi(e, m) \geq \gamma'\}$$

Enfin, le modèle procède à une désambiguïsation globale qui permet d'unifier les entités retenues en prenant en compte la cohérence globale entre toutes les entités sélectionnées. On compare donc chaque candidat retenu avec l'ensemble des candidats retenus pour les autres mentions

$$G(e_j, m) = cos(y_{e_j}, y_G^m) \text{ où } y_G^m = \sum_{e \in V_G^m} y_e \text{ et } V_G^m = \{e | (m', e) \in V_G \wedge m' \neq m\}$$

Le vote final consiste à la combinaison par le réseau Feed Foward appelé ***FFNN de Vote***, ou $FFNN_3$ dans la figure 1, de cette comparaison avec le score $\Psi$ :

$$\Phi(e_j, m) = FFNN_3([\Psi(e_j, m), G(e_j, m)])$$

On obtient ainsi en sortie, pour chaque mention initiale du texte, l'entité à laquelle elle fait référence. C'est à partir de cette architecture que nous allons chercher à améliorer le modèle.

# 3 Expériences et résultats

Cette section présente les travaux réalisés pour remplacer les embeddings de mots Word2Vec (Mikolov *et al.*, 2013) par des embeddings plus robustes. Il a été choisi d'utiliser BERT (Devlin *et al.*, 2018) car il s'agit des embeddings contextuels les plus performants de l'état de l'art. De plus, la première étape du modèle consiste à transformer les embeddings de mots en embeddings de contexte par l'intermédiaire d'un bi-LSTM.

## 3.1 Protocole expérimental

Trois manières d'utiliser les embeddings contextuels BERT (Devlin *et al.*, 2018) sont explorées.

— La première (*Hypothèse 1*) consiste à remplacer la couche d'embeddings de mots et le bi-LSTM de contextualisation par les embeddings BERT. L'idée est de considérer que les embeddings contextuels de BERT ont déjà l'information initialement produite par le bi-LSTM et peuvent donc le remplacer. Le but est de vérifier s'ils sont plus performants que ceux générés par le modèle.

— La seconde (*Hypothèse 2*) consiste à remplacer les embeddings de mots par les embeddings contextuels BERT. L'idée ici est de considérer que les embeddings contextuels BERT peuvent être assimilés à des embeddings de mots (et non plus contextels) plus puissants que ceux de Word2Vec. Ils passent donc par le bi-LSTM afin de créer du contexte.

— La dernière (*Hypothèse 3*) est de considérer que les embeddings BERT peuvent apporter de l'information supplémentaire et venir en soutien du bi-LSTM. L'idée est de combiner les embeddings BERT aux embeddings de contexte issus du bi-LSTM en supposant que la connaissance issue de BERT va ainsi améliorer la qualité de l'embedding contextuel global.

Les trois méthodes ont donc été testées selon les protocoles décrits ci-dessous. Les expériences ont toutes été menées sur 50 itérations. Les autres paramètres d'apprentissage ont été conservés identiques à ceux originellement utilisés par Kolitsas (Kolitsas *et al.*, 2018) à l'exception des modifications précisées.

### 3.1.1 Utilisation des embeddings en tant qu'embeddings de mots

Comme illustré dans la figure 2, le modèle subit peu de modifications pour l'*Hypothèse 1*. On remplace uniquement les embeddings de mots initiaux (Word2Vec) par ceux extraits depuis BERT.

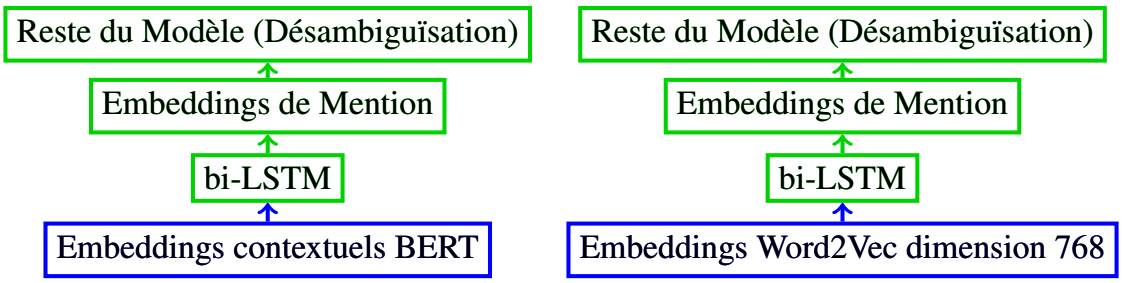


FIGURE 2 – `WordBERT`, à gauche, et équivalent `Word2Vec768` à droite

Seuls les embeddings de caractères ont été retirés du processus (WordBERT de gauche).
Afin de comparer réellement l'impact des embeddings BERT sur le résultat et le séparer de l'impact des changements d'architecture (c'est-à-dire la taille des embeddings et la suppression des embeddings de caractère), une version identique avec Word2Vec a été également testée (Word2Vec768). De plus, pour des raisons techniques, certains documents ont été retirés, car il était impossible de générer des embeddings BERT pour ces derniers. La figure 3 donne un exemple de tel document. Cela correspond à 1485 entités (8%) sur l'ensemble d'entraînement, 298 (6%) sur AIDA Test A et 314 (7%) sur AIDA Test B. Afin d'éviter des biais, le modèle initial (Kolitsas *et al.*, 2018) a été ré-entraîné en enlevant ces exemples supprimés.

```
TENNIS - FRIDAY 'S RESULTS FROM THE U.S. OPEN . NEW YORK 1996-08-30 Results from the U.S. Open Tennis Championships at the National Tennis Centre on
Friday ( prefix number denotes seeding ) : Women 's singles , third round Sandrine Testud ( France ) beat Ines Gorrochategui ( Argentina ) 4-6 6-2 6-1
Men 's singles , second round 4 - Goran Ivanisevic ( Croatia ) beat Scott Draper ( Australia ) 6-7 ( 1-7 ) 6-3 6-4 6-4 Tim Henman ( Britain ) beat Doug
```

