Liste des figures

Liste des tableaux

Table des matières

Introduction

Introduction

Introduction

1- Le contexte de la recherche

Les travaux de cette thèse se situent entre deux spécialités de l'informatique : l'intelligence artificielle (IA) et le traitement automatique du langage naturel (TALN). Ils constituent les premiers domaines de recherche en informatique. Le premier domaine caractérise l'adaptation des formes d'intelligence au traitement automatique de l'information sur ordinateur, et il était un centre d'intérêt des chercheurs dès le début. Le deuxième domaine s'intéresse à la forme de communication la plus courante, et avec l'arrivée des ordinateurs, il s'est avéré être très important de faire des recherches sur un tel domaine.

L'intelligence artificielle à elle seule regroupe plusieurs directions. Dans notre thèse, nous nous sommes focalisés sur l'approche du raisonnement à partir de cas (RàPC). D'après M. Richter et R. Weber [1], les travaux sur cette approche se lancèrent à la fin des années 80 avec des ateliers aux États-Unis et les premiers workshops européens en Allemagne en 1993 [2], puis au Royaume-Uni en 1995 [3]. La première conférence internationale spécialisée se tint au Portugal en 1995 [4]. Nous présentons un historique et une présentation de cette approche appropriés à notre cadre d'étude dans le premier chapitre de notre thèse.

Avec l'augmentation considérable de la quantité et des sources d'information, le besoin d'un traitement automatique, rapide et compréhensible de l'information est devenu indispensable. Dès lors, dans le domaine du traitement automatique du langage naturel (TALN) de nouveaux axes de recherche apparaissent chaque jour, afin de traiter par ordinateur les différentes formes du langage. Selon S. Zaidi, le challenge auquel se confronte la recherche dans ce domaine est celui de la compréhension d'un contenu textuel, dont le but est de concevoir des outils capables de saisir le sens véhiculé par un texte [5].

Vu l'importance de la compréhension, nous nous sommes intéressés à la tâche d'annotation des rôles sémantiques, car elle vise, la compréhension du contenu textuel. Le système de D. Gildea et D. Jurafsky [6] constitue l'article fondateur de l'annotation des rôles sémantiques. Leurs travaux ont fait l'objet de plusieurs contributions marquées par la réalisation de la ressource PropBank et la conférence CoNLL, qui a consacré deux (02) années (2004 [7] et 2005 [8]) à l'annotation des rôles sémantiques [9]. Les systèmes supervisés actuels d'annotation des rôles sémantiques utilisent généralement FrameNet et PropBank [10].

2- Problématique

Vu les diverses caractéristiques de la langue arabe, il y a un nombre important d'obstacles morphologiques, structurels, formes des mots, etc. [5], et autres difficultés telles que le manque de ressources et le peu de travaux dans le traitement de cette langue par rapport aux autres langues. Dans [11], les auteurs citent "*…we are yet to read about large Arabic NLP applications such as Machine Translation and Information Extraction that are on par with performance on the English language. The problem is not the existence of data, but rather the existence of data annotated with the relevant level of information that is useful for NLP.*". Ceci exprime catégoriquement, à la fois l'importance de l'annotation des rôles sémantiques dans la traduction automatique, et la difficulté du manque de ressources annotées dans la langue arabe.

L'attention que nous portons sur l'annotation des rôles sémantiques dans la langue arabe vient du constat suivant :

- Le manque de travaux sur un système d'annotation dans cette langue. Très peu de contributions dans cette branche. Depuis les deux travaux de M. Diab [11], [12], [13], [14] ,il n'y a pas eu de contribution sur un système d'annotation des rôles sémantiques pour la langue arabe ;

- L'intérêt de la compréhension et des rôles sémantiques dans de nombreux domaines du traitement du langage naturel (TLN). Le chapitre deux (02) consacre une section à l'importance des rôles sémantiques dans le traitement du langage ;

- Le manque de ressources annotées pour la langue arabe.

2.1- Objectifs et contributions

Dans le cadre de cette thèse, nous avons ciblé un ensemble de buts et nous avons noté une multitude de contributions.

- ***Traitement Automatique de la Langue Arabe***

  - Un libre accès des recherches pour encourager d'autres contributions dans le traitement de la langue arabe ;

  - Se focaliser sur le domaine de la compréhension, car il a un impact sur d'autres domaines ;

- Ouvrir la voie de contribution dans une branche importante, mais qui a un très faible taux de recherche pour la langue arabe.

- ***Ressources utilisées***

  - Utiliser des données en libre accès de taille importante et qui couvrent une grande variété de textes ;

  - Ressource crédible et reconnue par la communauté internationale du traitement automatique du langage (TAL).

- ***Méthode proposée***

  - Nous nous basons sur le cycle du raisonnement à partir de cas (RàPC) pour l'annotation des rôles sémantiques pour langue arabe, pour exploiter les annotations disponibles et tirer profit des nouvelles annotations, tout en utilisant un cycle d'organisation des tâches ;

  - Une méthode qui peut être exploitée dans l'aide à l'annotation ;

  - Réalisation d'un système qui valide la méthode proposée sur un large corpus.

Ces points sont à la fois des considérations que nous devrons prendre en compte, tout au long de notre étude et une évaluation de l'ensemble du travail réalisé.

3- Organisation du manuscrit

Le présent document et les chapitres qui le composent sont organisés d'une façon qui va des aspects généraux de l'approche du raisonnement à partir de cas (RàPC) et la tâche d'annotation des rôles sémantiques, aux détails autour de la méthode proposée et l'annotation dans la langue arabe.

Dans le premier chapitre, nous aborderons l'approche du raisonnement à partir de cas (RàPC) sur laquelle est basée notre méthode d'annotation. L'historique, l'idée de base, le cycle, les travaux et les caractéristiques de cette méthode seront décrits.

Le second chapitre est dédié à l'objet de notre étude. Nous aborderons la discipline d'annotation des rôles sémantiques et l'importance de la compréhension dans le traitement automatique du langage naturel (TALN). Les travaux sur cette approche permettent de montrer son importance et justifier l'intérêt que nous portons à l'étude de cette tâche.

En raison du manque des travaux qui couvrent la tâche d'annotation des rôles sémantiques dans la langue arabe, nous avons jugé essentiel de consacrer le troisième chapitre à la description de cette tâche pour la langue arabe, afin de présenter le peu de travaux et de ressources disponibles pour travailler sur l'annotation des rôles sémantiques.

Dans le quatrième chapitre, nous présenterons notre proposition pour l'annotation des rôles sémantiques pour la langue arabe. Nous donnerons une description détaillée des différents aspects de notre méthode et les différentes formules utilisées pour son fonctionnement.

Dans le cinquième et dernier chapitre, nous détaillerons les différentes étapes de la construction des données et l'implémentation de la méthode. Un cas d'utilisation et des tests sont présentés pour la description et le fonctionnement de la méthode.

Le document se termine par une conclusion qui résume notre contribution dans cet axe de recherche et quelques perspectives sont suggérées qui peuvent apporter des réponses aux situations non traitées.

Chapitre 1 :

Le Raisonnement à Partir de Cas (RàPC)

Chapitre 1 : Le Raisonnement à Partir de Cas (RàPC)

Chapitre1 : Le Raisonnement à Partir de Cas (RàPC)

1- Introduction

Depuis le début, il y avait une collaboration entre informaticiens, mathématiciens, psychologues, médecins et bien d'autres chercheurs dans de nombreuses disciplines, afin de caractériser d'une manière informatisée l'intelligence humaine, animale ou autre.

L'intelligence artificielle (IA) est une branche de l'informatique qui essaye de modéliser et de reproduire les formes d'intelligence dans la nature. Parmi les approches d'intelligence artificielle : les réseaux de neurones, les colonies de fourmis, etc. Nous nous intéressons dans ce chapitre à une approche basée sur le raisonnement à partir de cas (RàPC).

Les problèmes rencontrés par un humain et leurs solutions sont souvent stockés dans sa mémoire, pour être utilisés comme expériences. Ce comportement humain face aux problèmes rencontrés dans sa vie de tous les jours est modélisé par l'approche du raisonnement à partir de cas.

2- Historique du Raisonnement à Partir de Cas (RàPC)

Selon M. Richter et R. Weber [1], le travail de R.Schank sur la mémoire dynamique et le modèle d'organisation de la mémoire en paquet (Memory Organisation Packet – MOP) a eu une conséquence majeure sur le raisonnement à partir de cas (RàPC) (Case Based Reasoning – CBR) [15].

Le principe de la théorie de la mémoire dynamique est tiré du fait que les expériences passées développent la compétence humaine à résoudre les problèmes [16]. De même, certains auteurs expliquent que le raisonnement à partir de cas est une façon de caractériser la résolution des problèmes par les humains sur l'ordinateur [17].

