TALN 2011, Montpellier, 27 juin – 1er juillet 2011

Mesure non-supervisée du degré d'appartenance d'une entité à un type

Ludovic Bonnefoy[1,2], Patrice Bellot[1], Michel Benoit[2]
(1) Université d'Avignon - CERI/LIA, Agroparc – B.P. 1228, 84911 Avignon Cedex 9
(2) iSmart, Le Mercure A, 13851 Aix-en-Provence Cedex 3
patrice.bellot@univ-avignon.fr, {ludovic.bonnefoy,michel.benoit}@ismart.fr

Résumé. La recherche d'entités nommées a été le sujet de nombreux travaux. Cependant, la construction des ressources nécessaires à de tels systèmes reste un problème majeur. Dans ce papier, nous proposons une méthode complémentaire aux outils capables de reconnaître des entités de types larges, dont l'objectif est de déterminer si une entité est d'un type donné, et ce de manière non-supervisée et quel que soit le type. Nous proposons pour cela une approche basée sur la comparaison de modèles de langage estimés à partir du Web. L'intérêt de notre approche est validé par une évaluation sur 100 entités et 273 types différents.

Abstract. Searching for named entities has been the subject of many researches. In this paper, we seek to determine whether a named entity is of a given type and in what extent it is. We propose to address this issue by an unsupervised Web oriented language modeling approach. The interest of it is demonstrated by our evaluation on 100 entities and 273 different types.

Mots-clés : typage d'entités nommées, comparaison de distribution de mots, divergence de Kullback-Leibler.

Keywords: named entity identification, language modeling approach, Kullback-Leibler divergence.

1 Introduction

Depuis les années 1990, les entités nommées sont au centre de nombreux travaux en traitement de la langue naturelle écrite (résumé automatique, ontologies, ...). Un tel développement est, en grande partie, dû à l'impulsion donnée par de multiples campagnes d'évaluation, qui ont accordé une part importante à leur identification et utilisation au sein de leurs pistes tels que MUC (*Named Entity task* [1]), TREC (avec la tâche *Question Answering* (Voorhees, 1999))...

En l'absence de corpus d'apprentissage, les premières méthodes de recherche d'entités nommées, se basaient sur l'utilisation de larges ensembles de patrons d'extraction (Nadeau & Sekine, 2007) et aujourd'hui encore il est conseillé de procéder de la sorte si un corpus d'entraînement n'est pas disponible pour les types souhaités (Sekine & Nobata, 2004). Lorsque les premiers corpus d'apprentissage pour certains types (personne, lieu, organisation et date) firent leur apparition, la plupart des méthodes d'apprentissage automatique furent utilisées pour ce problème telles que les modèles de Markov cachés (Bikel *et al.*, 1997), les arbres de décision (Sekine, 1998) ou encore les SVMs (Asahara & Matsumoto, 2003) et les CRFs (McCallum, 2003). Des méthodes dites semi-supervisées ont aussi été étudiées telle le *bootstrapping* qui consiste à démarrer d'un petit jeu d'exemples et de l'agrandir par itérations successives en ayant recours à divers critères comme les relations syntaxiques (Cucchiarelli & Velardi, 2001) ou synonymiques (Pasca *et al.*, 2006).

La reconnaissance des entités nommées est centrale dans bon nombre de problématiques en recherche d'information comme par exemple Questions-Réponses (QR). Cette tâche a connu un fort engouement ces dernières années. En effet, on a pu voir plusieurs campagnes d'évaluation internationales en faire un sujet important (TREC, CLEF, INEX, Equer, ...). Un système QR présente au moins deux différences par rapport à un système de recherche d'information (RI). La première est la formulation de la requête qui est une phrase interrogative en langage naturel (par exemple *"Je veux connaître les spécifications techniques du nouveau Blackberry"*). Cela a de l'intérêt pour les utilisateurs (la formulation de requêtes efficaces sous forme de mots clés est une tâche difficile) et pour les systèmes (apport d'un contexte et d'informations supplémentaires). La seconde principale différence est la forme

1. http ://cs.nyu.edu/faculty/grishman/NEtask20.book_1.html

des résultats : un moteur de RI va retourner une liste de documents, dans lesquels l'utilisateur va être en charge de trouver la réponse par lui même, tandis qu'en QR, le système doit retourner une série de réponses précises (c'est-à-dire des chaînes correspondant exactement à ce que l'utilisateur recherche), généralement des entités nommées. C'est pourquoi une identification correcte (localisation et typage) des entités nommées est une étape vitale.