FIGURE 3 – Exemple de document retiré du corpus AIDA non tokenisable par BERT

### 3.1.2 Utilisation des embeddings BERT en tant qu'embeddings contextuels

Comme illustré dans la figure 4, le modèle subit une modification plus importante pour l'*Hypothèse 2*. On supprime la couche de bi-LSTM pour connecter directement la couche de génération de l'embedding de mention avec les embeddings contextuels BERT. On retire les embeddings de caractères, non pertinents si l'on considère que BERT possède déjà l'information nécessaire. On supprime enfin le mécanisme d'attention qui liait les embeddings d'entités aux embeddings de contexte (en sortie du bi-LSTM). Le reste du modèle est inchangé.

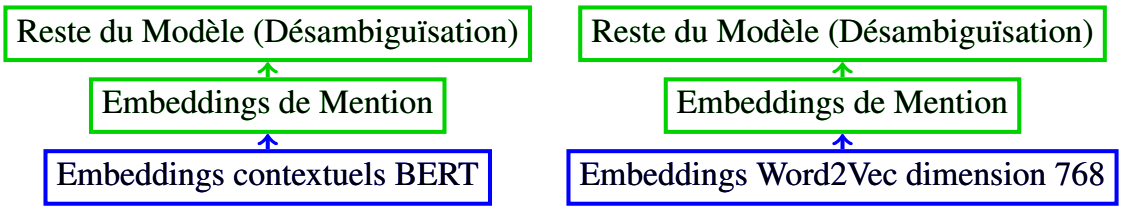


FIGURE 4 – `ContextBERT` à gauche, et équivalent `Context Word2Vec` à droite

### 3.1.3 Association des embeddings de BERT avec les embeddings contextuels du modèle

Comme présenté dans la figure 5, l'objectif dans l'*Hypothèse 3* est d'ajouter des embeddings contextuels BERT à ceux de contexte issus du bi-LSTM. Pour cela, l'architecture initiale est conservée, mais en concaténant les embeddings contextuels BERT aux embeddings de contexte du bi-LSTM correspondants.

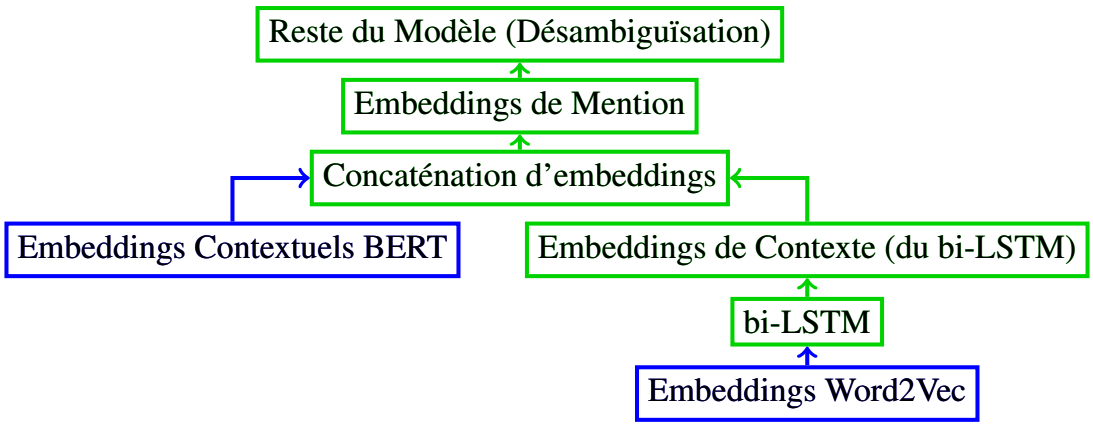


FIGURE 5 – Expérience BERT+LSTM

## 3.2 Données utilisées

Ces différentes expériences ont été menées sur le même corpus que celui utilisé par Kolitsas. Il s'agit du corpus AIDA/CoNLL (Hoffart *et al.*, 2011) qui est le plus gros corpus public de liage d'entités nommées en anglais. Il est composé d'un ensemble d'entraînement de 18448 mentions liées dans 946 documents. L'ensemble de validation (AIDA test A) contient 4791 mentions dans 216 documents et l'ensemble de test (AIDA test B) contient 4485 mentions dans 231 documents.
Les variantes BERT ont été obtenues en extrayant les embeddings depuis la version pré-entraînée de BERT de base pour chaque document du corpus AIDA/CoNLL. Comme BERT renvoie 12 couches d'embeddings, il faut choisir lesquelles considérer comme embeddings contextuels. Plusieurs méthodes existent comme indiqué dans (Devlin *et al.*, 2018).
Après expérience, nous avons choisi de sommer les quatre dernières couches ce qui donne de meilleures performances par rapport aux autres méthodes de combinaison des couches BERT (Devlin *et al.*, 2018).

