

Évaluation de l'apport des chaînes de coréférences pour le liage d'entités

Léo Labat^{1,2} Lauriane Aufrant²

(1) ENS, Paris, France

(2) Inria, Paris, France

prenom.nom@inria.fr

RÉSUMÉ

Ce travail propose de revisiter les approches de liage d'entités au regard de la tâche très proche qu'est la résolution de coréférence. Nous observons en effet différentes configurations (appuyées par l'exemple) où le reste de la chaîne de coréférence peut fournir des indices utiles pour améliorer la désambiguïsation. Guidés par ces motivations théoriques, nous menons une analyse d'erreurs accompagnée d'expériences oracles qui confirment le potentiel de stratégies de combinaison de prédictions au sein de la chaîne de coréférence (jusqu'à 4.3 F1 sur les mentions coréférentes, en anglais). Nous esquissons alors une première preuve de concept de combinaison par vote, en explorant différentes heuristiques de pondération, qui apporte des gains modestes mais interprétables.

ABSTRACT

Evaluating the benefits of coreference chains for entity linking.

This work proposes to revisit entity linking approaches in light of the closely related task of coreference resolution. Indeed, we observe several configurations (supported by examples) in which the rest of the coreference chain can yield valuable cues to improve disambiguation. Driven by these theoretical motivations, we conduct an error analysis as well as oracle experiments, which confirm the potential of combination strategies for predictions within the coreference chain (up to 4.3 F1 on coreferring mentions, in English). We then sketch a first proof of concept of voting-based combination, by exploring various weighting heuristics, which yields limited but interpretable gains.

MOTS-CLÉS : liage d'entités, résolution de coréférence, approches jointes.

KEYWORDS: entity linking, coreference resolution, joint approaches.

1 Introduction

Reconnaître les entités mentionnées dans un texte demeure un défi pour le traitement automatique des langues, qui approche cette compétence linguistique par deux tâches : le liage d'entité et la résolution de coréférence. Le liage d'entité (*entity linking* en anglais) consiste à associer toute chaîne de caractères se référant à une entité dans un texte, que l'on appelle une mention, à une entrée d'une base de connaissances, tandis que la résolution de coréférence regroupe les mentions se référant à des entités en ensembles de sorte que toutes les mentions d'un même ensemble se réfèrent à la même entité. Néanmoins, à chacune de ces tâches est souvent associé un concept de mentions différent : en liage, une mention est le plus souvent un nom propre¹, alors qu'en résolution de coréférence, tout

1. Le domaine biomédical constitue à cet égard une exception notable.

empan (*span*) se référant à une entité peut être coréférent, pourvu qu'un autre empan dans le même document se réfère à la même entité. Ces différences rendent bien souvent les performances de liage et de résolution incomparables car réalisées sur des corpus distincts. Ainsi, le corpus de référence pour le liage d'entité AIDA-CONLL (Hoffart *et al.*, 2011), qui comporte des liens entre les mentions de CONLL 2003 (Sang & Meulder, 2003) et la base YAGO 2 (Hoffart *et al.*, 2013) n'inclut pas de pronoms parmi ses mentions. Quant à la coréférence, le jeu de données le plus utilisé, ONTONOTES 5.0 (Pradhan *et al.*, 2013), inclut les pronoms et des mentions moins spécifiées qu'en liage dans ses chaînes de coréférence, mais ne contient pas d'annotation pour les singletons, c'est-à-dire que lorsqu'une entité n'est mentionnée qu'une fois, cette mention solitaire n'est pas annotée.

Nous faisons l'hypothèse que les informations de coréférence peuvent améliorer les décisions prises par un algorithme de liage. En effet, les mentions de coréférence peuvent contenir des indices utiles à la désambiguïsation d'entités, car tant les pronoms que les périphrases sont autant d'indices qui permettent de savoir de qui ou de quoi il est question. La plupart des algorithmes de liage prennent déjà en compte le contexte pour assigner un lien à une mention donnée, mais la particularité de la coréférence, en tant que dépendance à longue distance, est d'agréger au fil du texte des mentions qui elles-mêmes ou avec leur contexte, peuvent être directement désambiguïsantes, quelle que soit leur répartition dans le document. L'exemple présenté en Figure 1, inspiré de (Hajishirzi *et al.*, 2013) montre comment une chaîne de coréférence peut contenir des informations utiles à la désambiguïsation.

[*Michael Eisner*]₁ and [*Donald Tsang*]₂ announced the grand opening of [*Disneyland*]₃ yesterday.
 [*Eisner*]₁ thanked [*the President*]₂ and welcomed fans to [*the Hong Kong park*]₃.

FIGURE 1 – Exemple de partage d'information entre résolution de coréférence et liage d'entités

Les mentions *Disneyland* et *the Hong Kong park* sont ambiguës lorsqu'elles sont considérées isolément, même dans leur contexte à l'échelle de la phrase. Mais si on les identifie comme coréférentes, déterminer l'entité à laquelle elles se réfèrent est aisé : il s'agit de Disneyland Hong Kong, et non de Disneyland Paris ou de Hong Kong Park. Nous proposons donc dans cet article d'explorer comment ces indices potentiellement utiles fournis par la coréférence peuvent être exploités pour améliorer les décisions de liage d'entités.

Après un aperçu de ces deux tâches et des approches associées (§2), nous présentons dans cet article notre cadre expérimental reposant sur le corpus AIDA-CONLL (§3), qui nous permettra d'affiner nos motivations (§4) et d'explorer une première preuve de concept pour l'exploitation de ces indices (§5) en estimant les gains potentiels à insérer ces informations dans des modèles de l'état de l'art.