La principale barrière à l'utilisation des méthodes évoquées précédemment (patrons d'extraction et méthodes d'apprentissage) est la constitution des ressources propres à chaque domaine d'application. En effet, les méthodes supervisées apprennent différents paramètres sur des corpus annotés manuellement, où chaque entité d'un des types souhaités est relevée. On comprend que lorsque le nombre de types à reconnaître augmente, le temps d'annotation de tels corpus devient rédhibitoire. De manière similaire, la création de patrons d'extraction de plus en plus précis et difficile à maintenir. Dans cet article, nous proposons une méthode non-supervisée complémentaire à ces outils, ayant pour but de déterminer si une entité (dans son sens le plus large, c'est-à-dire toute réponse qu'un système de questions-réponses pourrait être amené à devoir retrouver) est d'un type donné et à quel degré.

L'objectif de cette méthode est de mesurer la proximité d'un entité et d'un type mais aussi de deux entités entre elles. Cette problématique est intéressante comme l'atteste la création en 2009 de la piste *Entity Relation Finding* à TREC, sa poursuite en 2010 et 2011 [2].

Un des points les plus importants, est d'arriver à traiter ce problème pour n'importe quel type d'entités et pas seulement les quelques types de très haut niveau (personnes, lieux, organisation, dates,...) que l'on a l'habitude de rencontrer depuis les campagnes MUC (Nadeau & Sekine, 2007) ou les quelques dizaines de types plus fins (c'est-à-dire des sous catégories des types de haut niveau (Sekine *et al.*, 2002)) qu'exploitent certains systèmes. Notre objectif est de pouvoir traiter de manière égale et automatique des types généraux tels que "espèce animale" ou beaucoup plus fins, tels que "coéquipier" ou encore "distilleries de whisky".

Les applications d'une telle méthode sont multiples. Tout d'abord la validation d'un type attribué à une entité pouvant servir à éliminer par exemple des réponses candidates dans un système de question-réponse. Un deuxième exemple d'application serait la construction automatique de lexique d'entité nommées ou le peuplement automatique d'ontologies (du moins pour les relations is-A). Enfin, l'on pourrait envisager par exemple, une aide à l'annotation semi-manuelle d'entité nommées avec des types sémantiques fins, où l'utilisateur sélectionnerait les types corrects parmiles premiers types retournés par la méthode.

Cet article est composé de la manière suivante : dans une première partie, nous présentons une solution pour mesurer le degré d'appartenance d'une entité nommée à un type donné de manière non-supervisée. Dans la seconde partie, nous proposons un cadre d'évaluation et discutons les résultats obtenus par notre proposition.

2 Mesure de la proximité sémantique d'une entité et d'un type donné

Comme nous l'avons évoqué plus haut, nous aspirons ici à une méthode efficace pour déterminer si une entité est d'un type donné sans apprentissage au préalable, afin de se passer de corpus coûteux et limitant le nombre de types que l'on peut traiter. C'est pourquoi nous avons opté pour une approche orientée Web.

Ce travail part de plusieurs constats formulés après une analyse manuelle des pages Web retournées par des moteurs de recherche du Web lorsqu'on les interroge avec des entités ou leurs types. Nous nous sommes aperçu que les pages associées à chaque type avaient tendance à posséder un vocabulaire spécifique, c'est-à-dire que certains mots avaient un nombre d'occurrences largement supérieur à celui qui est le leur dans un corpus générique (ensemble très large de documents, traitant de toutes sortes de sujets) et qu'au contraire, certains mots n'apparaissent pas ou peu. Par exemple, pour le type *"rasoir électrique"*, certains mots comme *"rasoir", "autonomie", "tondeuse", "rasage"*... sont très fréquents dans les pages Web retournées par le moteur de recherche alors qu'ils sont plutôt rares dans un corpus générique.