## 3.3 Résultats et Analyses

| Approche | mic/mac F1 AIDA Test A | mic/mac F1 AIDA Test B |
|---|---|---|
| Modèle de Base (Kolitsas *et al.*, 2018) | 86.6 / 89.4 | 82.6 / 82.4 |
| BERT fine-tuné (Broscheit, 2019) | 87.3 / 92.3 | 79.3 / 81.1 |
| Modèle Réajusté | 89.7 / 87.6 | **85.5** / 84.8 |
| Word BERT | 87.1 / 84.3 | 83.7 / 83.2 |
| Context BERT | 85.9 / 81.8 | 83.3 / 81.7 |
| BERT + LSTM | 89.2 / 87.1 | 85.4 / **84.9** |
| Word2Vec 768 | 83.0 / 79.8 | 79.9 / 76.4 |
| Context Word2Vec | 68.9 / 66.1 | 62.6 / 61.2 |

TABLE 1 – micro / macro F1 des différentes utilisations de BERT (en **gras** les meilleurs résultats et en vert les seconds meilleurs)

Les modèles sont évalués en termes de macro et micro F1. La micro F1 est la F1 calculée sur l'ensemble des documents du dataset. La macro F1 est la moyenne des F1 sur les différents documents du dataset. Un modèle stable dans ses prédictions sur les différents documents présentés aura donc une micro et une macro F1 similaires.

Une analyse de la signification des performances a été menée afin de déterminer à partir de quel seuil on peut dire qu'il y a une amélioration de résultat, certains biais empêchant une reproductibilité exacte des résultats. Chaque expérience a donc été lancée dix fois pour calculer un écart-type entre les différents résultats de chaque expérience puis un écart-type des différences entre toutes les expériences. Chaque itération de chaque expérience étant identique aux autres, les écarts de résultats traduisaient une variabilité qui ne pouvait pas être interprétée comme étant une amélioration du modèle. Cette analyse a montré que deux expériences ayant des performances avec un écart inférieur ou égal à 0.3 points de F1 pouvaient être considérées comme étant équivalentes.
Les résultats des expériences sont présentés dans le tableau 1. Les expériences reportées sont représentatives de celles répétées plusieurs fois sans en être avoir des résultats marginaux. Les analyses des résultats s'appuient exclusivement sur les expériences reportées dans le tableau 1. Le Modèle de Base correspond aux résultats du modèle de Kolitsas (Kolitsas *et al.*, 2018) tel qu'il est proposé tandis que le Modèle Réajusté correspond au même modèle entraîné sur le dataset tronqué des exemples non utilisés dans les autres expériences. C'est donc à ce modèle que l'on compare les résultats. BERT fine-tuné correspond à l'expérience de Broscheit consistant à utiliser directement BERT comme système de liage d'entités nommées. Il se compare lui-même à Kolitsas dans son article. Word BERT utilise un vocabulaire tel que chaque occurrence de chaque mot soit présente de manière distincte. Word2Vec 768 conserve l'architecture initiale adaptée pour les embeddings BERT mais en utilisant des embeddings Word2Vec de dimensions similaires à BERT (768). Context BERT utilise un vocabulaire tel que chaque occurrence de chaque mot soit présente de manière distincte. Les embeddings BERT sont directement utilisés pour la génération de l'embedding de mention. Context Word2Vec conserve l'architecture initiale adaptée pour les embeddings BERT mais en utilisant des embeddings Word2Vec de dimensions similaires à BERT (768). Le BERT + LSTM correspond au modèle où l'on vient concaténer les embeddings BERT aux embeddings de contexte du bi-LSTM en sortie de ce dernier. Le reste du modèle est inchangé.

### 3.3.1 Analyse Quantitative

Dans le tableau 1, on remarque que le Modèle Réajusté (*85.5 / 84.8*) est équivalent à BERT + LSTM (*85.4 / 84.9*). On observe également que Word BERT (*83.7 / 83.2*) est légèrement supérieur à Context BERT (*83.3 / 81.7*). On peut en conclure que les embeddings BERT ne possèdent initialement pas plus d'information que ce que peut fournir le bi-LSTM au cours de son apprentissage. En revanche, ils semblent plus adéquats en étant utilisés comme embeddings de contexte secondaires, pour appuyer les résultats du bi-LSTM.
Ensuite, on constate une forte différence entre le Modèle de Base (*82.1 / 83.6*) et le Modèle Réajusté (*85.5 / 84.8*). Cela signifie que les éléments retirés du datasets sont des éléments sur lesquels le modèle initial de Kolitstas se trompait. En les oberservant on remarque qu'il s'agit en effet pour la plupart de tableaux de résultats sportifs (exemple figure 3) qui ne correspondent pas à un texte narratif où l'on peut se reposer sur le contexte entre les mots pour inférer son sens.
On constate alors que Word BERT (*83.7 / 83.2*) ne permet pas d'atteindre les performances initiales (*85.5 / 84.8*). En revanche, la même expérience testée avec les embeddings Word2Vec (c'est-à-dire Wor2Vec 768) obtient des résultats encore plus faibles (*79.9 / 76.4*).
Context Word2Vec est très significativement moins bon (*62.6 / 61.2*). C'est un résultat attendu car on y considère des embeddings de mots comme embeddings de contexte. Cela a du sens de partir du principe que les embeddings BERT possèdent une information contextuelle car ils ont été appris dans ce but.
En revanche, il n'y a pas de raison que des embeddings de mots possède une information contextuelle. L'expérience permet néanmoins de mettre en évidence l'importance du bi-LSTM dans le traitement des embeddings de mots.

De manière générale, aucune des expériences n'atteint ou ne dépasse significativement les performances initiales du modèle. Plusieurs pistes d'explications ont été explorées.