2 Etat de l'art

2.1 Liage d'entités

Formellement, le liage d'entités consiste à prendre en entrée un document textuel D contenant un ensemble de mentions se référant à des entités ($M = \{m_i\}_i$), ainsi qu'une base de connaissances KB dont les entrées sont des entités nommées ($E = \{e_i\}_i$), et à retourner un ensemble de paires (m_i, e_i) où e_i est l'entité référencée par la mention m_i dans E . L'ensemble E peut parfois contenir une étiquette spéciale "NIL" pour dénoter qu'il s'agit d'une entité inconnue dans la base. Lorsqu'il est réalisé de bout en bout, les mentions ne sont pas données en entrée et doivent être détectées par

l’algorithme.

Approche modulaire L’approche modulaire peut être implémentée de diverses façons. Une décomposition proposée par [Piccinno & Ferragina \(2014\)](#) consiste en trois étapes : un module de détection associe d’abord à chaque mention possible un certain nombre de candidats, puis un module de désambiguïsation en sélectionne un pour chaque mention et un module d’élagage (*pruning*) élimine les liens les moins probables. D’autres divisions modulaires sont possibles, comme par exemple REL ([van Hulst et al., 2020](#)), qui réalise successivement une détection de mentions, une génération de candidats pour chaque mention et enfin une désambiguïsation, c’est-à-dire la sélection d’un seul candidat pour chaque mention.

Modèles de langue pré-entraînés Comme pour de nombreuses autres tâches de TALN, le liage d’entités a bénéficié de l’introduction de représentations pré-entraînées à l’aide de Transformers ([Vaswani et al., 2017](#)). Appliquer un encodeur de type BERT ([Devlin et al., 2019](#)) à des mots, empan et entités permet d’intégrer le contexte dans le calcul de similarité mention/entité. [Kolitsas et al. \(2018\)](#) complète cette approche avec une table de correspondance probabiliste entre mentions et entités construite à partir d’hyperliens Wikipédia ([Ganea & Hofmann, 2017](#)). Les modèles autorégressifs peuvent aussi être employés pour reformuler le liage comme une tâche de génération de texte : par exemple, GENRE ([Cao et al., 2021](#)) génère des séquences associant mentions tirées du texte et des identifiants d’entités, en contraignant la génération de ces derniers à partir d’une liste prédéfinie. [Mrini et al. \(2022\)](#) utilisent ce type de génération pour combiner détection de mentions, désambiguïsation et reclassement en mode multi-tâches.

Questions-Réponses En tant que tâche consistant en une forme de classification d’empan, le liage d’entité peut également être formulé comme une tâche de questions-réponses. C’est le cas de [Zhang et al. \(2021\)](#), qui obtiennent des performances comparables aux approches précédentes, mais au prix de ressources de mémoire et de calcul plus importantes, nécessitant de surcroît des canvas élaborés à la main.

Classification de tokens Une approche distincte consiste à poser le liage comme une tâche de classification de tokens. La décision de liage est alors prise à l’échelle du token sur l’ensemble du vocabulaire, c’est-à-dire sur l’ensemble des entités possibles : [Broscheit \(2019\)](#) mobilise un modèle BERT sur lequel est superposée un simple réseau à propagation avant pour obtenir une distribution de probabilités sur les 700 000 entités possibles pour chaque token. A la classifications de tokens peut également s’ajouter des classifications à des échelles suprasegmentales : le liage revient alors à réaliser une prédiction structurée. C’est le cas de SPEL ([Shavarani & Sarkar, 2023](#)), qui agrège de façon structurée des décisions de classification prises sur trois niveaux : au niveau des tokens, au niveau des mots puis au niveau des syntagmes (*phrase level*). Ces approches présentent l’avantage d’exploiter la capacité des encodeurs de modèles de langue à prendre en compte le contexte dans leur stratégie d’agrégation de prédictions pour former des annotations d’empan cohérentes.

2.2 Résolution de coréférences

Formellement, la résolution de coréférence consiste à prendre en entrée un document D constitué de texte et à identifier toutes les mentions qui se réfèrent à la même entité dans le monde réel. Contrairement au liage d’entités, où les mentions sont explicitement identifiées et liées à une base d’entités nommées, la résolution de coréférence vise à regrouper les mentions $M = \{m_i\}$ qui font référence à la même entité sans base de connaissances. Le résultat est un ensemble de chaînes de coréférence $C = \{c_i\}$, où chaque chaîne c_i est un sous-ensemble de M tel que toutes les mentions dans c_i se réfèrent à la même entité.

Approches par combinaisons de n-grams Le paradigme qui s’est imposé à partir de [Lee et al. \(2017\)](#) consiste à réaliser la résolution de coréférence sans détection de mentions préalable en attribuant des scores à des empanns ainsi qu’à des paires de mentions. Cette approche surgénère des mentions en considérant tout empann comme potentielle mention. Des stratégies d’élargissement de mentions ont été affinées avec [Lee et al. \(2018\)](#), mais les gains de performances tiennent principalement de l’amélioration de la qualité des représentations vectorielles : d’abord avec des LSTMs ([Hochreiter & Schmidhuber, 1997](#)) pour [Lee et al. \(2017\)](#) puis avec ELMo ([Peters et al., 2018](#)) pour [Lee et al. \(2018\)](#). L’avènement de modèles de langue tels que BERT ont donc également donné lieu à d’importants gains de performances en résolution de coréférence. Une variante de BERT spécifiquement dédiée à l’encodage d’empanns, SpanBERT, a été élaborée par [Joshi et al. \(2019\)](#) et a démontré l’intérêt d’un objectif d’entraînement consistant à prédire des empanns plutôt que des seuls mots dans le cadre d’une modélisation de langue masquée pour améliorer les performances de résolution de coréférence. Néanmoins, la méthode *Start-to-End*, ou S2E ([Kirstain et al., 2021](#)), montre qu’une approche semblable basée sur les Longformer ([Beltagy et al., 2020](#)) qui se contente de représentations contextuelles du premier et du dernier token de chaque mention réduit considérablement la complexité du problème et obtient des résultats comparables plus rapidement et avec moins de ressources de calcul.