Des concepts modernes et antérieurs sur la psychologie la linguistique et l'intelligence artificielle constituent la théorie de M. Minsky [18]. D'après M. K. Haouchine [19], R. Schank [15] donne un autre point de vue sur la théorie de M. Minsky. Il suppose que la façon répétitive dont la procédure d'explication est employée coïncide avec un mécanisme de compréhension, il essaye de tirer le meilleur parti et rendre opérationnelle l'attitude humaine.

Dans [20] l'auteur mentionne que le système CYRUS est l'un des premiers mis en œuvre informatique du modèle de la mémoire dynamique de R. Schank [15]. Selon I. Watson et F. Marir, ils existent d'autres

systèmes qui sont fondés sur le modèle de la mémoire de cas tel que : MEDIATOR, CHEF, PERSUADER, CASEY et JULIA [21].

B. Fuchs regroupe les systèmes du raisonnement à partir de cas en deux catégories [22] : la première regroupe la résolution de problèmes, tels que le diagnostic, la décision, la conception et la planification. Dans cette première catégorie de systèmes, il est nécessaire de modifier la solution remémorée. Par contre dans la deuxième catégorie dite interprétative, il n'est pas nécessaire de la modifier. Ces systèmes s'intéressent à la remémoration pour certains buts, comme l'éclaircissement d'un problème.

2.1- Le raisonnement à partir de cas en Algérie

Les références du raisonnement à partir de cas citées ci-dessous sont l'une des premières contributions dans ce paradigme en Amérique et en Europe. Dans ce qui suit, nous citerons des travaux sur cette approche en Algérie.

Dans [23] les auteurs se basent sur l'approche du RàPC pour la réalisation d'une application d'aide à la décision dans le domaine médical, pour une maladie respiratoire (broncho-pulmonaire primitif). Le système est appliqué sur soixante (60) patients du service d'oncologie du Centre Hospitalo-Universitaires d'Annaba. La base de cas est constituée de 40 patients et de 20 patients pour la phase de test, le système donne un résultat prometteur de 80% de bons diagnostics. Également dans le domaine médical A. Khelassi et M. Amin-Chick appliquent le RàPC pour la classification d'électrocardiogramme (ECG) [24]. Dans [25], A. Khelassi présente, un travail concernant les problèmes cardiaques et la détection du cancer du sein. Aussi, dans le domaine médical, le système d'aide au diagnostic SRimCas [26] permet d'extraire les problèmes médicaux précédents à partir d'un problème en court.

Motivés par les coûts et le temps nécessaire à l'entrainement des personnes, certains chercheurs proposent d'appliquer le RàPC dans l'aide à l'entrainement [27]. Ils conçoivent un Framework basé sur une approche du RàPC et les ontologies pour l'indexation des cas, dans le but d'entrainer divers types de personnes.

Une recherche de Knowledge Intensive CBR (KI-CBR) est appliquée dans le domaine du diagnostic industriel pour des turbines à vapeur [28]. Un autre travail proche du domaine industriel dans le bâtiment est appliqué pour les constructions en béton armé, afin d'avoir une estimation de la vulnérabilité sismique de ces constructions [29]. Le système utilise 50 cas et donne une précision de plus de 90%.

Notre approche du raisonnement à partir de cas pour la traduction a montré la possibilité d'utiliser le raisonnement à partir de cas pour ne pas traduire des phrases déjà traduites et éviter de faire deux fois

les mêmes erreurs de traduction avec la possibilité de corriger les traductions proposées par le système et d'augmenter au fur et à mesure la base des phrases traduites [30].

3- Principe du raisonnement à partir de cas (RàPC)

L'idée du raisonnement à partir de cas est basée sur le principe d'analogie. Si une nouvelle situation " `cas cible` " est similaire ou ressemble à une situation passée " `cas source` ", alors des liens d'analogie peuvent être construits entre les deux cas [16].

Pour le carré d'analogie (Figure 1.1) [31], Δ symbolise les liens entre le cas source et le cas cible :

- Δ problème représente des liens comme similarité et dissimilarité entre le problème source et le problème cible.
- Δ solution symbolise la relation entre la solution du problème source et celle du problème cible.

Les relations de dépendance β symbolisent la liaison entre le problème et sa solution. Au moment de l'adaptation de la solution d'un problème source à celle d'un problème cible, la relation β peut être utilisée.

Selon M. K. Haouchine, le carré d'analogie contient principalement deux relations [19]:

- Les relations entre le problème et la solution d'un cas cible et d'un cas source ;
- La relation entre le problème et la solution d'un cas (cible ou source).

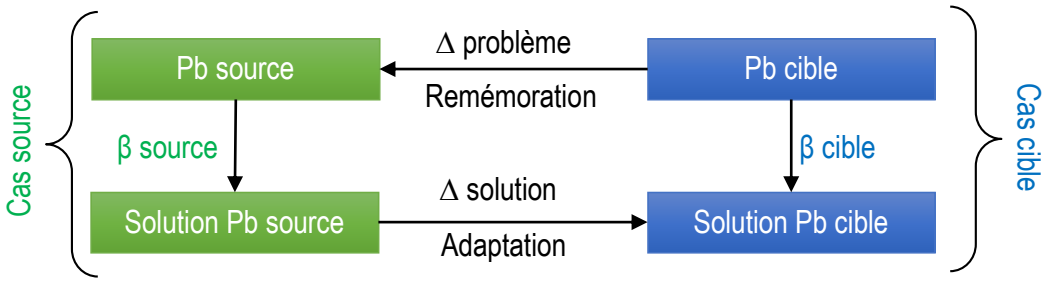


Figure 1.1. Carrée d'analogie (tiré de [31])

Pb source : problème source
Pb cible : problème cible

Nous avons expliqué au préalable que l'approche du raisonnement à partir de cas (RàPC) est basée sur l'analogie, puisqu'elle utilise l'expérience précédente pour résoudre un nouveau problème. Les cas sources (problèmes déjà résolus) sont stockés dans la base de cas, la résolution d'un cas cible (nouveau problème à résoudre) passe par plusieurs étapes du cycle du raisonnement à partir de cas. Les cas sources sont utilisés pour proposer une solution au problème cible.

4- Connaissances d'un système à partir de cas

Le raisonnement à partir de cas se compose de quatre (04) modules de connaissances [32]: vocabulaire d'indexation, base de cas, mesure de similarité et connaissances d'adaptation, décrits comme suit [33]:

- **Vocabulaire d'indexation** : ce module de connaissances contient les informations nécessaires à la représentation des cas, caractérisées par un ensemble fini de traits, afin de décrire le cas. Il a un impact majeur dans la construction de la base de cas et lors de la phase de remémoration.
- **Base de cas** : regroupe les expériences utilisées lors des différentes phases.
- **Mesure de similarité** : les différentes mesures nécessaires à l'estimation de la similarité entre les cas, lors de la phase de remémoration.
- **Connaissances d'adaptation** : Ces connaissances utilisées dans la phase d'adaptation permettent la modification ou l'évaluation de la solution proposée au nouveau problème. Généralement, les connaissances dans ce module ont un aspect de règles.

La figure 1.2 montre les quatre (04) modules de connaissances et les relations entre ces modules. Le module vocabulaire est la base des autres modules, les flèches symbolisent les liaisons entre les modules.

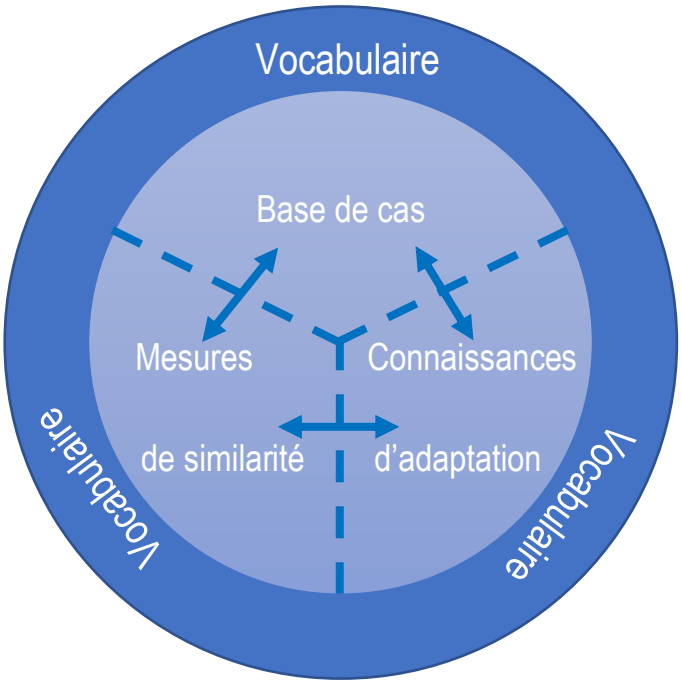


Figure 1.2. Les conteneurs de connaissances [34]

Dans la figure 1.3, L. Lamontagne et G. Lapalme regroupent les processus (phases) du cycle de raisonnement à partir de cas avec les modules de connaissances [33]. Ce modèle surligne les modules de connaissances et il est composé de deux parties, on-line et off-line [19].