### 3.3.2 Analyses Qualitatives

Nous avons commencé par regarder plus en détail les prédictions faites par le modèle et les différentes variantes BERT. Les analyses se sont focalisés sur WordBERT, BERT+LSTM et le Modèle de Base Réajusté. Afin de les mener, les mots proches des entités mal prédites ont été visualisées via une représentation T-SNE (exemple figure 6). Toutes les visualisations (figure 6 à figure 11) sont faites pour dix entités mal prédites par les trois variantes du modèle testé, sur le dataset AIDA Test A, ayant soit le plus haut, soit le plus bas score de similarité. La légende indique, pour chaque entité représentée, le mot issu de la mention choisi pour représenter l'entité, l'entité prédite par le modèle et enfin l'entité de référence qui aurait dû être prédite. Concrètement, pour chaque entité mal prédite, le premier mot de la mention menant à cette prédiction a été récupéré. Puis, les entités ont été triées en fonction du score de similarité attribué par le modèle entre la mention et l'entité. Dix entités ont été sélectionnées, soit avec le plus haut score de similarité, soit le plus faible. Les dix mots les plus proches sémantiquement du mot menant à l'entité prédite ont enfin été affichés.
Cette procédure a été réalisée en extrayant indépendamment pour chacun des 3 modèles les embeddings de mots puis dans un second temps les embeddings de contexte, c'est-à-dire à la sortie du bi-LSTM ou – pour le cas de l'expérience BERT+LSTM – juste avant la génération de l'embedding de mention. Ces analyses ont permis d'avancer certaines hypothèses.
Les figures 6 à 9 ne présentent pas les résultats pour le Modèle Réajusté mais sont équivalents. La figure 6 montre les embeddings de mots qui sont identiques entre le Modèle Réajusté et BERT+LSTM. De même, que la nature des embeddings de mots ne changent pas le traitement du bi-LSTM, ce qui s'est retrouvé dans les figures du Modèle Réajusté et de WordBERT.
Les figures 6 à 11 sont données en annexe.

#### Répartition des entités et impact du bi-LSTM

Les entités mal prédites ont été récupérées afin de les comparer en fonction de l'ensemble dont elle provenait. Les entités erronées sont réparties de manière homogène dans les datasets. C'est-à-dire qu'en prenant en compte le nombre d'occurrences de chaque entité mal prédite, il n'y a pas d'entité qui prend une place significative parmi celles-ci. Le maximum est à 2.5% de l'ensemble des occurrences apparaissant dans les erreurs et le minimum à 0.1% quand il n'y a qu'une seule occurrence de l'entité. Ainsi, la moitié des erreurs sur AIDA Test A et les $2/3$ des erreurs sur AIDA Test B sont sur des entités qui ne se trouvaient pas dans l'ensemble d'entraînement.
En observant les figures générées à partir des embeddings de mots (figure 6 et 8) et celles à partir des embeddings de contexte (figure 7 et 9), on peut visualiser l'impact du bi-LSTM sur l'adaptation des embeddings de mots afin de les faire coïncider avec le contexte du document. Certaines des erreurs de prédictions d'entités sont liées à une mauvaise désambiguïsation de la mention. Par exemple, on voit l'évolution du contexte entourant le mot `China` dans les figures 8 (embeddings de mots) et 7 (embeddings de contexte). Ce terme doit mener à la prédiction de l'entité `Qing Dynasty`. Dans la figure 8, ce dernier est très proche du terme `China`. On a peu d'information sur le contexte dans lequel il se trouve. Or, le bi-LSTM va créer du contexte et on observe dans la figure 7 que les termes proches sont désormais des noms de pays tels que `Spain`, `Norway` ou `Nigeria`. On perd ainsi tout contexte relatif à une dynastie impériale au profit seul d'un pays, ce qui correspond à la prédiction du modèle. Ces erreurs sont principalement des exemples qui se trouvent dans le dataset AIDA Test A mais pas dans le dataset d'entraînement.
En observant de plus les figures 6 à 9 sur les variantes BERT, on peut remarquer que le bi-LSTM ne produit aucun contexte sur la variante BERT + LSTM. En effet, on peut observer ceci en comparant

les figures sur la variante WordBERT (figure 6 et 7) et BERT + LSTM (figure 8 et 9). Dans la figure 6, on a la représentation d'embeddings BERT seuls qui prennent ensuite un contexte produit par le bi-LSTM dans la figure 7. Or, cette représentation est très similaire à celle des embeddings de contexte de BERT + LSTM (figure 9), et plus similaire qu'avec la représentation typique après le bi-LSTM. Cela montre que les informations apprises par le bi-LSTM sont diluées par l'ajout des embeddings BERT qui sert de contexte seul. Cependant, l'usage d'embeddings BERT comme seul contexte ne fonctionne pas entièrement comme l'a montré la différence de performance entre BERT + LSTM (*85.4 / 84.9*) et ContextBERT (*83.3 / 81.7*). Le mécanisme d'attention porté sur les embeddings de mots, présent dans la version BERT + LSTM mais absent dans la version ContextBERT, pourrait dans ce cas expliquer cette différence.