Formulation générative Il existe également une approche générative du problème de la résolution de coréférence. En se fondant sur un modèle multilingue T5 ([Raffel et al., 2019](#)), d’aucuns ont proposé un système de transitions pour faire réaliser des décisions d’association de mentions à des chaînes de coréférence par un modèle génératif ([Bohnet et al., 2023](#)). Les modèles génératifs peuvent également être utilisés pour générer des prédictions de coréférence en reproduisant directement le texte fourni en entrée avec des annotations ([Zhang et al., 2023](#)).

Questions-réponses D’autres ([Wu et al., 2020](#)) ont reformulé la tâche de coréférence comme une tâche de réponse aux questions (*Question Answering*) en s’appuyant notamment sur SpanBERT. Les empanns proposés au système de questions-réponses sont générés de façon similaire à [Lee et al. \(2017\)](#) et sont introduits dans des modèles de question prédéfinis contenant des encarts à remplir avec une mention donnée.

2.3 Approches jointes

Une proposition de modélisation jointe du liage d’entités et de la résolution de coréférence élaborée par [Hajishirzi et al. \(2013\)](#) montre comment la réalisation simultanée d’une tâche peut contribuer à améliorer les performances de l’autre et réciproquement. L’implémentation est modulaire et construit de façon incrémentale des chaînes de coréférence par application de règles déterministes de constitution d’ensembles de mentions coréférences qui satisfassent des contraintes d’entités, c’est-à-dire qui garantissent la cohérence des décisions de liage.

D’autres formulations jointes des deux tâches explorent le potentiel des informations de coréférence pour améliorer les performances de liage. En effet, les implémentations du liage d’entité sus-mentionnées considèrent tantôt une liste finie de candidats tantôt une liste générée *ad hoc* pour la mention courante. Si la bonne entité n’est pas présente dans la liste prédéfinie ou générée lors de l’inférence, alors l’algorithme de liage n’a aucune chance de prendre une décision de liage exacte. C’est le problème que des auteurs comme [Zaporojets et al. \(2022\)](#) cherchent à résoudre en s’inspirant de [Angell et al. \(2021\)](#) qui, dans leur implémentation de liage d’entités sur des données biomédicales, rendent possibles les décisions de liage entre les mentions plutôt qu’uniquement entre mentions et sommets de la base de connaissances. De la sorte, une mention pour laquelle la bonne entité

n'est pas présente dans la liste de candidats générée pour elle peut néanmoins être liée à une autre mention, elle-même susceptible d'avoir été bien liée. Néanmoins, cette approche telle qu'elle a été implémentée s'en tient à une conception des mentions référentes ancrée du côté du liage, du fait de la nature-même des données d'entraînement : AIDA-CONLL et le corpus DWIE (Zaporojets *et al.*, 2020) ne contiennent pas d'annotation de liage pour les pronoms, par exemple. La notion de coréférence est donc réduite au fait que les mentions liées à la même entité sont coréférentes dans le sens où elle se réfèrent à la même entité, mais ignore donc le phénomène linguistique de coréférence dans ses manifestations les plus ordinaires (pronoms, périphrases, etc.).

3 Conditions expérimentales

3.1 Données

Toutes nos expériences reposent sur AIDA-CONLL (Hoffart *et al.*, 2011), corpus anglais journalistique de 20 744 phrases annotées en liage d'entités avec la base de connaissances YAGO 2 (Hoffart *et al.*, 2013) (28813 mentions liées à 5586 entités, parmi les 50 millions de YAGO 2). Nous utilisons la version du corpus fournie par Zaporojets *et al.* (2022), qui inclut des liens supplémentaires (Kolitsas *et al.* (2018) ayant identifié la version originale comme incomplète à l'échelle du document). Nous conservons leur partition en données d'entraînement (946 documents, 19190 mentions, 4085 entités), de développement (TEST-A, 216 documents, 4960 mentions, 1649 entités) et de test (TEST-B, 231 documents, 4663 mentions, 1547 entités), issue du corpus sous-jacent CONLL 2003 (Sang & Meulder, 2003). Plus de la moitié des entités apparaissant en test (57%, 882 entités parmi 1547) sont nouvelles, c'est-à-dire jamais rencontrées à l'entraînement.

3.2 Métriques

On considère la précision, le rappel et la métrique F1 (micro-F1) appliqués à la détection de mentions ($F1^m$), au liage d'entités lui-même ($F1^e$, pour les cas où les mentions correctes sont déjà données), et à la combinaison de la détection et du liage ($F1^{m+e}$). Nous employons la variante stricte du score F1 : une mention est bien liée uniquement si ses frontières sont correctement prédites et si le lien prédit correspond au lien de référence dans le corpus.

Nous suivons la pratique usuelle de l'état de l'art de restreindre l'évaluation aux mentions dont l'entité (prédite ou de référence) est effectivement présente en base (configuration "InKB" de Röder *et al.* (2018)).