- Le cycle du raisonnement à partir de cas forme la partie on-line ;
- La partie off-line contient les ressources du domaine tel que connaissances humaines, documents, base de données, informations sur le domaine, etc. À partir de ces informations, la partie "`authoring`[1]" aide à la construction des modules de connaissances.

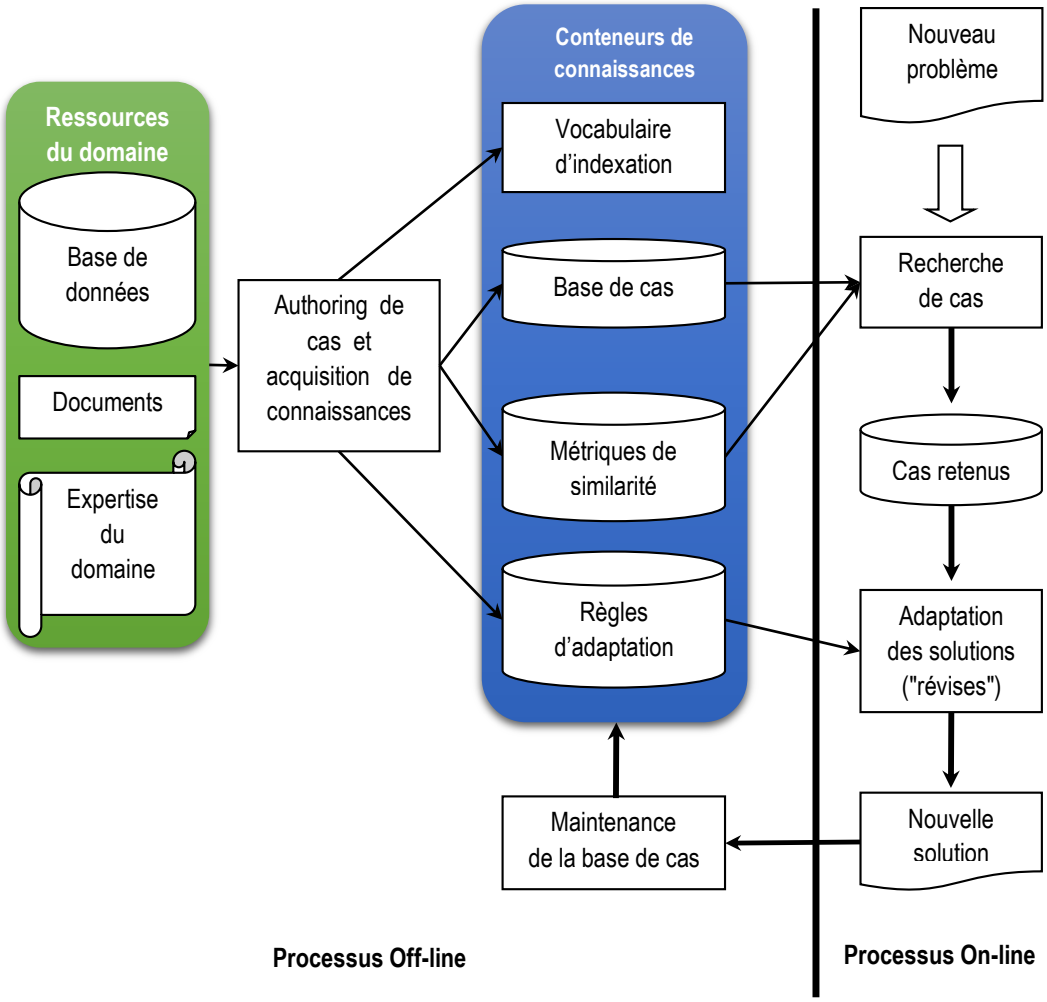


Figure 1.3. Modèle générique d'un système de RàPC [33]

[1] L. Lamontagne et G. Lapalme préfèrent utiliser le terme "`authoring`" à cause de l'absence d'un terme qui définit parfaitement la signification du terme anglais.

5- Cycle du raisonnement à partir de cas (RàPC)

Le cycle du RàPC est représenté par deux modèles [35]:

- **Modèle du processus du cycle RàPC** : ce modèle dynamique détermine les quatre phases (04) majeures de ce cycle (remémoration, adaptation, révision et mémorisation), les relations entre ces phases et le résultat de chaque phase. La figure 1.4 montre le cycle du RàPC selon A. Mille [36], les origines de ce cycle remontent aux travaux de A. Aamodt et E. Plaza [35].

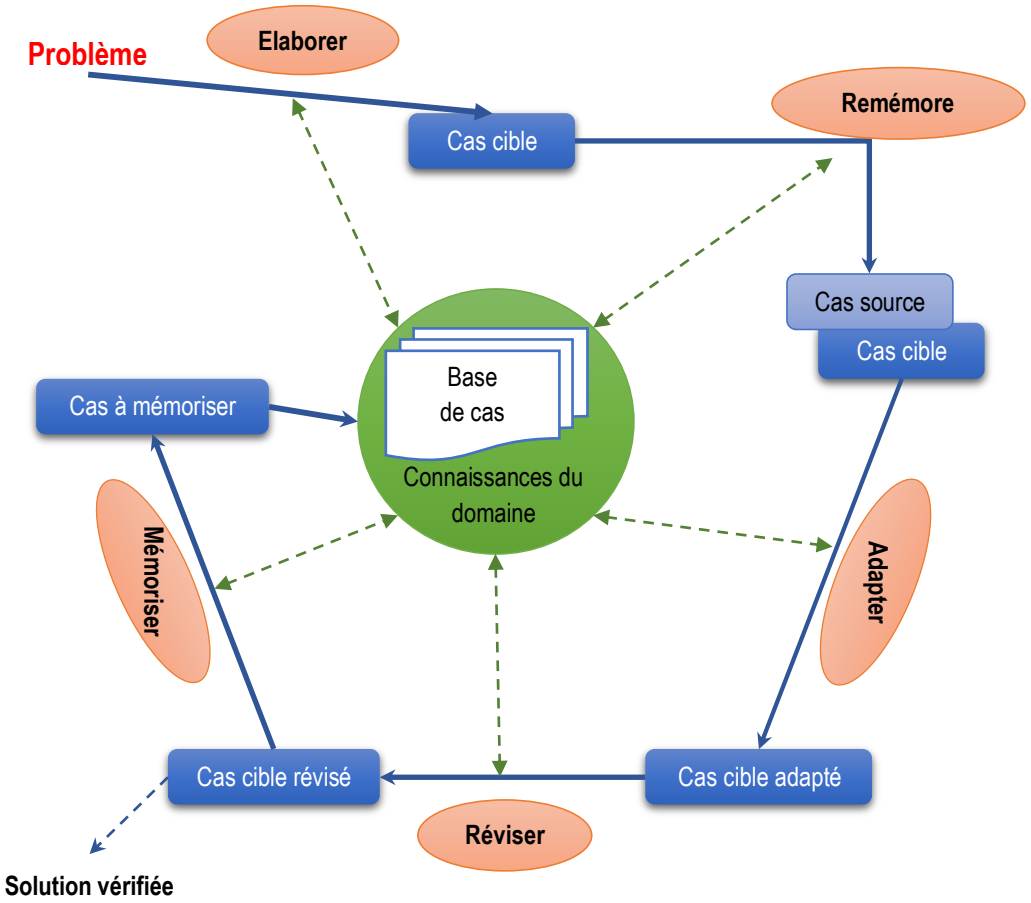


Figure 1.4. Cycle de raisonnement à partir de cas [36]

- **Structure de tâche-méthode** : Chaque phase est représentée par une tâche (Figure 1.5). Ainsi, la vue est orientée sur les tâches dans le but de diviser et de détailler les quatre phases majeures.

Le modèle du processus du cycle de RàPC est le plus utilisé dans la littérature et dans de nombreux secteurs d'application [37]. Pour cela, nous nous intéressons à l'utilisation de ce modèle dans notre travail.