**Erreurs liées au seuil de validation**

Jusque là, les observations se sont portées que sur les prédictions erronées par les trois principales variantes. Or, en prenant les mauvaises prédictions effectuées uniquement par un modèle, on remarque que beaucoup d'entités sont en réalité correctement prédites (exemple figure 10). Cet effet s'accentue lorsque l'on regarde les entités mal prédites avec un score de similarité faible (exemple figure 11) où la quasi totalité des entités sont correctement prédites. Le modèle s'évaluant simultanément sur la tâche de repérage des mentions et de désambiguïsation, cela aurait pu souligner des erreurs de positionnement des mentions. Or, l'évaluation autorise une position des mentions permissive de sorte que seule une mention entièrement extérieure à celle attendue sera décomptée comme fausse. De plus, les figures 10 et 11 montre que le début des empans est cohérent avec l'entité prédite, ce qui exclut l'hypothèse d'une mention mal positionnée. L'entité est donc bien correctement placée et prédite. Le seul élément pouvant la discréditer reste le seuil de validation d'une prédiction. Ce seuil permet d'exclure toutes les prédictions qui se feraient sur des éléments qui n'ont pas à être prédits et est appris par le modèle en fonction des documents qu'ils voient durant son apprentissage. Si une entité est correctement prédite avec un score en dessous de ce seuil, elle sera décomptée comme une errance du modèle. Ces cas, présents également sur WordBERT et BERT+LSTM, représentent 40% des erreurs de prédiction sur le dataset d'entraînement et 45% sur les datasets AIDA Test. On remarque donc que les erreurs de prédiction se répartissent entre les erreurs communes surtout représentées par des entités mal désambiguïsées, et les erreurs spécifiques à chaque modèle surtout représentées par des scores de similarité trop faibles.

## Conclusion

Dans cet article, nous avons présenté la tâche de liage d'entités nommées et nous nous sommes intéressés plus en détail au modèle *End-to-End Neural Entity Linking* (Kolitsas *et al.*, 2018). Ce modèle permet de réaliser la tâche selon une approche end-to-end de la détection de mention jusqu'à la désambiguïsation. Il est au niveau de l'état de l'art et nous nous sommes donc intéressés à l'étudier plus en profondeur pour voir comment l'améliorer. Cet objectif était motivé par l'architecture basique du modèle, n'utilisant que des outils éprouvés. Il y avait donc un potentiel d'amélioration en usant d'approches plus récentes. La piste explorée ici fut l'intégration d'embeddings contextuels BERT (Devlin *et al.*, 2018) pour remplacer les embeddings de mots Word2Vec (Mikolov *et al.*, 2013) initialement présents. Malheureusement, les différents modèles testés n'ont pas apporté d'amélioration significative. Plusieurs pistes ont été envisagées et seront suivies pour expliquer la sous-performance des embeddings BERT, parmi lesquelles réduire l'impact de la taille des embeddings ou l'architecture globale. Cela pourra être dans l'optique de réussir à obtenir les mêmes performance entre le modèle initial et Word768 pour appliquer les modification aux versions avec BERT. Dans la suite de ce premier travail, nous allons analyser la répartition sémantique des mots afin de comprendre d'où proviennent

les erreurs de prédiction d'entités, aussi bien du côté initial que du côté des versions utilisant BERT. Enfin, nous modifierons l'architecture du modèle. Le point clé étant la transformation des embeddings de mots en embeddings de contexte, l'approche envisagée sera de changer le bi-LSTM se chargeant de cette tâche par un transformer.

# Références


AUER S., BIZER C., KOBILAROV G., LEHMANN J., CYGANIAK R. & IVES Z. (2007). Dbpedia : A nucleus for a web of open data. In *The semantic web*, p. 722–735. Springer.

BOLLACKER K., EVANS C., PARITOSH P., STURGE T. & TAYLOR J. (2008). Freebase : a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, p. 1247–1250.

BROSCHEIT S. (2019). Investigating entity knowledge in bert with simple neural end-to-end entity linking. *In Proceedings of the 23rd Conference on Computational Natural Language Learning, 677–85. Association for Computational Linguistics, 2019.*

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

FABIAN M., GJERGJI K., GERHARD W. *et al.* (2007). Yago : A core of semantic knowledge unifying wordnet and wikipedia. In *16th International World Wide Web Conference, WWW*, p. 697–706.

GANEA O.-E. & HOFMANN T. (2017). Deep joint entity disambiguation with local neural attention. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2619–29. Association for Computational Linguistics, 2017.*

GETMAN J., ELLIS J., STRASSEL S., SONG Z. & TRACEY J. (2018). Laying the groundwork for knowledge base population : Nine years of linguistic resources for tac kbp. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

HOFFART J., YOSEF M. A., BORDINO I., FÜRSTENAU H., PINKAL M., SPANIOL M., TANEVA B., THATER S. & WEIKUM G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 782–792.

KOLITSAS N., GANEA O.-E. & HOFMANN T. (2018). End-to-end neural entity linking. *In Proceedings of the 22nd Conference on Computational Natural Language Learning, 519–29. Association for Computational Linguistics, 2018.*

LI S., CUI W., LIU Y., MING X., HU J., HU Y. & WANG Q. (2020). Pel-bert : A joint model for protocol entity linking. *ArXiv*, **abs/2002.00744**.

MCNAMEE P. & DANG H. T. (2009). Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, p. 111–113.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv :1310.4546*.

PETERS M. E., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. *arXiv preprint arXiv :1802.05365*.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, **1**(8), 9.

RAIMAN J. R. & RAIMAN O. M. (2018). Deeptype : multilingual entity linking by neural type system evolution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

SHEN W., WANG J. & HAN J. (2014). Entity linking with a knowledge base : Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, **27**(2), 443–460.