En étudiant les pages Web associées à des entités, nous avons vérifié que, pour chacune, l'on obtenait des probabilités d'apparition des mots généralement éloignées de celles que l'on trouve dans un corpus générique (par exemple pour *iPod* certains mot comme *"apple","mp3","musique","écouteurs","media", ...* ont une probabilité d'apparition élevée).

Notre dernière observation est que la distribution des probabilités d'apparition des mots, dans les pages associées à une entité donnée, est proche de celle des mots dans les pages Web associées au *type* la caractérisant le plus

2. http ://ilps.science.uva.nl/trec-entity/2010/11/plans-for-entity-2011/

(par exemple, pour un *"Philips HQ 6990/33"* on a un ensemble de mots comme *"Philips"*,*"rasoir"*,*"électrique"*, *"confortable"*,*"rasage"*...qui ont une fréquence élevée et qui est proche de l'ensemble de mots récupéré pour *"rasoir électrique"*).

L'idée de la méthode que nous avons mise en œuvre découle de ces observations. Elle consiste à comparer le modèle de langage L_E (c'est-à-dire la distribution de probabilité des mots dans une collection) associé à une entité donnée à un modèle de langage L_{type} associé à un type d'entité donné.

Tout d'abord il faut collecter deux ensembles de documents, un premier lié à une entité (ex : *"Isaac Asimov"*) et un second correspondant au type que l'on veut tester (ex : *"science-fiction writers"*). Ces documents seront ici des pages Web récupérées en interrogeant un moteur de recherche. Ensuite, calculer la distribution de probabilité des mots pour chacun de ces deux ensembles et les lisser avec le lissage de Dirichlet.

$$p'(w|E) = \begin{cases} p_s(w|E) \text{ si w est présent dans un ensemble de pages Web } E \\ \alpha_d p(w|C) \text{ sinon} \end{cases} \quad (1)$$

où $p'(w|E)$ est la probabilité du mot w dans un ensemble de pages Web E, $p_s(w|E)$ est la probabilité lissée de w, $p(w|C)$ est la probabilité de w dans une collection C (consiste ici en 10% du corpus ClueWeb09B [3] soit environ 5 millions de pages Web choisies aléatoirement). La probabilité $p(w|C)$ est lissée avec un lissage de Laplace (donne un nombre d'occurences de 1 à un mot non présent et rajoute 1 à un mot présent) et α_d est un facteur. $p_s(w|E)$ et α_d sont estimées de la manière suivante :

$$p_s(w|E) = \frac{tf(w,E) + \mu.p(w|C)}{\sum_{w' \in V} tf(w',E) + \mu} \qquad \alpha_d = \frac{\mu}{\sum_{w \in V} tf(w,E) + \mu} \quad (2)$$

où $tf(w,E)$ est le nombre d'occurrences du mot w dans l'ensemble E, V est l'ensemble des mots w' présents dans E et μ est un facteur dont la valeur est empiriquement fixée à 2000 (valeur choisie par (Chen & Goodman, 1998) pour de larges collections journalistiques).

L'ultime étape est la comparaison des probabilités p'_E d'apparition des mots dans les pages Web associées à l'entité à celles p'_{type} des mots dans les pages Web relatives au type. Pour cela nous calculons la divergence de *Kullback-Leibler* (KLD) entre les deux modèles :

$$KLD(E,type) = \sum_i p'_E(i).log\frac{p'_E(i)}{p'_{type}(i)} \quad (3)$$

où $KLD(E,type)$ est la divergence de Kullback-Leibler pour une entité E et un type donné, $p'_E(i)$ (resp. $p'_{type}(i)$) est la probabilité d'apparition du i^e mot dans les documents associés à l'entité E (resp. au $type$).