À des fins d'analyse, nous mesurons aussi le score F1 sur des sous-ensembles spécifiques de mentions : mentions de référence ($F1_R$), mentions détectées lors de la détection de mention du liage ($F1_L$), mentions détectées lors de la détection de mention de la coréférence ($F1_C$), mentions de référence qui ont été détectées pour le liage ($F1_{R+L}$), mentions détectées à la fois pour le liage et la coréférence ($F1_{L+C}$), etc.

3.3 Modèles

Liage d'entités Nous expérimentons avec trois modèles de liage d'entités couvrant des approches variées :

- **Un liage modulaire classique** : REL (van Hulst *et al.*, 2020) combine des modules de détection de mentions (reconnaissance d'entités nommées par FLAIR (Akbik *et al.*, 2018)), de génération d'entités candidates (suivant (Ganea & Hofmann, 2017) : 4 candidats les plus probables d'après

la mention et 3 d’après leur similarité au contexte, en sommant les plongements sur 50 tokens) et de désambiguïsation d’entités (sélection d’un candidat par mention, avec prise en compte des relations latentes entre mentions pour la cohérence de leurs liages (Le & Titov, 2018), sur une fenêtre de 200 tokens). Nous utilisons le modèle *REL 2019*, où les plongements Wikipedia2Vec (Yamada *et al.*, 2016) utilisés pour la génération de candidats sont calculés sur l’export Wikipédia de juillet 2019. Le module de désambiguïsation utilise les plongements GloVe (Pennington *et al.*, 2014) et est entraîné sur AIDA-CONLL.

- **Une approche par prédiction structurée** : SPEL (Shavarani & Sarkar, 2023) utilise RoBERTa (Liu *et al.*, 2019) pour estimer pour chaque sous-mot les probabilités d’un ensemble (fixe) d’entités candidates, puis combine les 20 candidats les plus probables à l’échelle du mot, et refait encore une prédiction à l’échelle du syntagme. Nous utilisons les deux versions `SpEL-base` et `SpEL-large` (suivant la version de RoBERTa), avec les mêmes hyperparamètres que Shavarani & Sarkar (2023) : une fenêtre de contexte de 284 tokens avec chevauchement de 20 tokens, et un ensemble fixe de 5601 candidats (dont "null").
- **Une approche générative** : GENRE (Cao *et al.*, 2020) affine un modèle de langue pré-entraîné BART (Lewis *et al.*, 2019) pour générer des identifiants d’entités (avec un décodage contraint pour qu’ils soient valides) afin de désambiguïser les mentions pré-identifiées. Dans sa version de bout en bout, les prédictions sont réalisées en insérant dans le texte des crochets aux limites de mentions en plus des identifiants d’entité. Nous utilisons les mêmes hyperparamètres que Cao *et al.* (2020) : décodage avec faisceau de taille 10 à l’apprentissage (6 au test) et au plus 15 étapes, 384 tokens de contexte maximal.

Résolution de coréférence Notre approche est applicable à n’importe quel algorithme de coréférence qui génère ses propres mentions. Pour nos expériences nous utilisons S2E (Kirstain *et al.*, 2021), que nous avons réentraîné avec les mêmes hyperparamètres et les mêmes données OntoNotes 5.0 (Pradhan *et al.*, 2013). Les auteurs rapportent un score $F1^m$ moyen de 80.3.

4 Motivations théoriques et empiriques

Les interactions fortes entre la tâche de liage d’entités et celle de résolution de coréférences amènent naturellement la question de leur combinaison à émerger. Afin de préciser cette intuition, nous évaluons les modèles de liage comparativement sur l’ensemble des mentions et uniquement sur les mentions coréférentes. Il en ressort (Table 1) que les mentions coréférentes sont typiquement plus difficiles à lier, en particulier pour les modèles les moins performants.

	R^{m+e}	R_C^{m+e}	$R_{C(\text{ref})}^{m+e}$
REL	75.5	71.4	74.8
SPEL-BASE	88.9	87.6	88.3
SPEL-LARGE	89.8	90.5	89.8

TABLE 1 – Impact de l’appartenance à une chaîne de coréférence sur le rappel

Une analyse manuelle sur des exemples du jeu de développement TEST-A a révélé 3 configurations théoriques d’intérêt :

1. Suite à une mention non-ambiguë (ex : prénom et nom), les suivantes y font référence d’une manière ambiguë (ex : prénom) et leur contexte contient trop peu d’indices pour les désambiguïser individuellement.

2. La première mention est ambiguë, mais la suite du texte ajoute de l'information en la centrant sur des mentions anaphoriques (ex : "Il habite à Paris. Cette petite ville du Texas a été fondée en 1845.").
3. Chacune des mentions présente une ambiguïté lorsque prise individuellement, mais en combinant tous leurs indices l'entité est claire. C'est le cas de l'exemple illustré en Figure 1.

Afin de quantifier ces effets, du moins les configurations 1 et 2 (la configuration 3 étant plus dure à objectiver), on mesure séparément les précisions, rappels et F1 suivant la position des mentions dans la chaîne de coréférence. Plus précisément, on affecte chaque mention à un quintile suivant la valeur $\frac{position-1}{taille_{chaîne}-1}$. Les résultats rapportés en Table 2 montrent une nette baisse de performance du deuxième jusqu'au dernier quintile, ce qui est cohérent avec la configuration 1. Le résultat bas sur le premier quintile est moins intuitif, mais peut facilement s'interpréter au regard du style du corpus AIDA-CONLL (articles de presse), où chaque document commence par un titre. Comme on peut l'observer en Figure 2, les premières mentions présentes dans un titre d'article sont souvent concises et ambiguës (configuration 2), et c'est le début de l'article qui apporte des informations substantielles. Dans cet exemple particulier on observe bien une combinaison des configurations 1 et 2.

