L’annotation sémantique

Mohamed CHABCHOUB

Plan

1) Annotation Sémantique
2) Dbpedia Spotlight
3) AIDA
4) Conclusion

1. Annotation sémantique

L'Annotation sémantique est une tâche de fouille de texte proche des méthodes de traitement automatique des langues qui consiste à étiqueter dans un document les mots avec des liens qui pointent vers une description sémantique.

Défis (désambiguisation) :

Un des plus grands défis de l'annotation est l'ambiguïté. Le nom d'une entité peut être utilisé dans différents contexte et se référer à différents contextes sur DBPedia. Par exemple si on retrouve le nom de surface (un nom de surface est un mot ou un ensemble de mots, qui est un possible candidat à l'annotation) Washington, il peut se référer aux ressources DBPedia:George_Washington, dbpedia:Washington,_D.C. et dbpedia:Washington_(U.S._state). Pour les humains, la désambiguïsation qui est le choix entre plusieurs candidats pour un mot ambigu est basée sur les connaissances du lecteur et le contexte actuel. Cependant la désambiguïsation est un problème difficile.

2) Dbpedia Spotlight :

C’est un système permettant de connecter les documents présents sur le web avec le Linked Open Data. Et cela en effectuant des annotations de textes en entrés par des URI pointant sur des ressources en DBpedia.