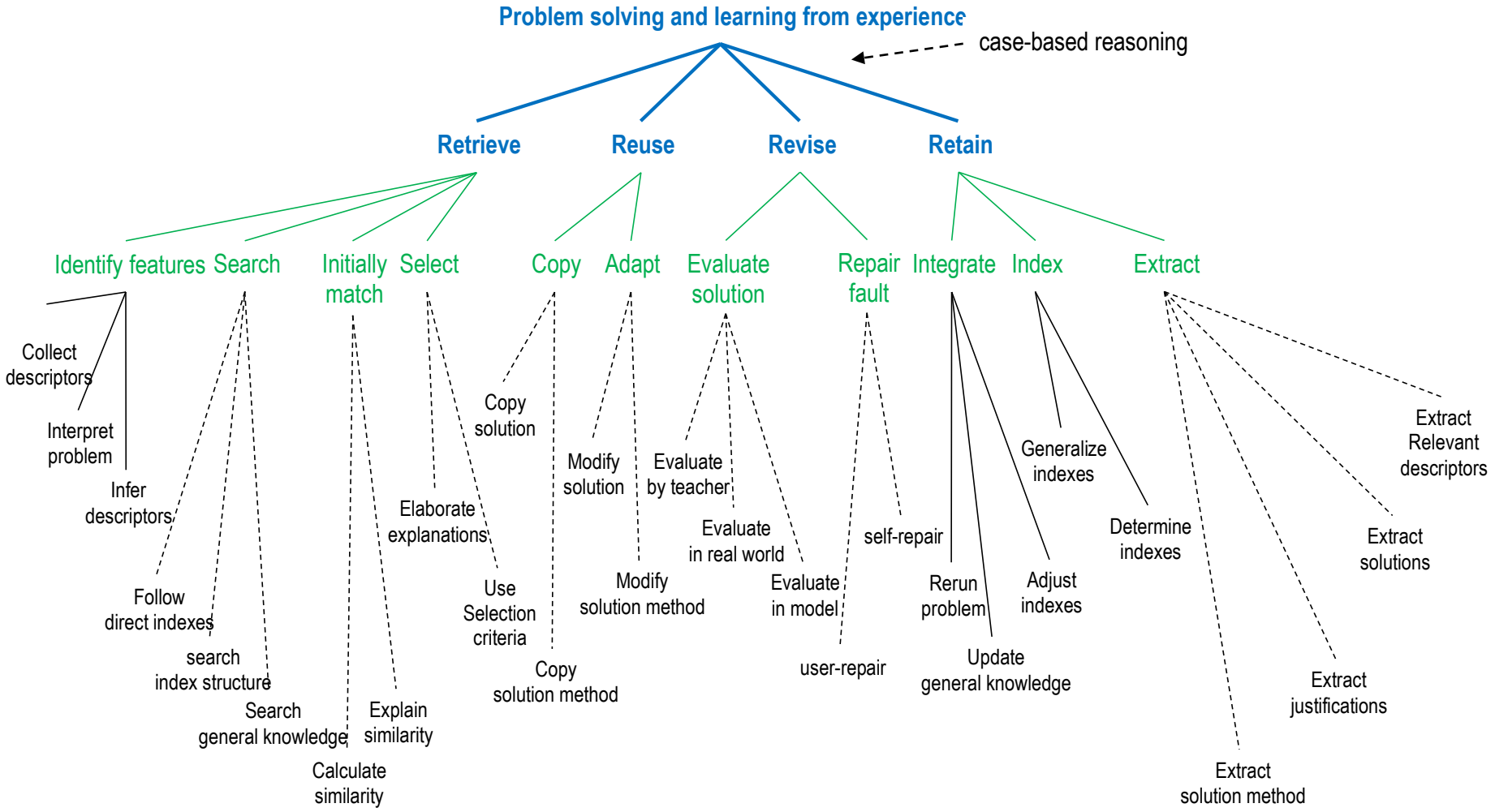


Figure 1.5. Décomposition de tâche-méthode en RàPC [35]

Dans [35] A. Aamodt et E. Plaza distinguent quatre (04) phases du cycle RàPC : remémoration, adaptation, révision et mémorisation. Une première phase d'élaboration est ajoutée avant les autres phases [38].

Selon nos recherches et les objectifs de notre thèse. Nous avons jugé que les cinq (05) phases sont importantes.

5.1- Phase 1 : Élaboration

Dans cette phase, le choix se porte sur les connaissances importantes, selon les objectifs du système et le domaine d'étude. Ces connaissances représentées sous forme de cas sont utilisées dans toutes les autres phases du cycle RàPC. C'est une phase élémentaire pour le bon fonctionnement et l'optimisation d'un système de RàPC [37].

"*On définit informellement l'élaboration comme une étape du RàPC qui a pour objectif de préparer la remémoration en enrichissant la description du problème posé au système.*" [38]. Les auteurs cherchent à donner une première formalisation de la phase d'élaboration, définie par la formule suivante (1.1) :

$$\text{Élaboration} : \text{pré-cible} \in \text{Problèmes} \rightarrow \text{cible} \in \text{Problèmes} \qquad (1.1)$$

5.1.1- Représentation d'un cas

Dans [38], les auteurs donnent une définition claire et simple au cas : un cas permet de décrire, par des concepts informatiques, une démarche de résolution de problèmes. Plus généralement, un cas est une représentation d'une connaissance du domaine, les connaissances et leurs représentations varient selon le domaine et les intérêts d'application du RàPC.

Il y a plusieurs formalismes pour la représentation des cas [39], certaines sont plus connues que d'autres. Nous citons : les couples d'attributs-valeurs [40], orientés objet [41], textuels [42], [43] et Frame [44]. Dans notre cadre d'étude, nous utilisons la représentation attributs-valeurs.

Souvent, un cas est défini par la paire `(pb, sol(pb)).` $pb \in$ `problème`, `sol(pb)` $\in$ `solution` [31]. Il peut y avoir des dépendances pour modéliser les liens entre `pb` et `sol(pb)`.

- La partie "`pb`" contient la description du problème ;

– La partie "`sol(pb)`" contient la description de la solution du "`pb`". Il y a deux types de cas : sources et cibles.

Un nouveau problème à résoudre est appelé un cas cible, représenté par (1.2) :

cible : (pb, sol(pb))	(1.2)

Où :

`pb` : description du problème cible.

`sol(pb)` : solution proposée à la partie "`pb`"

Un problème déjà résolu est appelé cas source. Les cas sources sont stockés dans la base des cas, représentés par (1.3) :

source : (srce, sol(srce))	(1.3)

Où :

`srce` : description du problème source.

`sol(srce)` : solution de la partie "`srce`".

5.1.2- Organisation de la base des cas

Il est nécessaire d'avoir une bonne organisation de la base de cas pour faciliter la phase de remémoration, et en général, pour faciliter l'accès à la base de cas. Dans la phase de mémorisation, le nouveau problème et sa solution sont mémorisés dans la base de cas avec les autres cas sources. Durant cette phase d'apprentissage, le nombre des cas sources augmente à chaque résolution d'un nouveau problème. Il existe plusieurs modèles pour permettre une recherche optimale dans la base de cas [37] :

- **Le modèle simple (Figure 1.6)** : ce modèle est sous une forme d'un arbre de décision (linéaire). Les nœuds de l'arbre sont des interrogations sur les index et les fils de chaque nœud représentent les réponses. Les questions posées permettent de construire l'arbre

dynamiquement. Ce modèle d'arbre peut être construit par des prototypes organisés d'une façon hiérarchique d'héritage. La figure 1.6 montre une structure d'arbre de décision avec les prototypes.