Cette méthode propose ainsi une manière de calculer le degré d'appartenance de n'importe quelle entité donnée à n'importe quel type donné.

3 Résultats

Il n'existe que très peu de travaux qui se proposent de mesurer le degré d'appartenance d'une entité à n'importe quel type (Pasca, 2004) (Talukdar & Pereira, 2010). Cela a pour conséquence, qu'il n'existe pas à notre connaissance de jeux de données de référence dédiées à l'évaluation de cette tâche et qui soient disponibles. Pour cette raison nous avions décidé de déterminer l'intêret de notre approche indirectement, en mesurant son impact dans un système de type QR. Pour cela nous avions participé à la tâche Entity à Trec 2010. Bien que les résultats présentés dans (Bonnefoy *et al.*, 2011) aient montré une amélioration, il fut difficile d'en mesurer exactement sa responsabilité tant le nombre de paramètres à prendre en compte dans de tels systèmes est important. Dans cet article, afin d'avoir une évaluation de la qualité intrinsèque de notre proposition, nous avons construit un jeu d'évaluation. Nous avons pour cela utilisé DBPedia qui met à disposition un grand nombre d'entités associées à plusieurs types (spécifiés dans différentes ontologies). Nous avons collecté 100 différentes entités (aléatoirement) de cette base ainsi que les types qui leur sont associés et définis dans l'ontologie légère *owl*. Cette ontologie définit 273

3. http ://boston.lti.cs.cmu.edu/Data/clueWeb09/

types différents, ainsi que leurs liens entre eux [4]. Par ce biais il est possible d'associer de 1 à 4 types (profondeur maximum de l'ontologie) par entité, pour une moyenne de 2,83 types.

Avec ces éléments, l'évaluation consiste à classer les 273 types, pour chaque entité, en fonction de son degré d'appartenance à chacun d'entre eux (les plus pertinents en haut du classement). Trois mesures nous ont semblé intéressantes. La première est la précision à 1 (P@1) qui va permettre d'évaluer la capacité du système à ramener un type correct en première position. La seconde est la *moyenne des réciproques des rangs* [5] (MRR) qui est l'inverse du rang moyen auquel le premier élément correct est retrouvé. La dernière est la *précision interpolée pour un rappel de 1* (iPR1) qui permet d'avoir le rang moyen auquel tous les éléments corrects ont été ramenés.

L'évaluation de notre contribution va se faire au regard de deux différentes "baselines". La première, très simple et qui fera office de borne inférieure, consiste à trier les 273 types pour chaque entité, de manière aléatoire. Cependant un seul classement aléatoire pourrait ne pas être représentatif, c'est pourquoi nous avons décidé que plutôt que d'effectuer un tel classement, nous mesurerions la probabilité d'obtenir avec des tirages aléatoires un classement aussi "bon" que le meilleur des notres. Ceci peut être estimé avec la fonction de répartition d'une fonction géométrique. Elle représente la somme des probabilités d'avoir au minimum autant de bonnes réponses en x tirages aléatoires (donc au rang x). Cette "baseline" sera désignée à partir de ce point sous le nom *Aléatoire*.

La deuxième baseline consiste en l'utilisation de patrons d'extractions spécifiquements étudiés pour déterminer les relations is-A entre une entité et un type ((Hearst, 1992), (Pasca, 2004),...) comme *"type such as entité"*. L'idée de la baseline est de compter le nombre de pages Web retrouvées par un moteur de recherche lorsque l'on lui soumet la requête *"type * entité"*, qui signifie que nous recherchons tous les documents dans lesquels on retrouve le type suivi de l'entité et séparés au maximum par un mot (*"including"* par exemple) et de classer les types par ordre décroissant. Cette méthode est appelée *Patron*.