Figure 1 dbpedia Spotlight Demo

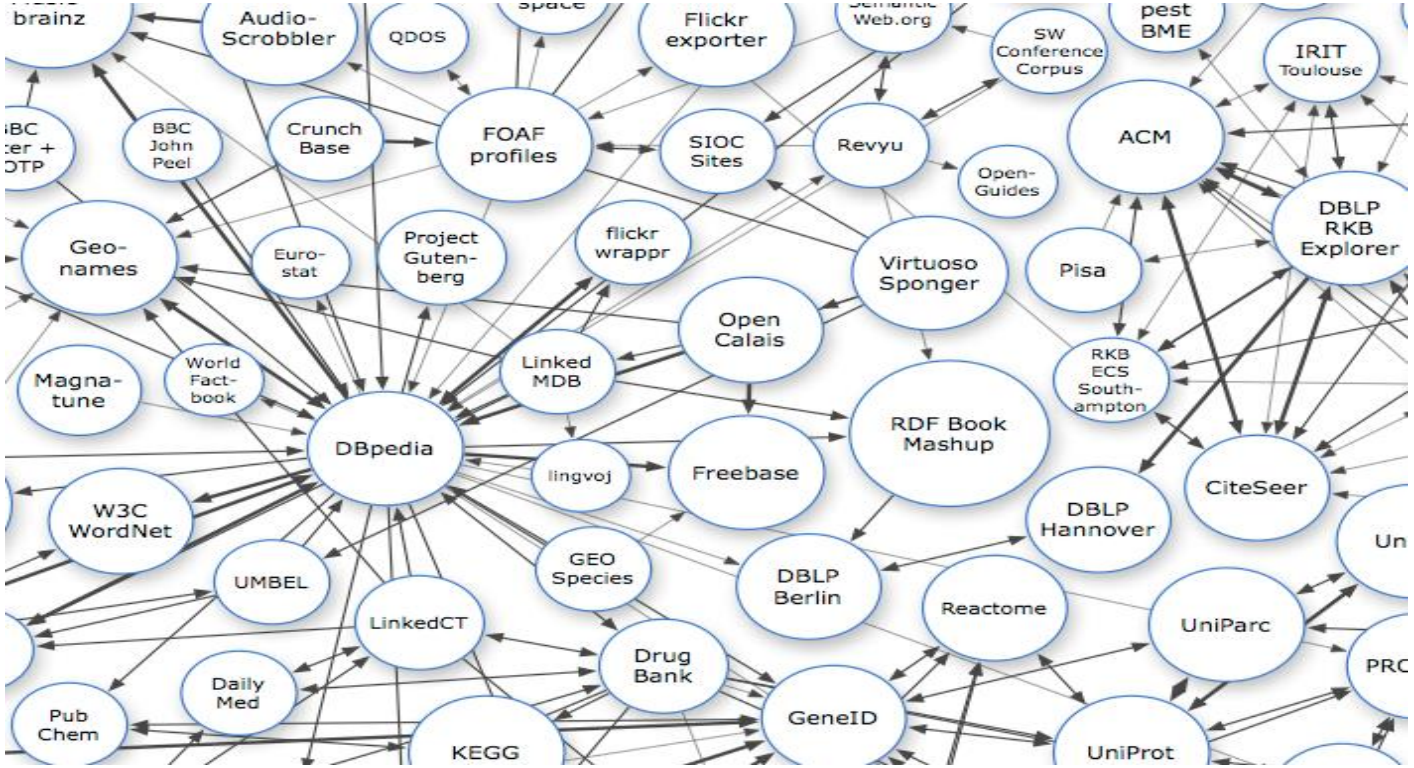


Figure 2 Linked Open Data

Dbpedia:

Base de connaissance contenant des données structurés extraites à partir de Wikipedia.
Elle contient environ 3.5 millions de ressources. Contient des relation comme la « owl :sameas » pour référer à d'autres bases de connaissances. Maintenant DBpedia représente le centre du Linked Open Data

Dataset:

Construction d'un graphe de label, en commençant par les noms de surfaces qui sont les titres des pages de wikipedia, les reliants avec les Redirects qui sont les URIs indiquant des synonymes ou des noms de surfaces alternatives. Qui sont par la suite reliés Les désambiguïsations fournissent les noms de surfaces ambigus qui peuvent être confus avec les ressources auxquelles elles sont liées.

Approche:

1-Repérage (Spotting)

L'ensemble des étiquettes a été utilisé comme lexique pour la phase de spotting. L'implémentation utilisée est le *LingPipe Exact Dictionary-Based Chunker* basée sur l'algorithme de correspondance de chaîne de caractères de Aho-Corasick.

Le système ignore les spots qui se composent seulement verbes, adjectives, adverbes ou prépositions. Le part of speech Tagger utilisé est l'implémentation de LingPipe basé sur le modèle de Hidden Markov.

2-Candidats

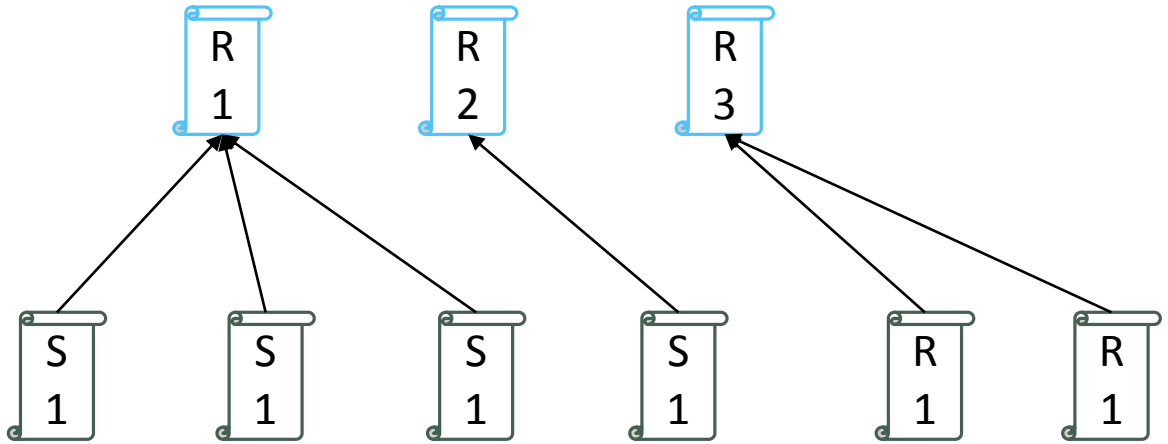
En utilisant la lexicalisation de dbpedia. Il relie chaque nom de surface à des ressources candidates.