		Quintile de la position dans la chaîne				
		0-19%	20-39%	40-59%	60-79%	80-100%
REL	$F1_{C(\text{ref})}^{m+e}$	73.6	84.2	82.5	83.5	73.7
SPEL-BASE	$F1_{C(\text{ref})}^{m+e}$	91.4	94.7	93.1	93.0	91.2
SPEL-LARGE	$F1_{C(\text{ref})}^{m+e}$	93.4	94.9	94.2	95.5	91.9
GENRE	$F1_{C(\text{ref})}^e$	81.4	82.2	83.2	78.2	80.2

TABLE 2 – Score F1 de liage en fonction de la position de la mention dans sa chaîne de coréférence

BASEBALL-GONZALEZ HOMERS TWICE AS RANGERS BEAT INDIANS.
 ARLINGTON, Texas 1996-08-31
Juan Gonzalez homered twice and Ivan Rodriguez added a two-run shot as the Texas Rangers defeated the Cleveland Indians 5-3 in a matchup of division leaders Friday.
 Rodriguez's 18th homer, off Chad Ogea (7-5) in the first, gave Texas a 2-0 lead. One out later, **Gonzalez** smacked his 40th homer, extending his hitting streak to 20 games.
Gonzalez, who hit in 21 straight games earlier this season, joined Mickey Rivers as the only players in Texas history with two 20-game streaks in the same year.
Gonzalez hit his second homer in the third for his fifth multi-homer game of the season. **Gonzalez** has three 40-homer seasons and his 121 RBI broke Ruben Sierra's team record of 119 set in 1989. (...)

FIGURE 2 – Répartition de l'ambiguïté le long d'une chaîne de coréférence (document 1059, TEST-A)

5 Amélioration du liage d'entités via les coréférences

Les résultats de la section précédente soulignent un certain potentiel à exploiter l'information de coréférence pour enrichir le liage, d'une part car les autres mentions de la chaîne peuvent apporter des informations utiles et d'autre part car les mentions coréférentes présentent souvent de plus grandes difficultés pour le liage.

Nous suggérons ici une approche où pour chaque mention à lier (tour à tour), on commence par désambiguïser les autres mentions de la chaîne (y compris si non-détectées comme mentions à lier : on force une désambiguïstation) puis on combine ces éléments pour informer le liage en cours.

Il est à noter que cette stratégie n'est applicable que pour une partie des mentions à lier : celles pour lesquelles des mentions coréférentes existent bien sûr, mais surtout celles qui sont détectées comme telles par le modèle de coréférence. En effet, outre les inévitables erreurs de ce dernier, les divergences de conventions d'annotation mènent aussi à des décalages entre mentions détectées de part et d'autre (par exemple "vendor Valery Ivankov" côté coréférence et "Valery Ivankov" côté liage), et donc une impossibilité d'exploiter les indices de coréférence, faute d'adéquation des mentions. En l'occurrence, sur les 5616 mentions de référence dans TEST-B, seulement 1804 (32%) d'entre elles appartiennent à une chaîne de coréférence prédite par S2E. Il serait envisageable d'explorer des heuristiques plus permissives pour utiliser les indices de coréférences fournis par des mentions non-identiques, mais avec un risque important d'introduire du bruit, et cela sort du cadre du présent travail.

Nous commençons par des expériences oracles pour estimer les gains envisageables, puis nous évaluons quelques heuristiques de vote, permettant d'esquisser des pistes de recherche future. L'ensemble de ces expériences est réalisé sur le jeu de test TEST-B. Nous expérimentons avec GENRE uniquement, dont les performances sont davantage interprétables, dans la mesure où forcer la désambiguïsation de mentions supplémentaires a en soi un impact sur la performance globale de REL (qui considère l'interaction entre prédictions), et la notion de mention n'existe que partiellement dans SPEL étant donné qu'elles sont construites hiérarchiquement à partir de sous-mots.

5.1 Expériences oracles

Afin d'estimer le gain potentiel que peuvent apporter les indices issus du reste de la chaîne de coréférence, on regarde si l'entité de référence est au moins présente parmi toutes les prédictions faites au sein de la chaîne de coréférence. Le cas échéant, une stratégie optimale de combinaison de ces prédictions permettrait de faire ressortir la bonne entité et l'attribuer à l'ensemble de la chaîne.

Les résultats présentés en Table 3 révèlent une marge de progression de +1.2 F1 dans l'ensemble, qui monte à +4.3 F1 lorsqu'on ne regarde que les mentions détectées par la coréférence (les seules pour lesquelles l'approche est donc applicable), la prédiction étant mécaniquement inchangée pour les autres.

	$F1_R^e$	$F1_R^e$ avec oracle	$F1_{R+C}^e$	$F1_{R+C}^e$ avec oracle
GENRE	72.4	73.6	75.7	79.4

TABLE 3 – Mesure oracle du gain de performance accessible avec une sélection parfaite de l'entité parmi les prédictions de la chaîne de coréférence

Ces résultats expérimentaux confirment le caractère prometteur de l'approche. En effet dans l'exemple "Il habite à Paris. Cette petite ville du Texas a été fondée en 1845.", même si les indices de la première occurrence ne sont pas suffisants pour désambiguïser "Paris", des indices supplémentaires sont disponibles (et suffisants) pour désambiguïser la mention "Cette petite ville" ("Texas", "1845") et cette information peut alors être remontée pour désambiguïser "Paris". Cela suggère qu'en exploitant bien ces autres prédictions au sein de la chaîne, il serait possible de corriger une part substantielle des erreurs de liage.