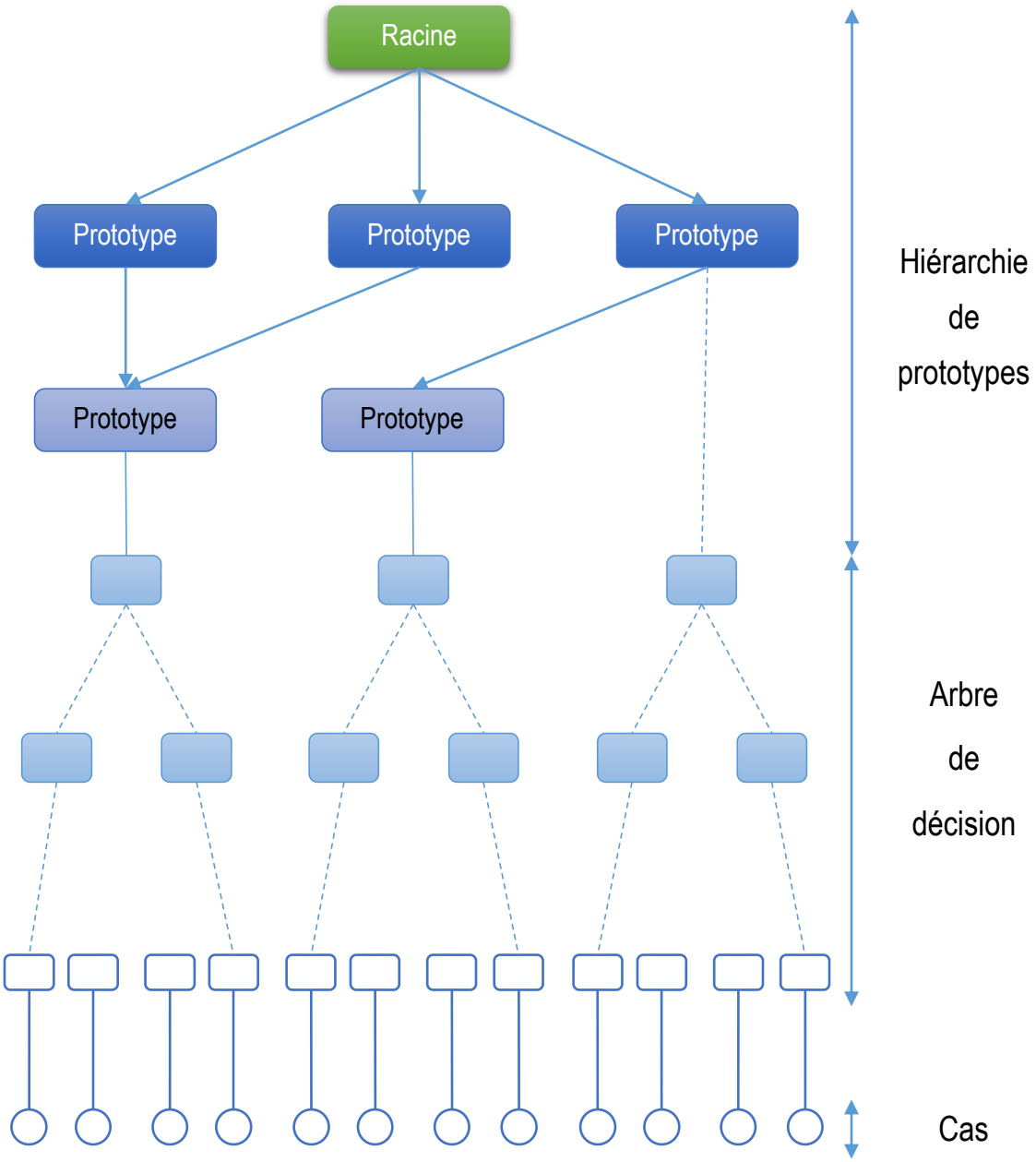


Figure 1.6. Le modèle hiérarchique de prototypes [37]

- **Le modèle dynamique (Figure 1.7)** : R. Schank et J. Kolodner sont les fondateurs de ce modèle. Ils ont construit la base de cas sous forme hiérarchique nommée "Memory Organisation Packets". Dans ce modèle, un épisode généralisé réunit les cas qui ont des caractéristiques semblables. La figure 1.7 illustre les trois types d'objets dans ce modèle. Il y a les propriétés semblables symbolisées par les normes. La distinction des cas de l'épisode généralisé est représentée par le deuxième type appelé index. Le dernier type correspond aux connaissances du système.

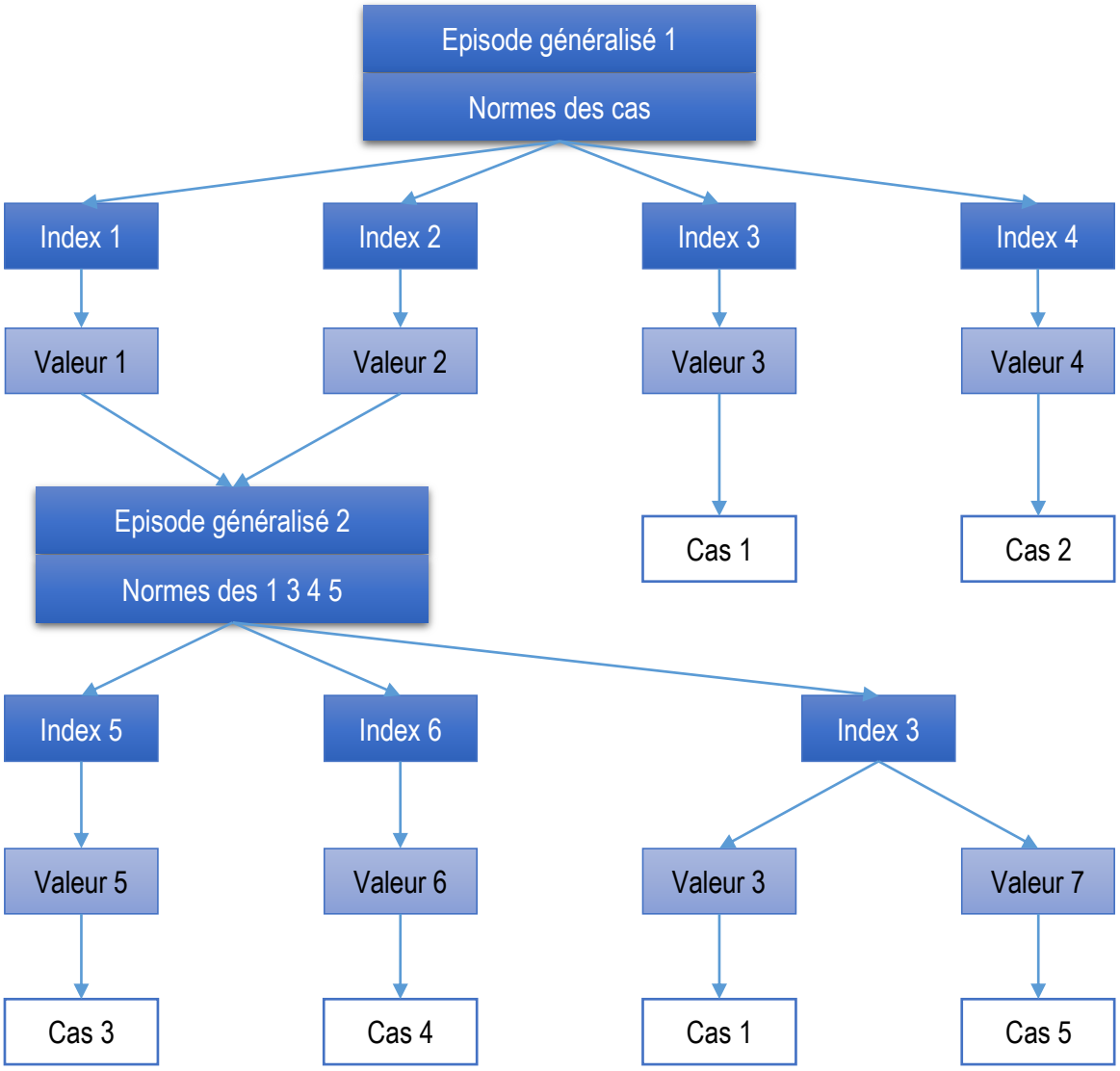


Figure 1.7. Le modèle dynamique [37]

- **Le modèle à base de catégories (Figure 1.8)** : ce modèle est caractérisé par un réseau de catégories et de cas caractérise ce modèle, il comporte trois (03) types de liens (index) :
  - Lien entre propriété d'une catégorie et un cas, ce lien appelé `rappel` ;
  - Entre des cas et la catégorie liée, appelé `exemple` ;
  - Le dernier entre deux cas qui ont quelques différences de propriétés, appelé `différence`.