3-Désambiguïsation

-Métrique

- 1er métrique utilisé : probabilité à priori

$P(r|s) = n(s, r)/n(s)$



Mais cette métrique seule ne suffit pas pour permettre la désambiguïsation, il faut prendre en considération le contexte

❑ 2ème métrique tf :icf

-Le système utilise le paragraphe comme contexte pour les noms de surface

La modélisation des occurrences des ressources de dbpedia dans un vecteur (VSM) ou chaque ressource pointe à un espace multidimensionnel de mots

TF : Dans DBPedia Spotlight, il représente l'importance d'un mot pour une ressource donnée.

IDF : (Inverse Document Frequency) représente l'importance générale d'un mot dans une collection de ressource.

Cette métrique, bien qu'efficace dans le domaine de la recherche d'information, ne l'est pas pour la désambiguïsation parce qu'elle capture très bien l'importance global d'un mot (pour toutes les ressources) mais ne parvient pas à saisir l'importance d'un mot pour un ensemble spécifique de ressources candidats. Mais dans notre cas elle n'aide pas pour connaitre l'importance d'un mot dans un nombre de ressources restreint.

IDF ⟹ ICF

Soit Rs l'ensemble des ressources candidates pour un nom de surface s et n(wj) le nombre de ressources en Rs associé au mot wj.

$$ICF(w_j) = \log \frac{|R_s|}{n(w_j)} = \log |R_s| - \log n(w_j)$$

Avec la représentation VSM des ressources DBPedia et les poids associés TF*ICF, la désambiguïsation peut être vu comme un problème de classement ou l'on peut trouver la ressource qui se trouve à la première position. L'approche de DBPedia spotlight est de classer les ressources candidates à partir de leur score de similarité entre leur contexte et le contexte autour du nom de surface.

Test

Test sur 155,000 wikilink

$$Mixed(r, s, C) = 1234.3989 * P(r|s) + 0.9968 * contextualScore(r, s, C) - 0.0275$$

Disambiguation Approach	*Accuracy*
Baseline Random	17.77%
Baseline Default Sense	55.12%
Baseline TF*IDF	55.91%
DBpedia Spotlight TF*ICF	73.39%
DBpedia Spotlight Mixed	80.52%

Tableau 1 Précision des approches

Limites:

DBpedia Spotlight fait la désambiguïsation de chaque mot tout seul ce qui ne donne pas de résultats préférable si le texte en input est court . comme par exemple un texte parlant de football entre Madrid et Manchester qui se déroule dans Barcelone, le système peut lier les 3 noms de surfaces pour trois villes ou les trois équipes.