Pour cette première évaluation, nous souhaitons étudier la qualité de notre modèle avec différentes variations. Nous avons imaginé quatre voies qui diffèrent sur le jeu de documents utilisé pour estimer les distributions de mots. Trois différentes manières de construire un jeu de documents ont été envisagées. Les deux premières consistent à récupérer les pages Web ou les snippets retournés par un moteur de recherche (ici Yahoo !) en lui soumettant le type ou l'entité. La troisième voie est spécifique au jeu de documents relatif au type. Dans les deux méthodes précédentes, nous récupérions des documents en relation avec le type. Nous proposons ici de récupérer des pages chacune dédiée à une entité du type. Pour cela nous récupérons des entités du type et leur page Wikipédia (via DBPedia). Ici, nous ne récupérons pas la première page Web que pourrait nous retourner un moteur de recherche car nous ne pourrions être sûr qu'il s'agit bien d'une page liée à cette entité et non pas à un homonyme. Nous ferons référence à ces trois variantes sous les noms : *sWeb*, *sSnippet* et *sEntities*.

Les quatre premières méthodes que nous allons comparer proposent différentes combinaisons de ces trois manières de construire les jeux de documents. Les méthodes sont *sWeb*, *sSnippet*, *sWeb/sSnippet* et *sEntities/sWeb*. Le nom décrit la méthode utilisée pour la récupération des documents pour le type et l'entité (séparé par "/") [6].

	Methodes				Baselines	
Mesure	sWeb	sSnippet	sWeb/sSnippet	sEntities/sWeb	Aléatoire	Patron
MRR	0,3410 (2,93)	**0,4224 (2,36)**	0,3609 (2,77)	0,2299 (4,35)	< 0,0020	0,3019 (3,31)
iPR1	0,0693 (40,84)	0,0753 (37,58)	0,0569 (49,74)	**0,1185 (23,88)**	< 0,0246	0,0627 (44,34)

TABLE 1 – Évaluation avec la moyenne des réciproques des rangs (MRR) et la précision interpolée pour un rappel de 1 (iPR1) comparé aux deux "baselines". Pour chaque méthode, nous utilisons 100 pages Web ou snippets. *Aléatoire* est calculé par rapport à la meilleure méthode pour la mesure étudiée. Le score entre parenthèses est le rang moyen correspondant au score associé.

Les premiers résultats Tableau 1 montrent plusieurs choses intéressantes. La principale est que tous les essais de notre proposition obtiennent de meilleurs résultats que les "baselines". Par exemple, obtenir un classement au moins aussi bon que sSnippet, au regard de la moyenne des réciproques des rang, en classant aléatoirement les types pour les entités, n'a que 0,2% chances de se réaliser. De plus elle est meilleure que la baseline utilisant des patrons inspirés de (Hearst, 1992) qui pourtant permettent d'obtenir dans de nombreuses tâches des résultats état

4. http ://mappings.dbpedia.org/server/ontology/classes
5. $MRR = \frac{1}{Q}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$
6. L'absence de "/" signifie que la même méthode est utilisée pour les deux éléments (sWeb = sWeb/sWeb et sSnippet = sSnippet/sSnippet).

de l'art. La deuxième observation est qu'il est préférable d'utiliser les snippets à la place des pages Web pour construire les jeux de documents. Cela est en partie dû au bruit que les pages Web contiennent. Une explication complémentaire serait que les pages Web avec un rang élevé ne sont pas (ou peu) relatives à la requête et leur utilisation modifient à ce point les distributions de mots, qu'elles ne correspondent plus aux types et aux entités.

Pour valider cette hypothèse, nous avons testé deux autres méthodes : *sWeb10* et *sSnippet10*. Elles diffèrent de *sWeb* et *sSnippet* dans le sens où seuls les 10 premiers éléments retournés par le moteur de recherche sont utilisés pour construire les jeux de documents. Les résultats sont présentés Tableau 2. La comparaison des deux premières colonnes donne l'avantage à l'utilisation de 10 pages Web au lieu de 100. Cela confirme notre intuition que les pages Web avec un rang élevé ne sont pas assez pertinentes et perturbent la mise au point d'un modèle lié à un type ou une entité. Cette tendance n'est pas retrouvée en ce qui concerne l'utilisation de snippets car nous disposons alors de trop peu d'information pour estimer une distribution des mots pertinente.