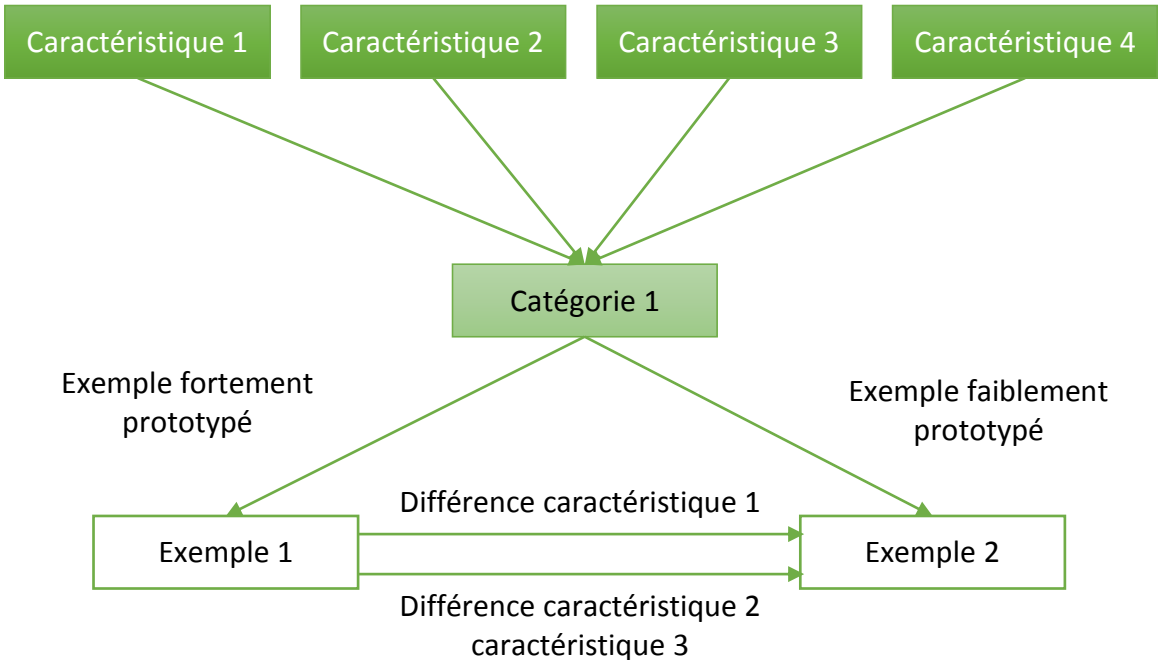


Figure 1.8. Le modèle à base de catégories [37]

5.2- Phase 2 : Remémoration

Le raisonnement à partir de cas (RàPC) est basé sur le principe que les problèmes précédents qui sont similaires à un problème en cours, leurs solutions sont utilisées pour trouver une solution au problème en cours. Cela fait de la remémoration, une phase cruciale pour l'approche du raisonnement à partir de cas [16].

Dans la première phase d'élaboration, nous avons abordé : la représentation des cas, l'indexation et l'organisation de la base de cas. Ces trois éléments essentiels de la phase d'élaboration sont à la base du fonctionnement de la remémoration. Cette phase est constituée de deux parties : filtrage et sélection. La première étape, optionnelle, de filtrage, consiste à éliminer un grand nombre de cas pour faciliter l'étape de sélection. La deuxième étape de sélection utilise des mesures de similarité entre le problème

cible et les problèmes sources. Son objectif est de trouver le ou les cas sources les plus similaires au cas cible [37].

5.2.1- Mesures de similarité

Il existe deux types de similarité, locale et globale [19]:

- **Similarité locale** : les mesures de similarité locale utilisent les propriétés du cas. Le type de propriétés (numérique, symbolique ou taxonomiques) généralement joue un rôle majeur dans cette similarité.
- **Similarité globale** : la similarité globale entre un cas cible et un cas source est calculée en utilisant la similarité locale. Il y a plusieurs types de similarité globale et ils sont choisis selon le domaine d'étude, nous citons :
  - Weighted block-city : $sim(A,B)=\sum_{i=1}^{n} w_i sim_i(a_i,b_i)$
  - Mesure euclidienne : $sim(a,b)=\left[\frac{1}{n}\sum_{i=1}^{n} sim_i(a_i,b_i)^2\right]^{\frac{1}{2}}$
  - Distance de Minkowski : $sim(A,B)=\left[\frac{1}{n}\sum_{i=1}^{n} sim_i(a_i,b_i)^r\right]^{\frac{1}{r}}$
  - Maximum based : $sim(A,B)=max_i w_i\, sim_i(a_i,b_i)$

Où :

a et b : attributs.
A et B : deux sujets d'étude.
n : nombre d'attributs.
w_i : le poids de l'attribut $_i$.
sim_i : similarité locale de l'attribut $_i$.

La sélection d'une mesure de similarité ouvre la voie à plusieurs algorithmes, comme l'algorithme des K plus proches voisins, les approches inductives, la logique floue, les réseaux de neurones, etc. Les deux premières approches citées sont les plus importantes dans le raisonnement à partir de cas [16]:

- **L'algorithme des K plus proches voisins** : c'est souvent, la méthode utilisée dans le raisonnement à partir de cas. Elle extrait les K problèmes sources, les plus similaires au problème cible (le problème à résoudre) [37].

- **Les approches inductives** : basées sur la discrimination des cas, elles organisent les cas de la base sous forme d'arbre de décision. Il existe une approche qui combine le RàPC et l'arbre de décision appelée KD-arbres, principalement utilisée dans le diagnostic et l'aide à la décision [37]. Ses algorithmes, comme ID3 [45], et CART (`Classification And Regression Trees`) structurent les cas sous forme d'arbre de décision. Généralement, les cas sources similaires sont groupés dans des clusters [16].

La sortie de cette phase donne un ensemble fini de cas sources. Ces cas similaires au cas cible servent comme entrée à la prochaine phase d'adaptation, afin de proposer une solution au problème cible.

5.3- Phase 3 : Adaptation

Dans cette phase, le ou les cas similaires sélectionnés dans la phase de remémoration sont utilisés. Le but est de proposer une solution au problème cible à partir de la solution de/des cas sources remémorés [46]. Dans [47], les auteurs considèrent l'adaptation comme une procédure de planification qui commence par la solution du problème connu (remémoré) et se termine par une solution adaptée au nouveau problème.

Selon M.K. Haouchine, l'adaptation est manuelle (humain) ou automatique, par algorithme, méthode, formules, règles, etc., (Figure 1.9) [19]. Des types d'adaptation automatique sont définis dans [48]:

- **Adaptation nulle** : c'est une adaptation simple qui consiste à prendre la solution du cas le plus similaire, comme solution au nouveau problème. Généralement, les systèmes commerciaux du RàPC utilisent ce type d'adaptation.

- **Transformationnelle** : basée sur les différences entre les attributs du problème remémoré et le nouveau problème, ce type d'adaptation utilise des règles pour transformer la solution du problème remémoré à une nouvelle solution pour le nouveau problème. Ainsi, des problèmes différents donnent des solutions différentes. Cette adaptation comporte deux types : substitutionnelle et structurelle.

- **Générative** : clairement différente de l'adaptation transformationnelle, elle ne transfère pas la solution d'un problème remémoré à un nouveau problème, mais elle transfère le chemin qui a mené à la solution du problème remémoré.
- **Compositionnelle** : ce type d'adaptation consiste à composer une solution à partir de plusieurs autres solutions, à condition que les solutions de plusieurs problèmes puissent être adaptées d'une manière non reliée et infaillible.
- **Hiérarchique** : les cas sont stockés de façon hiérarchique d'abstraction. L'adaptation est effectuée dans le niveau le plus haut d'abstraction.

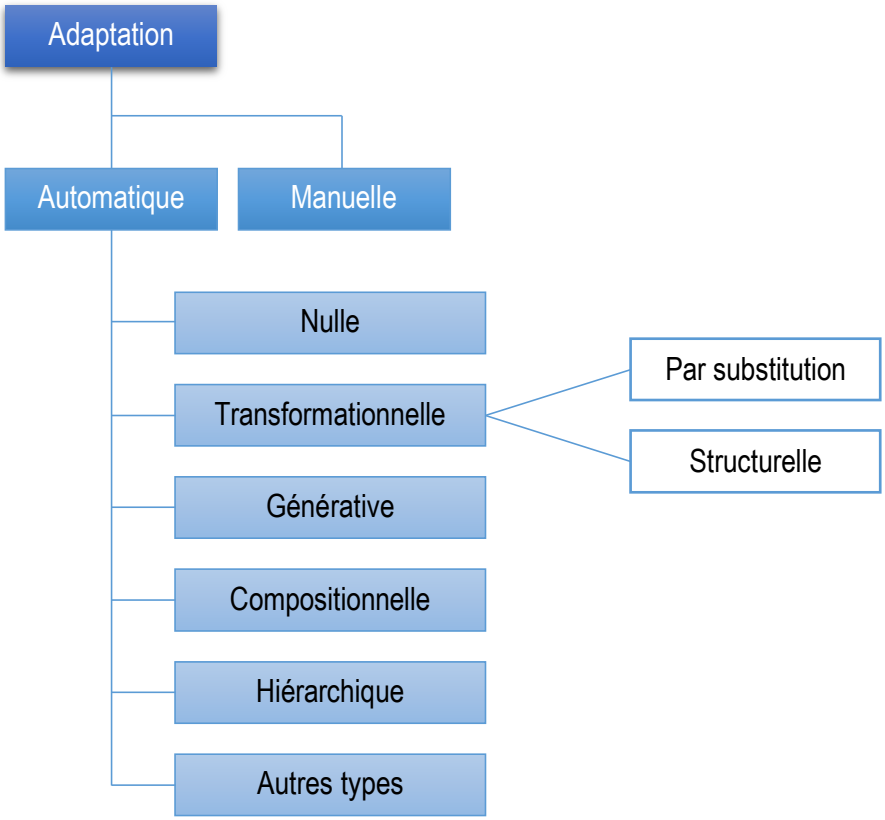


Figure 1.9. Types d'adaptation dans les systèmes de RàPC

Après l'adaptation d'une ou de plusieurs solutions des problèmes sources au problème cible, la solution proposée par le système doit être évaluée dans la prochaine phase du RàPC.