3) AIDA

-Système pour annotation sémantique

-Des liens vers Wikipedia

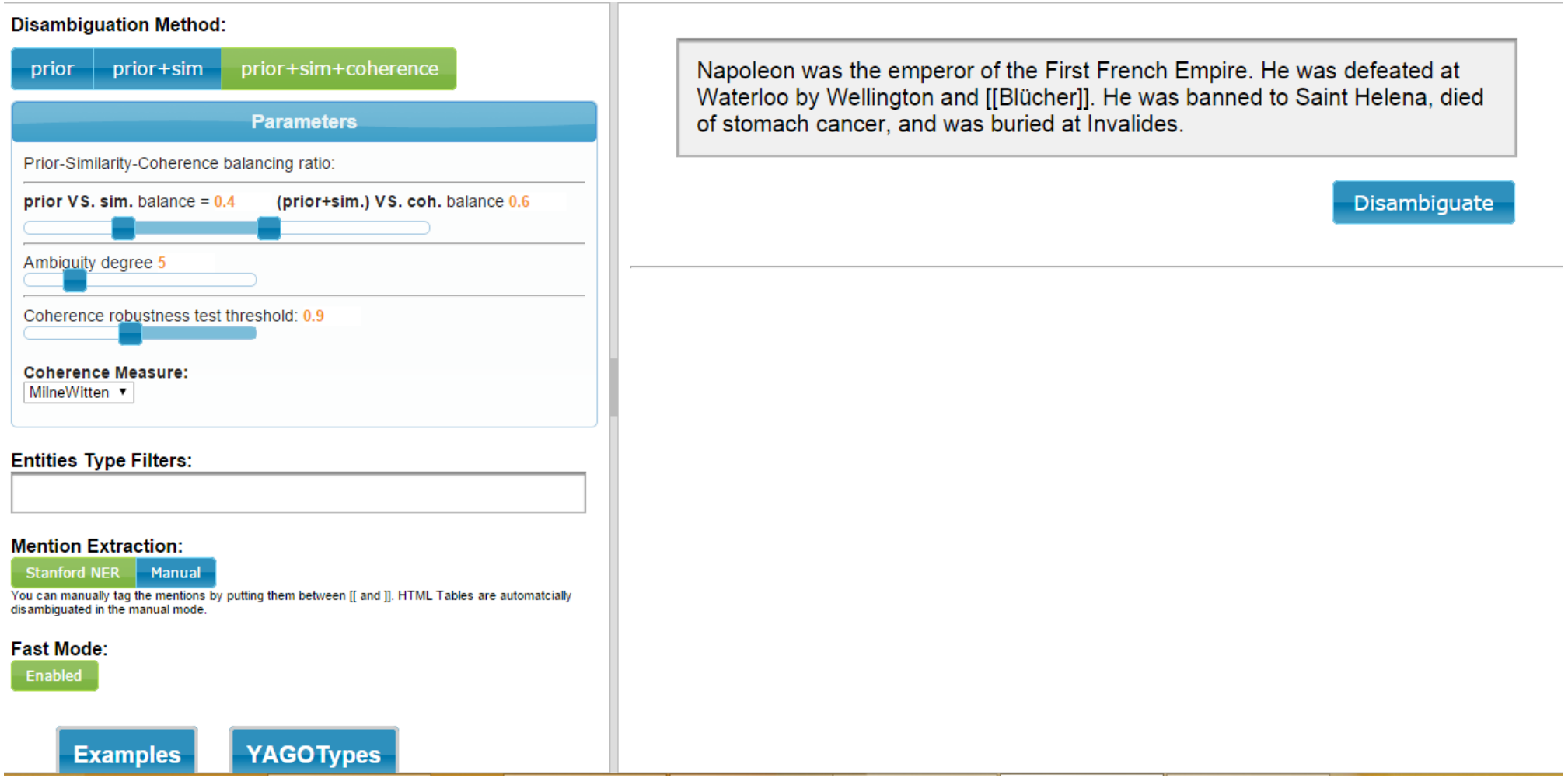


Figure 3 AIDA

Approche

1-Le spotting

Il utilise Stanford NER tagger pour identifier les formes de surfaces dans le textes en input

2-Candidates

La base de connaissances YAGO2

3-Désambiguïsation

-Métriques

- ❑ La probabilité à priori (comme Dbpedia Spotlight)

- ❑ Similitude

Du coté des noms de surface on représente le contexte comme l'ensemble de tous les mots du texte en entrée. On pourrait aussi utiliser des scores de similitude basés sur l'analyse sémantique du texte comme par exemple savoir qu'un certain verbe est associé le plus souvent des fois à une entité
Du coté des entités, on associe à chaque entité un ensemble de mots clés pré-calculés des articles de Wikipédia. On peut alors calculer la similitude entre une entité et un nom de surface en utilisant des techniques statistiques comme la divergence, la cooccurrence des mots etc…