Mesure	sWeb	sWeb10	sSnippet	sSnippet10
MRR	0,3410 (2,93)	**0,3756 (2,66)**	**0,4224 (2,36)**	0,3550 (2,82)
iPR1	0,0693 (40,84)	**0,0781 (36,24)**	**0,0753 (37,58)**	0,0690 (41,01)

TABLE 2 – Évaluation pour sWeb et sSnippet de l'impact du nombre de pages Web utilisées pour construire les jeux de documents. Pour chaque méthode, nous comparons une version avec 100 ou 10 éléments (pages Web ou snippets). Le score entre parenthèses est le rang moyen correspondant au score.

Dans le Tableau 1, nous pouvons aussi remarquer (colonne 4) que l'utilisation de pages Wikipédia d'entités d'un type donné pour constituer le jeu de documents se situe largement en deça des autres méthodes pour la moyenne des réciproques des rangs (presque moitié moins qu'avec les snippets). Il est alors surprenant de remarquer qu'elle est la meilleure pour la précision interpolée (iPR1) (environ 1,5 fois supérieur à celui des snippets).

Dans le Tableau 3 nous cherchons à déterminer quel est le rang moyen où le type le plus fin (celui de plus bas niveau) d'une entité est retrouvé. Une évaluation dans ce sens a été faites pour *sWeb* et *sSnippet* sous les noms *sWebMax* et *sSnippetMax*. On peut tout d'abord constater que les types les plus fins sont retrouvés 1 fois sur 4 en première position (resp. environ 1/5) lorsque l'on utilise les snippets (resp. les pages Web). De plus, les résultats obtenus pour MRR montrent qu'en moyenne le type le plus fin est retrouvé dans les premiers rangs et que cela est très proche du rang où est trouvée la première réponse correcte pour *sWeb* et *sSnippet*. Dans le Tableau 1, l'iPR1 nous indique que tous les bons types d'une entité sont retrouvés aux alentours du rang 40. On peut donc déduire de ces deux informations que ce sont les types de plus haut niveau (ex : personne, lieu…) qui sont les plus difficiles à retrouver. La raison d'une telle différence est que le vocabulaire dans les documents liés aux types fins est plus précis, donc plus discriminant, qu'il ne l'est dans les documents liés aux types généraux. Ceci positionne notre proposition non pas comme une alternative aux systèmes de reconnaissance d'entités nommées mais comme un complément pour caractériser plus finement des entités.

Pourquoi l'utilisation des pages Wikipédia d'entités du type étudié est-il moins performant pour trouver les types fins que les autres méthodes et pourquoi l'est-il plus pour les types plus larges ? Pour le premier élément, la raison est probablement qu'étant donné que nous sommes dans le cas de documents encyclopédiques, les rédacteurs tendent à éviter les répétitions et les mots spécifiques (à une entité ou un type) ne sont pas vus plusieurs fois mais se retrouvent sous la forme de synonymes. Le deuxième, s'explique par le formatage spécifique des pages Wikipédia. En effet toutes les fiches concernant, par exemple, des personnes, commencent avec un texte du type : "X est né en DD-MM-YYYY à Y" et possèdent des infobox communes.

Mesure	sWeb	sWebMax	sSnippet	sSnippetMax
P@1	0,2348	**0,1900**	0,2762	**0,2500**
MRR	**0,3410 (2,93)**	0,2616 (3,82)	**0,4224 (2,36)**	0,3159 (3,17)

TABLE 3 – Mesure de la précision au rang 1 et la moyenne des réciproques des rangs (MRR). Les méthodes *sWebMax* et *sSnippetMax* considèrent uniquement le type le plus fin d'une entité comme correct. Le score entre parenthèses est le rang moyen correspondant au score associé.