5.4- Phase 4 : Révision

Dans cette phase le résultat de la phase précédente est examiné. Les échecs de raisonnement sont le résultat des différences entre la solution proposée et la solution révisée. Généralement, elle est

composée de deux phases : évaluation de la solution proposée et la réparation, si nécessaire, de la solution [16].

Dans le cas où la solution devrait être corrigé, elle sera effectuée par [19]:

- **L'utilisateur** : intervention directe de l'utilisateur pour évaluer la solution proposée par le système, comme cité dans [49].
- **L'expert du domaine** : intervention d'un spécialiste du domaine d'application du système. Dans [50], les auteurs présentent, un travail autour d'un expert virtuel lors de l'étape de révision.
- **Automatiquement** : en se basant sur les cas sources de la base de cas, le système effectue une évaluation automatique de la solution.

La validation de la solution proposée par le système de RàPC dans cette phase permet un bon apport de celle-ci après sa mémorisation dans la phase mémorisation.

5.5- Phase 5 : Mémorisation

Après la vérification du cas dans la phase de Révision, il est nécessaire de mémoriser le cas cible (le problème cible et sa solution) avec les cas sources dans la base de cas. Cette phase de mémorisation et d'apprentissage permet l'expansion de la base de cas. La mémorisation n'est pas directe au risque d'avoir un doublement des connaissances dans la base de cas [16].

Il est important de tenir compte de certains aspects [51] :

- Faut-il ajouter ce cas dans la base de cas ? Pour résoudre cette considération, il faut comparer le niveau de la similarité avec ceux des cas sources dans la base de cas. Aussi, vérifier si le cas contient une particularité essentielle pour une éventuelle réutilisation ultérieure ;
- La manière d'indexer et d'ajouter ce cas dans la base.

Selon R. Bénard et P. De Loor [52] :

- Aucun cas source n'a le même contexte que le cas à mémoriser ;
- Même avec des contextes semblables, le cas à mémoriser a un but différent ;
- Le chemin qui mène à un but est inédit.

Éventuellement, la source de connaissances la plus importante dans le cycle de RàPC est la base de cas. Avec l'évolution et l'augmentation du nombre de cas dans la base, il y a un risque d'une mauvaise organisation et structuration de celle-ci. Ainsi, il est important d'effectuer une maintenance de la base [19].

6- Modèles du raisonnement à partir de cas (RàPC)

Il y a trois (03) groupes de modèles dans le RàPC [33]:

6.1- Le modèle structurel

Les premiers systèmes du raisonnement à partir de cas ont donné ce type de modèle. Les caractéristiques des cas sont données en avance par le concepteur du système pour présenter les éléments importants. Dans ce modèle, des paires (attribut/valeur) représentent les cas structurés où chaque paire désigne une caractéristique importante. Plusieurs types de valeurs peuvent représenter une caractéristique et les plus utilisés sont: entier, réel, booléen ou symbolique. La phase d'adaptation du RàPC est mise en œuvre seulement dans ce modèle.

Notre travail avec l'approche du raisonnement à partir de cas est basé sur le modèle structurel. Notre choix s'est porté sur ce modèle, car il est le plus adapté à notre contexte vu la possibilité et l'efficacité pour représenter les informations du domaine du traitement du langage naturel (TLN) sous formes structurelles. La figure 1.10 illustre un cas structurel dans notre approche de traduction automatique basée sur le RàPC.

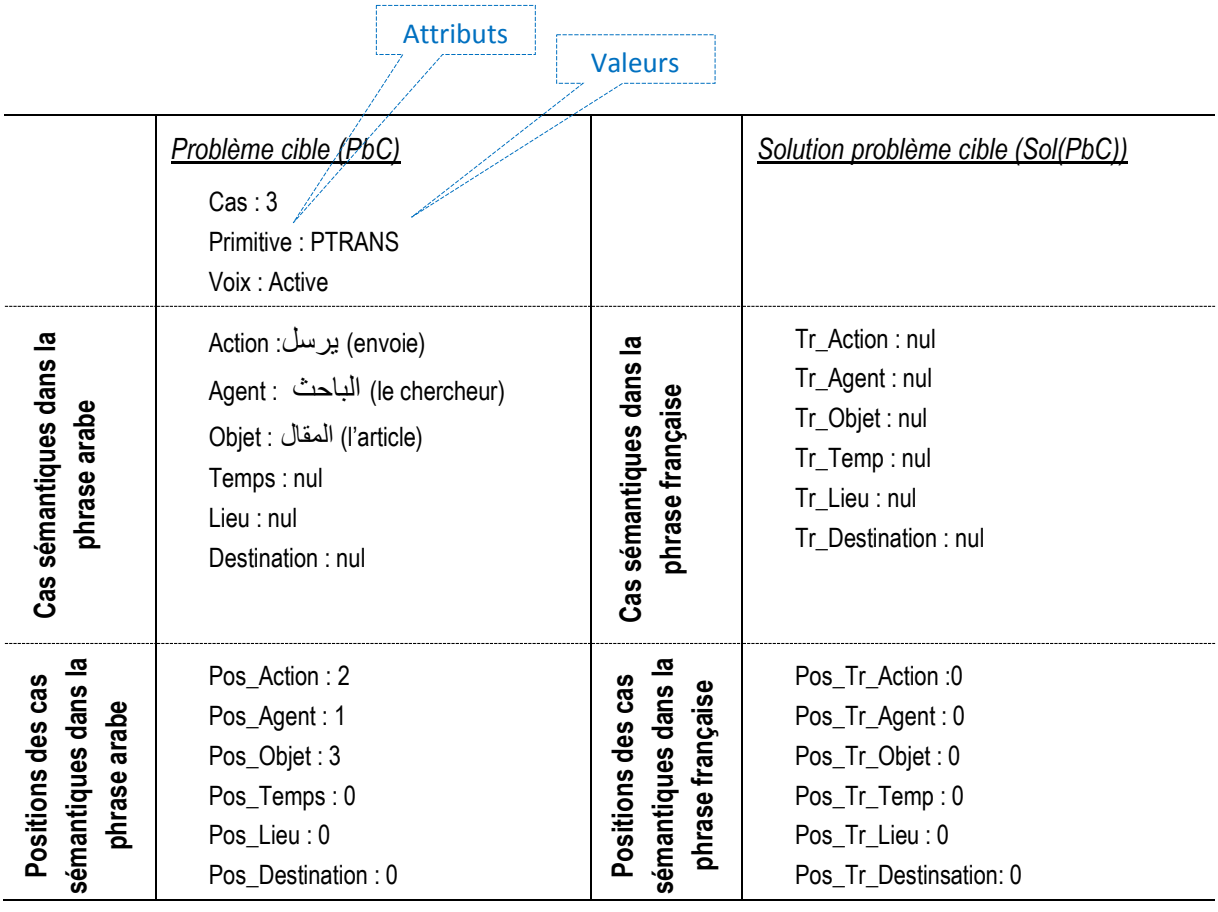


Figure 1.10. Cas cible avec le modèle structurel [30]

6.2- Le modèle conversationnel

La description d'un cas dans le modèle conversationnel est une représentation plus large des cas dans le modèle structurel. Il est souvent utilisé dans les systèmes commerciaux. La spécificité des domaines d'application de ce modèle est la difficulté de caractériser les problèmes dès le début par des caractéristiques bien définies, comme dans le modèle structurel.

La situation à résoudre est définie au fur et à mesure de l'interaction entre le système et l'utilisateur. Pour la sélection de la solution la plus adaptée au nouveau problème, une interaction est nécessaire entre l'utilisateur et le système [53]. Trois parties (03) caractérisent la représentation des cas dans ce modèle et sont décrites par (Figure 1.11) [33]:

- **Problème** : description textuelle courte de la situation.
- **Série de questions/réponses** : les questions représentent les index d'un problème et chaque question est pondérée, l'objectif étant d'avoir plus d'informations sur le problème.
- **Action** : description textuelle de la solution du problème.