❑ Cohérence

Le calcule se base sur le nombre de inlink que les articles des 2 entités se partagent

$$mw\_coh(e_1, e_2) = \quad 1 - \frac{\log\left(\max(|IN_{e_1}|, |IN_{e_2}|)\right) - \log(|IN_{e_1} \cap IN_{e_2}|)}{\log(|N|) - \log\left(\min(|IN_{e_1}|, |IN_{e_2}|)\right)}$$

$$\text{if} > 0 \text{ and else set to } 0.$$

IN : représente l'ensemble des inlinks pour chaque entité

N: le nombre totale dans la collection (Wikipedia)

Formule Principale

$$\alpha \cdot \sum_{i=1..k} \text{prior}(m_i, e_{j_i}) + \quad \beta \cdot \sum_{i=1..k} \text{sim}(\text{cxt}(m_i), \text{cxt}(e_{j_i})) + \quad \gamma \cdot \text{coh}(e_{j_1} \in \text{cnd}(m_1) \ldots e_{j_k} \in \text{cnd}(m_k)) = \max!$$

$$\alpha + \beta + \gamma = 1$$

$$\alpha = 0.43, \beta = 0.47, \gamma = 0.10$$

Modèles de graphe et algorithme

-Graphe

On construit un graphe pesé qui a comme nœuds les noms de surfaces et les entités candidates. Ce graphe a deux types d'arcs :

Des arcs entre les noms de surface et les entités. Ces arcs ont un poids qui représente la similarité

Des arcs entité-entité ou le poids des arcs représente la cohérence

Ce graphe est dense pour les arcs nom de surface-entité car en générale les bases de connaissance fournissent un vaste ensemble de résultats quand on effectue une requête sur un nom donnée (l'ordre de grandeur est de centaine de milliers de nœuds pour un nom de surface).

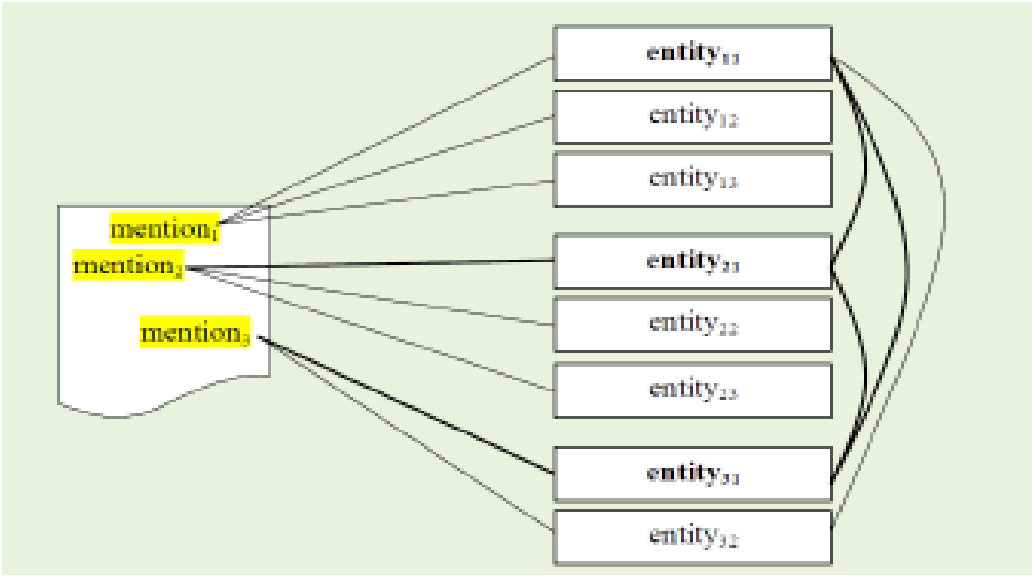


Figure 4 Graphe de désambiguïsation

A partir du graphe défini, on veut calculer un sous-graphe qui idéalement contiendra tous les nœuds des noms de surface et exactement un arc nom de surface-entité pour chaque nom de surface, désambiguïsant ainsi chaque nom de surface.
Cette approche doit surmonte deux défis principaux :