Une limite de cette approche est toutefois sa dépendance à une bonne génération de candidats, surtout pour les mentions non-nommées : en ne considérant que la mention "Cette petite ville", il est en effet peu probable que l'entité "Paris (Texas)" soit identifiée comme candidate, auquel cas la désambiguïsation échouerait et ne permettrait pas de contribuer à la désambiguïsation de "Paris". Ce

point particulier dépend de la méthode utilisée pour le liage d’entités, car toutes n’incluent pas une étape de génération de candidats, ou pas sous la même forme. Pour les méthodes de liage où cela a du sens, une approche possible pour pallier cette difficulté serait de fusionner au préalable les candidats générés pour toutes les mentions de la chaîne de coréférence, augmentant ainsi les options aussi bien pour "Paris" que pour "Cette petite ville". Avec cette stratégie, la marge de progression est en fait plus importante encore.

5.2 Combinaison de prédictions

Nous évaluons ici empiriquement l’intuition suivant laquelle les autres mentions de la chaîne de coréférence peuvent apporter des indices utiles pour désambiguïser la mention d’intérêt. On teste donc plusieurs heuristiques de combinaison des prédictions au sein de la chaîne. Plus précisément, pour chaque mention M à lier pour laquelle une chaîne de coréférence C est trouvée par S2E (pour les autres nous gardons la prédiction initiale du module de désambiguïstation), nous évaluons 4 approches de combinaison :

- **Indépendant** : L’entité retenue pour M est celle prédite par le module de désambiguïstation.
- **Vote simple** : L’entité est obtenue par vote entre les entités prédites pour chaque mention de C (en forçant le module de désambiguïstation sur ces mentions). Les égalités sont résolues en faveur de l’entité prédite pour M , ou à défaut priorité est donnée aux premières mentions du texte.
- **Vote pondéré (longueur)** : Similaire au vote simple, mais en pondérant chaque vote par le nombre de tokens de la mention : par exemple "Gonzalez" aura un poids de 1 et "Juan Gonzalez" de 2. L’intuition sous-jacente est que plus l’entité est longue plus elle est informative, donc la prédiction est de base plus faible sans même considérer le contexte. Ce n’est toutefois pas systématique : par exemple "Paris" aura un poids de 1 et "Cette petite ville" de 3.
- **Vote pondéré (position)** : Similaire au vote simple, mais en donnant un poids supérieur (1.5) aux mentions de la première moitié (arrondie à l’inférieur) de la chaîne, où l’information est a priori plus riche. Ainsi, dans l’exemple de la Figure 2, les mentions 1-3 auront un poids 1.5 et les mentions 4-6 un poids 1.

Les résultats présentés en Table 4 montrent un gain modeste (+0.1 en global, +0.3 sur les mentions coréférentes). Dans l’ensemble le fait de combiner ainsi les prédictions apporte donc bien de l’information utile, même si cela pose des questions de significativité statistique, qui mériteraient une exploration empirique plus approfondie. Les heuristiques de pondération expérimentées apparaissent toutefois trop simples pour permettre une réelle émergence de la bonne entité.

	$F1^e$	$F1_C^e$	# Dégradations	# Corrections
Indépendant	72.4	75.7	–	–
Vote simple	72.5	76.0	-21	+25
Vote pondéré (longueur)	72.5	76.0	-21	+25
Vote pondéré (position)	72.5	75.8	-33	+39

TABLE 4 – Effet de plusieurs heuristiques de vote sur la performance de GENRE

Afin de mieux caractériser l’apport des indices issus du reste de la chaîne de coréférence, on mesure également le nombre de prédictions correctes devenues erronées du fait de la combinaison (# Dégradations), et le nombre de prédictions erronées corrigées par cette combinaison (# Corrections). Cette mesure confirme que la pondération par longueur de mention n’a pas d’effet concret, alors que

la priorité au début de la chaîne modifie effectivement le comportement du vote. Il en ressort aussi que les gains modestes résultent d'un nombre important de dégradations qui fait perdre le bénéfice des corrections. Il y a donc là encore une marge importante d'amélioration dans la recherche d'une heuristique de pondération plus fine pour éliminer ces dégradations.

5.3 Discussion

Bien que la méthode montre déjà des gains prometteurs, appuyés autant théoriquement qu'empiriquement, il semble toutefois y avoir un important manque à gagner en se limitant aux prédictions déjà fournies par le module de désambiguïsation. L'une des marges supplémentaires d'amélioration, déjà évoquée ci-dessus, est de renforcer la génération de candidats (pour les modèles qui en utilisent), en tirant parti du fait que chaque mention peut contribuer son lot de candidats au profit de l'ensemble des mentions de la chaîne. Cette amélioration peut aussi être appliquée au module de génération de candidats de manière indépendante (y compris sans combinaison des prédictions), et constitue une piste intéressante pour de futures explorations.