4 Conclusions et perspectives

Nous avons présenté une méthode non-supervisée pour estimer dans quelle mesure une entité est d'un type donné (quel qu'il soit) en comparant les distributions de mots dans des documents liés à l'entité et liés au type.

L'évaluation que nous avons réalisée, sur 100 entités et 273 types, montre que cette voie est prometteuse est permet d'obtenir des résultats intéressants. Nous avons montré que notre proposition fonctionne mieux sur des types fins et spécifiques que sur des types trop larges et pourrait se situer comme complément aux outils de reconnaissance d'entités nommées existants.

En terme de perspectives, nous considérons différentes manières de calculer les distributions de probabilité (concepts sémantiques, concepts latents, utilisation de listes de mots outils, suppression des mots uniques...), différentes techniques de lissage (par exemple basées sur la hiérarchie des types), de calcul de similarité entre distributions, l'utilisation du contexte d'apparition de l'entité...

Retenons enfin que notre méthode de mesure d'appartenance d'une entité à un type possède des applications intéressantes pour l'aide à la constitution de bases de connaissances puisqu'elle pourrait permettre d'identifier, à partir d'un exemple (instance) et du nom de sa catégorie (concept), d'autres instances similaires ou proches. L'étude des éléments linguistiques ayant permis cette identification pourrait à son tour conduire à définir les contextes d'apparition possibles des instances ainsi que certaines de leurs propriétés ontologiques. Tout cela fait l'objet de nos travaux actuels.

Références

ASAHARA M. & MATSUMOTO Y. (2003). Japanese named entity extraction with redundant morphological analysis. *Human Language Technology conference - North American chapter of the ACL*.

BIKEL D., MILLER S., SCHWARTZ R. & WEISCHEDEL R. (1997). Nymble : a high-performance learning name-finder. *Proc. Conference on Applied Natural Language Processing*.

BONNEFOY L., BELLOT P. & BENOIT M. (2011). Une approche non supervisée pour le typage et la validation d'une réponse à une question en langage naturel : application à la tâche entity de trec 2010. *Huitième édition de la COnférence en Recherche d'Information et Applications*.

CHEN S. F. & GOODMAN J. (1998). An empirical study of smoothing techniques for language modeling.

CUCCHIARELLI A. & VELARDI P. (2001). Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics 27 :1.123-131, Cambridge : MIT Press*.

HEARST M. (1992). Automatic acquisition of hyponyms from large text corpora. *In Proceedings of the 14th International Conference on Computational Linguistics (COLING-92), 539–545*.

MCCALLUM A. (2003). Early results for named entity recognition with conditional random fields, features induction and web-enhanced lexicons. *Proc. Conference on Computational Natural Language Learning*.

NADEAU D. & SEKINE S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes, Vol. 30, No. 1. (January 2007), pp. 3-26.*

PASCA M. (2004). Acquisition of categorized named entities for web search. *2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13*.

PASCA M., LIN D., BIGHAM J., LIFCHITS A. & JAIN A. (2006). Organizing and searching the world wide web of facts-step one : The one-million fact extraction challenge. *Proc. National Conference on Artificial Intelligence*.

SEKINE S. (1998). Description of the japanese ne system used for met-2. *Message Understanding Conference.*

SEKINE S. & NOBATA C. (2004). Definition, dictionaries and tagger for extended named entity hierarchy. *Proc. Conference on Language Resources and Evaluation*.

SEKINE S., SUDO K. & NOBATA C. (2002). Extended named entity hierarchy. *Proceedings of 3rd International Conference on Language Resources and Evaluation (LREC'02)*.

TALUKDAR P. P. & PEREIRA F. (2010). Experiments in graph-based semi-supervised learning methods for class-instance acquisition. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (2010), pp. 1473-1481*.

VOORHEES E. M. (1999). The trec-8 question answering track report. *NIST Special Publication 500-246 : The Eighth Text REtrieval Conference (TREC-8)*.