- Fournir évidemment la meilleure désambiguïsation possible et donc il faut trouver une notion de densité qui capture le mieux la cohérence du résultat. L'approche plus naturelle serait de mesurer la densité d'un sous graphe en fonction de la somme des poids arcs mais cette approche ne fonctionne pas pour la désambiguïsation. Pour ce, il définisse le weighted degree (WD) d'un nœud comme la somme des poids des arcs incidents. Ils ont définit alors la densité d'un sous-graphe comme le WD minimum parmi les nœuds. Le but est donc de calculer le sous graphe avec la plus grande densité.
- Le deuxième problème critique à affronter est la complexité computationnelle. Les problèmes du sous-graphe sont en générale presque inévitablement NP-HARD.
  Pour résoudre donc le problème, on adopte une version étendu de l'algorithme pour trouver des groupes fortement connectés, limité en taille d'un réseau sociale. L'algorithme commence à partir d'un nœud et itérativement enlève le nœud entité avec le plus petit WD. Parmi tous les sous-graphes obtenus à chaque itération, on retourne celui qui maximise le plus petit WD (la densité).

-Pour garantir que la solution trouvée soit concrète, ils ont ajouté la contrainte que chaque nom de surface soit connecté à au moins une entité. Cette contrainte pourrait emmener à des solutions pas très optimales. Pour ce, ils ont fait un prétraitement pour éliminer les entités qui sont trop distantes des noms de surface. Pour chaque nœud entité, il calcule sa distance de l'ensemble nœuds de surface. Comme métrique, ils utilisent la somme des carrés du plus cours chemin entre une entité et un nom de surface.

Ils ont restreint ainsi le graph en input à seules les entités plus proches des noms de surface.

Empiriquement, ils ont déterminé qu'un bon choix de la taille des entités est 5 fois la taille des noms de surface. A partir d'ici, le graphe est assez petit, ils procèdent à une énumération des solutions pour choisir la meilleur. Comme alternative plus rapide ils ont utilisé un algorithme probabiliste qui choisit de manière aléatoire les entités associés à un nom de surface. On

La répétition de l'algorithme plusieurs fois et il retourne la meilleure parmi les solutions obtenues.

Test:
news (CoNLL), difficiles et courts contexts (WP), Web pages (Wiki-links), et articles Wikipedia.

Dataset	AIDA	Spotlight
CoNLL-YAGO	82.5%*	75.0%
WP	**84.7%***	63.8%
Wikipedia articles	**90.0%**	89.6%
Wiki-links	80.3%	80.7%

Tableau 2 Comparaison AIDA Spotlight

Limites :
Les résultats de l'approche de centralité de graphe ont été comparés à ceux de l'approche de DBPedia Spotlight, et il s'est avéré que l'approche de centralité de graphe à de meilleurs résultats. Toutefois l'approche de désambiguïsation robuste combine plusieurs métriques et résout le problème avec une optique d'optimum globale. Il donne donc de meilleurs résultats si le texte en entrée est cohérant, mais il faudra faire attention lorsque le texte parle de plusieurs concepts hétérogènes.

Conclusion

-L’annotation sémantique être un premier pas pour transformer le web syntaxique actuelle en web sémantique.
-Toutefois, le mappage automatique de nom de surface à des entités est un problème difficile à cause de désambiguïsation.
-Il existe des approches basées toutes sur des heuristiques.

- Un futur travail de recherche pourrait être celui de trouver quelles caractéristiques un texte devrait avoir, et en fonction de ces caractéristiques lui associer la meilleure heuristique possible.

Références

- Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen F¨urstenau,Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, Gerhard Weikum.(2011) ‘’ Robust Disambiguation of Named Entities in Text ”, Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 782–792.

- Pablo N. Mendes, Max Jakob, Andrés García-Silva, Christian Bizer(2011). ‘’ DBpedia Spotlight: Shedding Light on the Web of Documents ”, I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria.

- Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, Gerhard Weikum, ‘’AIDA-light:High-Throughput Named-Entity Disambiguation’’, LDOW2014, April 8, 2014, Seoul, Korea.