# Annexe

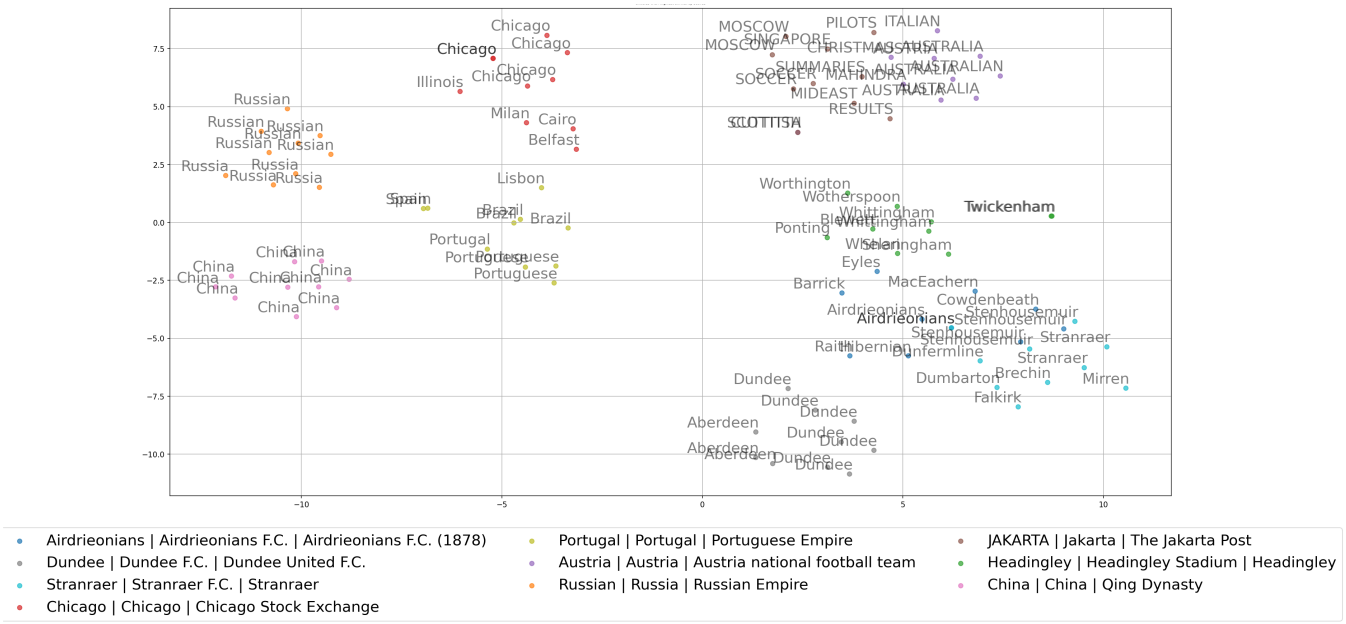


FIGURE 6 – Embeddings de mots de WordBERT, entités mal prédites, plus haut score de similarité.

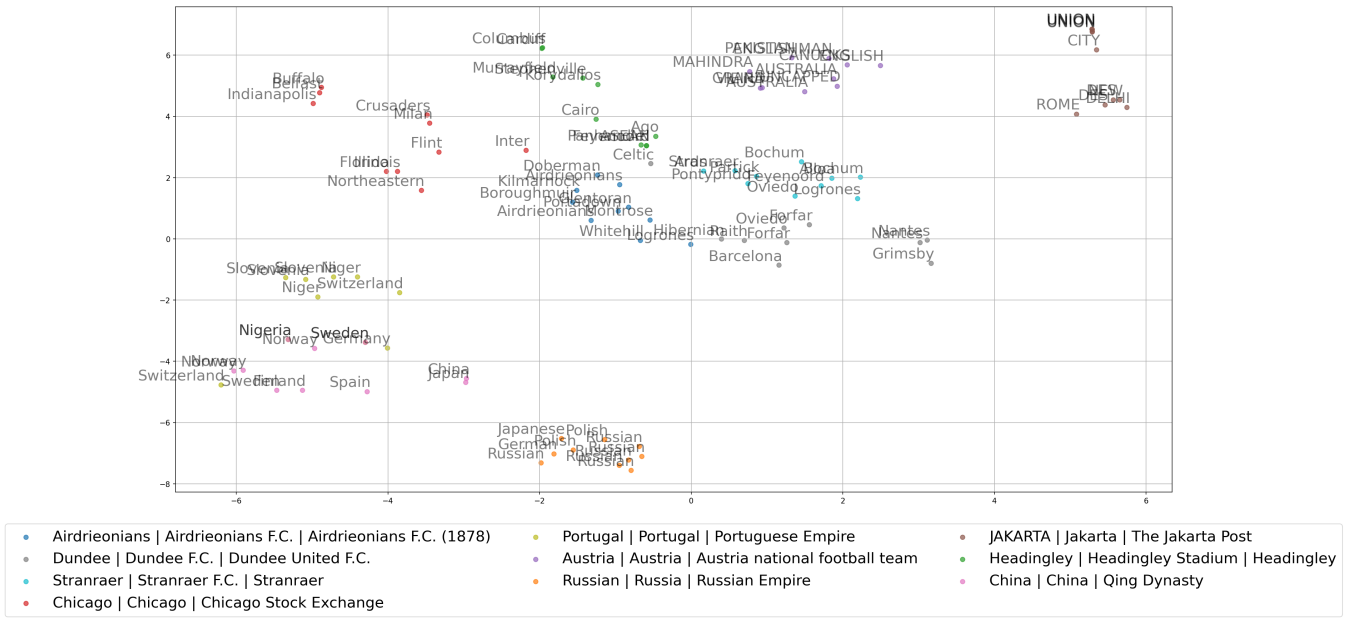


FIGURE 7 – Embeddings de contexte WordBERT, entités mal prédites, plus haut score de similarité.