Par ailleurs, il arrive aussi que les indices offerts dans le reste de la chaîne ne soient pas suffisants pris isolément (tous les liages sont erronés) mais qu'une fois combinés ils permettent une désambiguïsation complète de la mention : c'est la configuration 3 de la section précédente. Par exemple dans le cas de la Figure 1, les prédictions pour "Disneyland" et "the Hong Kong park" sont toutes deux fausses, donc une stratégie de sélection parmi les prédictions ne corrigerait pas le liage, pourtant l'apport des indices issus de la chaîne est clair en théorie. Cette intuition oriente vers l'idée d'une combinaison précoce de ces indices, par exemple en combinant directement les représentations contextuelles des mentions (moyenne, somme, max-pooling...). Cette stratégie n'est toutefois applicable qu'à certains types de modèles de désambiguïsation (où les représentations de mentions jouent un rôle prépondérant), or SPEL par exemple ne fait intervenir que les représentations de sous-mots et pas directement des représentations de mentions.

6 Conclusion

Nous avons exploré dans ce travail les interactions possibles entre liage d'entités et résolution de coréférences. Notre analyse à la fois théorique et empirique nous a permis d'identifier plusieurs configurations d'intérêt au sein des chaînes de coréférence, soulignant des difficultés spécifiques que peuvent rencontrer les modules de désambiguïsation sur les mentions coréférentes, mais aussi le bénéfice potentiel à exploiter des indices associés à d'autres mentions de la chaîne, potentiellement moins ambiguës. Sur la base d'expériences oracles confirmant ce bénéfice, nous avons proposé une première preuve de concept pour exploiter ces informations issues des mentions coréférentes, par un mécanisme de vote entre entités prédites dans la chaîne. Nous identifions plusieurs perspectives de prolongement de ces travaux : tout d'abord le recours à la chaîne de coréférence pour compléter la génération d'entités candidates (plutôt qu'améliorer la désambiguïsation uniquement), mais aussi l'option d'une fusion plus précoce de l'information offerte au sein de la chaîne, permettant de couvrir également la configuration où toutes les mentions de la chaîne sont ambiguës mais le tout est univoque. Enfin, les divergences de conventions d'annotation entre liage et coréférence limitent l'applicabilité de la méthode à un tiers des mentions environ (où les deux prédictions coïncident), et il serait intéressant d'explorer des stratégies d'appariement entre mentions des deux modules.

Remerciements

Ce travail a été partiellement financé par l'Agence de l'Innovation de défense, dans le cadre du projet CapiTAL.

Références

- AKBIK A., BLYTHE D. A. J. & VOLLGRAF R. (2018). Contextual string embeddings for sequence labeling. In *International Conference on Computational Linguistics*.
- ANGELL R., MONATH N., MOHAN S., YADAV N. & MCCALLUM A. (2021). Clustering-based inference for biomedical entity linking. In K. TOUTANOVA, A. RUMSHISKY, L. ZETTLEMOYER, D. HAKKANI-TUR, I. BELTAGY, S. BETHARD, R. COTTERELL, T. CHAKRABORTY & Y. ZHOU, Éd.s., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 2598–2608, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.naacl-main.205](https://doi.org/10.18653/v1/2021.naacl-main.205).
- BELTAGY I., PETERS M. E. & COHAN A. (2020). Longformer : The long-document transformer. *CoRR*, **abs/2004.05150**.
- BOHNET B., ALBERTI C. & COLLINS M. (2023). Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, **11**, 212–226. DOI : [10.1162/tacl_a_00543](https://doi.org/10.1162/tacl_a_00543).
- BROSCHET S. (2019). Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In M. BANSAL & A. VILLAVICENCIO, Éd.s., *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, p. 677–685, Hong Kong, China : Association for Computational Linguistics. DOI : [10.18653/v1/K19-1063](https://doi.org/10.18653/v1/K19-1063).
- CAO N. D., IZACARD G., RIEDEL S. & PETRONI F. (2020). Autoregressive entity retrieval. *CoRR*, **abs/2010.00904**.
- CAO N. D., IZACARD G., RIEDEL S. & PETRONI F. (2021). Autoregressive entity retrieval. In *International Conference on Learning Representations*.
- DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In J. BURSTEIN, C. DORAN & T. SOLORIO, Éd.s., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423).
- GANEVA O. & HOFMANN T. (2017). Deep joint entity disambiguation with local neural attention. *CoRR*, **abs/1704.04920**.
- HAJISHIRZI H., ZILLES L., WELD D. S. & ZETTLEMOYER L. (2013). Joint coreference resolution and named-entity linking with multi-pass sieves. In D. YAROWSKY, T. BALDWIN, A. KORHONEN, K. LIVESCU & S. BETHARD, Éd.s., *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 289–299, Seattle, Washington, USA : Association for Computational Linguistics.
- HOCHREITER S. & SCHMIDHUBER J. (1997). Long Short-Term Memory. *Neural Computation*, **9**(8), 1735–1780. DOI : [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- HOFFART J., SUCHANEK F. M., BERBERICH K. & WEIKUM G. (2013). Yago2 : A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelli-*

gence, **194**, 28–61. Artificial Intelligence, Wikipedia and Semi-Structured Resources, DOI : <https://doi.org/10.1016/j.artint.2012.06.001>.

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 R. BARZILAY & M. JOHNSON, Édts., *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 782–792, Edinburgh, Scotland, UK. : Association for Computational Linguistics.

JOSHI M., CHEN D., LIU Y., WELD D. S., ZETTLEMOYER L. & LEVY O. (2019). Spanbert : Improving pre-training by representing and predicting spans. *CoRR*, **abs/1907.10529**.

KIRSTAIN Y., RAM O. & LEVY O. (2021). Coreference resolution without span representations. In *Annual Meeting of the Association for Computational Linguistics*.