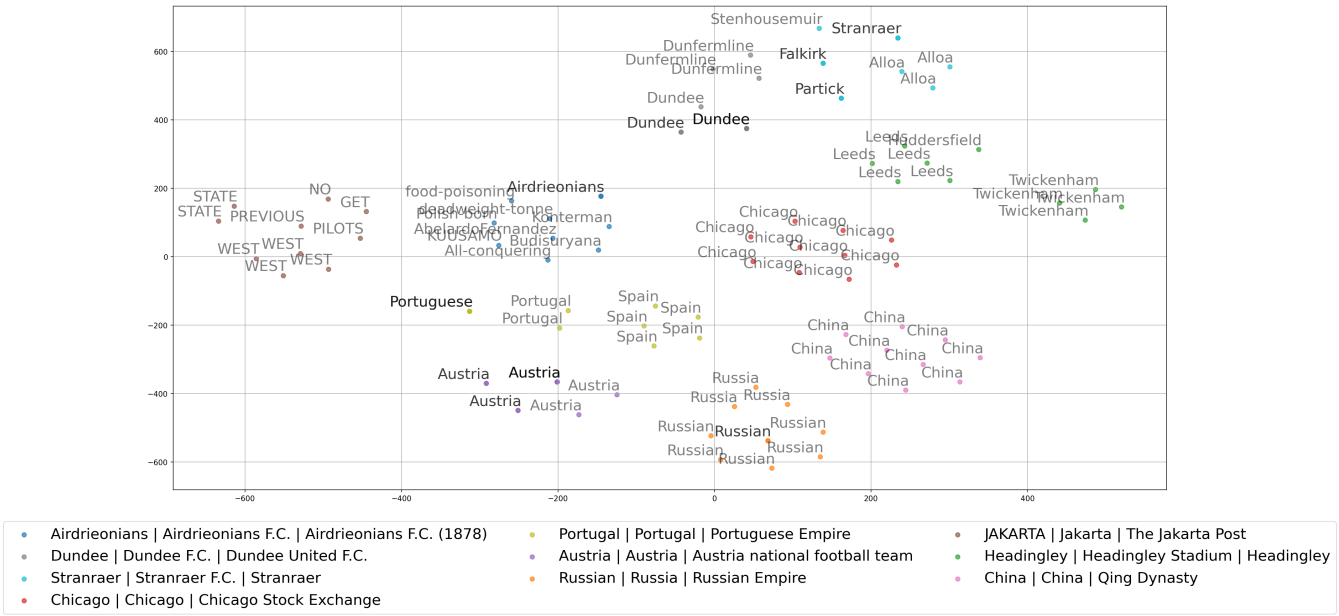


FIGURE 8 – Embeddings de mots BERT+LSTM, entités mal prédites, plus haut score de similarité

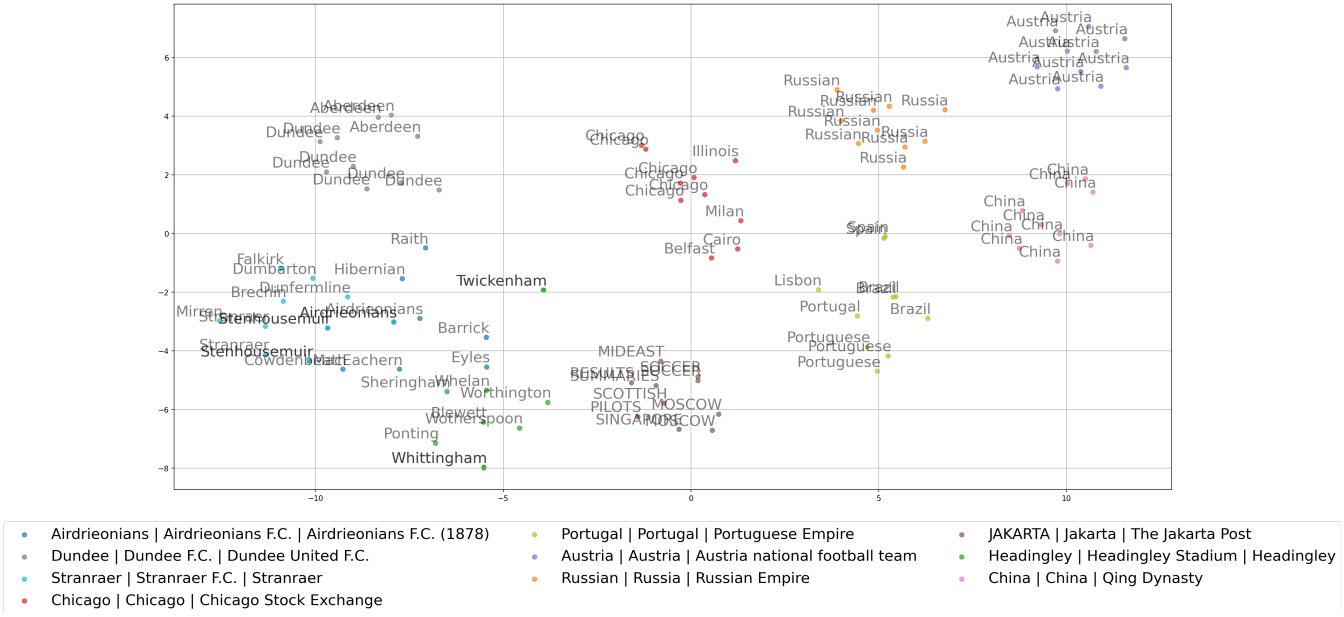


FIGURE 9 – Embeddings de contexte BERT+LSTM, entités mal prédites, plus haut score de similarité.

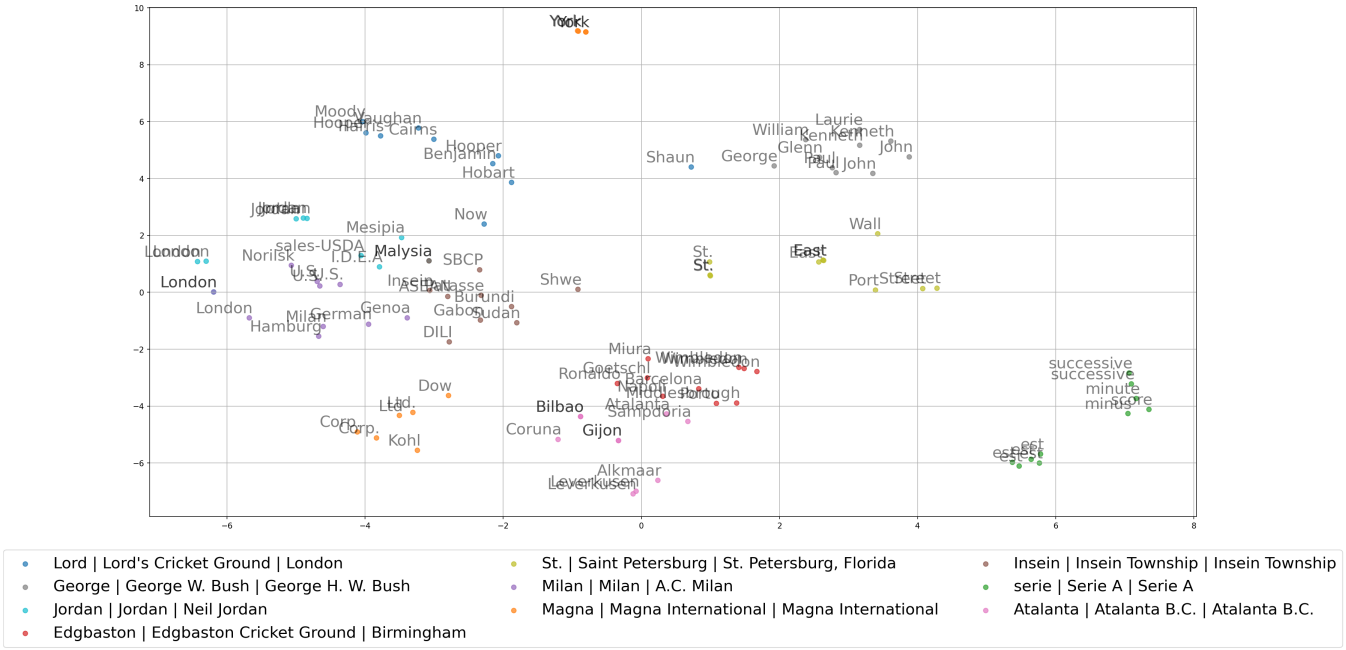


FIGURE 10 – Embeddings de contexte du Modèle Réajusté, entités mal prédites, plus haut score de similarité.

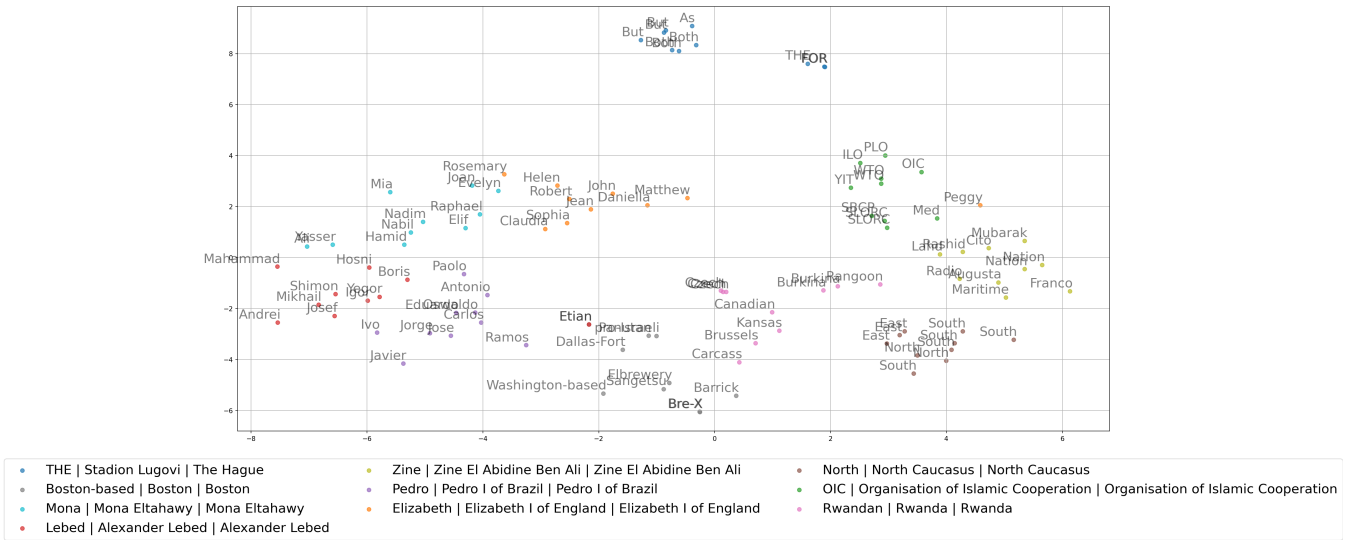


FIGURE 11 – Embeddings de contexte du Modèle Réajusté, entités mal prédites, plus bas score de similarité.