KOLITSAS N., GANEA O.-E. & HOFMANN T. (2018). End-to-end neural entity linking. In A. KORHONEN & I. TITOV, Édts., *Proceedings of the 22nd Conference on Computational Natural Language Learning*, p. 519–529, Brussels, Belgium : Association for Computational Linguistics. DOI : [10.18653/v1/K18-1050](https://doi.org/10.18653/v1/K18-1050).

LE P. & TITOV I. (2018). Improving entity linking by modeling latent relations between mentions. *CoRR*, **abs/1804.10637**.

LEE K., HE L., LEWIS M. & ZETTLEMOYER L. (2017). End-to-end neural coreference resolution. In M. PALMER, R. HWA & S. RIEDEL, Édts., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 188–197, Copenhagen, Denmark : Association for Computational Linguistics. DOI : [10.18653/v1/D17-1018](https://doi.org/10.18653/v1/D17-1018).

LEE K., HE L. & ZETTLEMOYER L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In M. WALKER, H. JI & A. STENT, Édts., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, p. 687–692, New Orleans, Louisiana : Association for Computational Linguistics. DOI : [10.18653/v1/N18-2108](https://doi.org/10.18653/v1/N18-2108).

LEWIS M., LIU Y., GOYAL N., GHAZVININEJAD M., MOHAMED A., LEVY O., STOYANOV V. & ZETTLEMOYER L. (2019). BART : denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, **abs/1910.13461**.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized BERT pretraining approach. *CoRR*, **abs/1907.11692**.

MRINI K., NIE S., GU J., WANG S., SANJABI M. & FIROOZ H. (2022). Detection, disambiguation, re-ranking : Autoregressive entity linking as a multi-task problem. In S. MURESAN, P. NAKOV & A. VILLAVICENCIO, Édts., *Findings of the Association for Computational Linguistics : ACL 2022*, p. 1972–1983, Dublin, Ireland : Association for Computational Linguistics. DOI : [10.18653/v1/2022.findings-acl.156](https://doi.org/10.18653/v1/2022.findings-acl.156).

PENNINGTON J., SOCHER R. & MANNING C. (2014). GloVe : Global vectors for word representation. In A. MOSCHITTI, B. PANG & W. DAELEMANS, Édts., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543, Doha, Qatar : Association for Computational Linguistics. DOI : [10.3115/v1/D14-1162](https://doi.org/10.3115/v1/D14-1162).

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

PICCINNO F. & FERRAGINA P. (2014). From TagME to WAT : a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation - ERD '14*, p. 55–62, Gold Coast, Queensland, Australia : ACM Press. DOI : [10.1145/2633211.2634350](https://doi.org/10.1145/2633211.2634350).

- PRADHAN S., MOSCHITTI A., XUE N., NG H. T., BJÖRKE LUND A., URYUPINA O., ZHANG Y. & ZHONG Z. (2013). Towards robust linguistic analysis using OntoNotes. In J. HOCKENMAIER & S. RIEDEL, Édts., *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, p. 143–152, Sofia, Bulgaria : Association for Computational Linguistics.
- RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, **abs/1910.10683**.
- RÖDER M., USBECK R. & NGONGA NGOMO A.-C. (2018). Gerbil–benchmarking named entity recognition and linking consistently. *Semantic Web*, **9**(5), 605–625.
- SANG E. F. T. K. & MEULDER F. D. (2003). Introduction to the conll-2003 shared task : Language-independent named entity recognition. *CoRR*, **cs.CL/0306050**.
- SHAVARANI H. & SARKAR A. (2023). SpEL : Structured prediction for entity linking. In H. BOUAMOR, J. PINO & K. BALI, Édts., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, p. 11123–11137, Singapore : Association for Computational Linguistics. DOI : [10.18653/v1/2023.emnlp-main.686](https://doi.org/10.18653/v1/2023.emnlp-main.686).
- VAN HULST J. M., HASIBI F., DERCKSEN K., BALOG K. & DE VRIES A. P. (2020). REL : an entity linker standing on the shoulders of giants. *CoRR*, **abs/2006.01969**.
- VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need. *CoRR*, **abs/1706.03762**.
- WU W., WANG F., YUAN A., WU F. & LI J. (2020). CorefQA : Coreference resolution as query-based span prediction. In D. JURAFSKY, J. CHAI, N. SCHLUTER & J. TETREAU LT, Édts., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 6953–6963, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.acl-main.622](https://doi.org/10.18653/v1/2020.acl-main.622).
- YAMADA I., SHINDO H., TAKEDA H. & TAKEFUJI Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. In S. RIEZLER & Y. GOLDBERG, Édts., *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, p. 250–259, Berlin, Germany : Association for Computational Linguistics. DOI : [10.18653/v1/K16-1025](https://doi.org/10.18653/v1/K16-1025).
- ZAPOROJETS K., DELEU J., DEVELDER C. & DEMEESTER T. (2020). DWIE : an entity-centric dataset for multi-task document-level information extraction. *CoRR*, **abs/2009.12626**.
- ZAPOROJETS K., DELEU J., JIANG Y., DEMEESTER T. & DEVELDER C. (2022). Towards consistent document-level entity linking : Joint models for entity linking and coreference resolution. In S. MURESAN, P. NAKOV & A. VILLAVICENCIO, Édts., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 778–784, Dublin, Ireland : Association for Computational Linguistics. DOI : [10.18653/v1/2022.acl-short.88](https://doi.org/10.18653/v1/2022.acl-short.88).
- ZHANG W., HUA W. & STRATOS K. (2021). Entqa : Entity linking as question answering. *CoRR*, **abs/2110.02369**.
- ZHANG W., WISEMAN S. & STRATOS K. (2023). Seq2seq is all you need for coreference resolution.