Cas : 241
Titre : *cartouche d'encre endommagée causant des traces noires*
Description : *l'imprimante laisse de petits points noirs sur les deux côtés de la page. Parfois des larges tâches couvrent également la région à imprimer.*
Questions :
Est-ce que les copies sont de mauvaise qualité ? Réponse : oui Score : (-)
Quels types de problèmes avez-vous ? Réponse : trace noires Score : (default)
Est-ce qu'un nettoyage de l'imprimante règle le problème ? Rép : non …
Actions : *vérifier la cartouche d'encre et la remplacer si le niveau d'encre est faible*

Figure 1.11. Exemple de représentation d'un cas dans un modèle conversationnel de RàPC [33]

Dans ce type de modèle, l'interaction utilisateur/système commence par une première étape de courte description textuelle du problème par l'utilisateur. Puis, le système cherche le cas le plus similaire à la description de l'utilisateur, pour fournir à l'utilisateur un nombre **n** de questions/réponses. Ensuite, la similarité est recalculée progressivement avec les choix de l'utilisateur qui sont portés sur les questions auxquelles il veut répondre. Lorsque la similarité atteint un seuil prédéfini, une solution est proposée à l'utilisateur. Dans le cas où la similarité n'atteindra pas le seuil, aucune solution ne sera proposée et le problème est mémorisé sans solution.

6.3- Le modèle textuel

C'est un modèle récent et il n'a pas une description standard. Les cas dans ce modèle sont sous une forme textuelle, cette forme peut-être non structurée ou semi-structurée. Les cas non-structurés sont des textes simples en langage naturel (free-text). Si le texte est découpé en plusieurs parties, les cas sont dits semi-structurés. Les parties dans ce type de cas sont décrites par des étiquettes : problème, solution, etc.

Selon certains travaux, on distingue deux (02) grandes catégories dans ce modèle. La première basée sur la structuration des cas. Elle regroupe les travaux qui représentent le texte avec un nombre **n** de traits, en conformité avec le domaine d'étude (concepts, catégories, sujets, mots-clés, etc.). La deuxième catégorie est un élargissement du modèle de recherche d'information. Elle regroupe les travaux qui favorisent une indexation élémentaire, lors de la phase d'élaboration, mais une phase de remémoration complexe. Le système de questions/réponses du projet FAQFinder [54] est basé sur cette catégorie du modèle textuel. L. Lamontagne et G. Lapalme donnent une large description du modèle textuel, noté par le raisonnement à partir de cas textuel (RàPCT) [33].

7- Domaines d'application

Les systèmes basés sur le raisonnement à partir de cas sont utilisés dans de nombreux domaines, tels que : la cuisine, la médecine, le génie civil, le commerce, etc. Selon le domaine, il faut utiliser des algorithmes, des méthodes, des techniques, etc., adaptés à chaque domaine d'application [37].

Dans [55], l'auteur donne quatre (04) niveaux d'application. Elles sont classées hiérarchiquement et selon la figure 1.12, l'augmentation d'interaction entre l'utilisateur et la machine aide à la résolution de problèmes complexes:

- **Classification** : dans cette catégorie, toutes les informations nécessaires sont considérées comme disponibles. Elle regroupe des applications sur l'évaluation du risque, détermination des objets biologiques, analyse des données, etc.

- **Diagnostic** : dans cette catégorie les applications sont caractérisées par l'information insuffisante, donc un procédé de collecte d'information est nécessaire. Ce niveau regroupe des domaines comme le diagnostic des défauts de moteur de voiture, moteur d'avion, etc.

- **Aide à la décision** : le problème à résoudre est bien défini dans les autres catégories. Dans cette catégorie, le problème est défini lors de sa résolution. Il regroupe des applications de recherche de maisons pour achat, voyage improvisé, etc.

- **Gestion des connaissances** : c'est une catégorie plus complexe par rapport aux autres catégories. Elle ne comporte pas une seule démarche à suivre, mais plusieurs démarches peuvent être utilisées pour la gestion des connaissances.

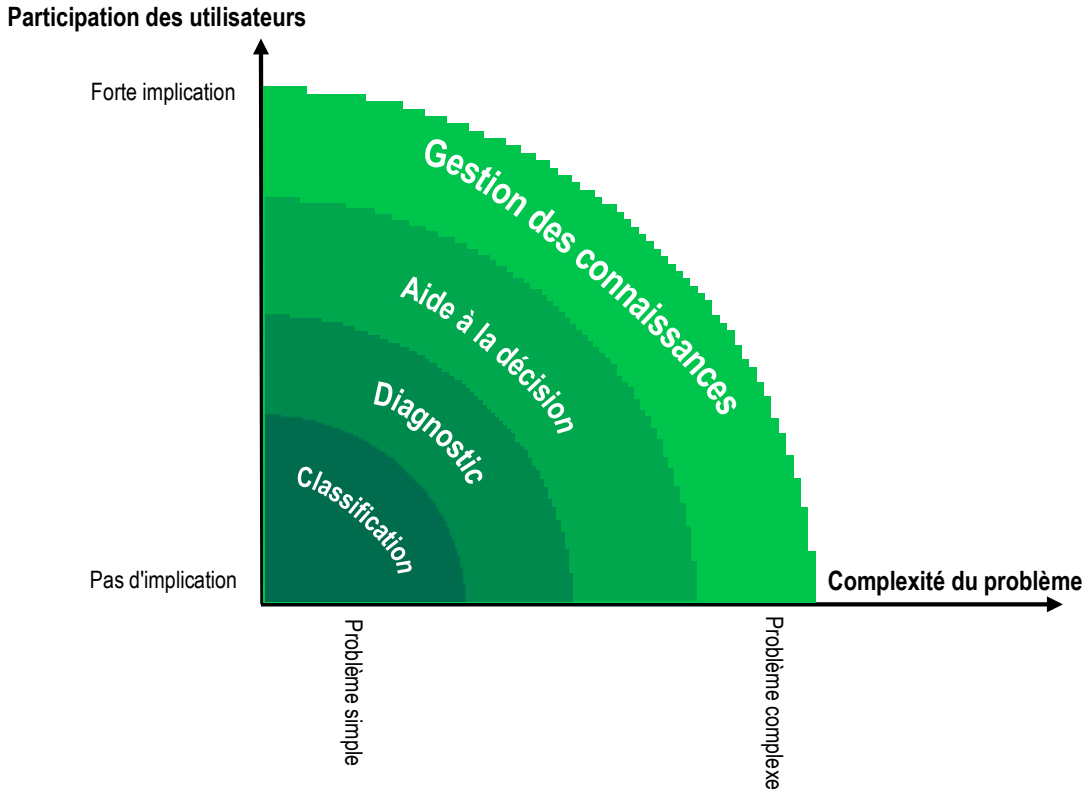


Figure 1.12. Les niveaux d'application dans le RàPC [55]

À travers le tableau 1.1, N. Armaghan [16] résume : les domaines d'application, auteurs, méthodes et systèmes par ordre chronologique.

Tableau 1.1. Domaines d'application

Auteurs	Année	Méthode/Système	Domaine de l'application du RàPC
Schank	1982	Memory Organisation Packets (MOPs) Theory	Résolution de problèmes et apprentissage
Kolodner	1983	CYRUS (premier système du RàPC)	Modèle de mémoire dynamique et MOP théorie de résolution de problème et l'apprentissage
Simpson	1985	MEDIATOR	Arbitrage
Hammond	1986	CHEF	Réparation et adaptation
Bain	1986	JUDGE	Raisonnement juridique
Sycara	1987	PERSAUDER	Arbitrage
Collins	1987	COACH	Réparation et adaptation

Farrel	1987	DECIDER	Enseignement
Alterman	1988	PLEXUS	Adaptation
Sharma et Sleeman	1988	REFINER	Acquisition de connaissances
Ashely	1988	HYPO	Raisonnement légaux
Koton	1989	CASEY	Diagnostic
Navichandra	1989	CYCLOPS	Conception
Goodman	1989	BATLE	Planification
Althoff	1989	MOLTKE	Diagnostic
Bareiss	1989	PROTOS	Diagnostic
Mostow et al.	1989	BOGART	Adaptation
Strube	1990	Event Project	Le rôle de la connaissance des épisodes dans les modèles cognitifs
Aamodt	1989 1991	GREEK	L'aspect apprentissage du RàPC dans l'acquisition des connaissances
Branting	1991	GREBE	Raisonnement juridique
Richter et Weiss	1991	PATDEX	Diagnostic
Simoudis	1992	CASCADE	Diagnostic
Redmond	1992	CELIA	Adaptation
Goel	1992	KRITIK	Adaptation
Hinrichs	1992	JULIA	Design
Sycara et al.	1992	CADET	Design
Pearce et al.	1992	ARCHI	Design
Moorman	1992	ACBARR	Adaptation
Oehlmann	1992	IULIAN	Révision de la théorie
Skalak	1992	CABARET	Combinassions du RàPC et RBR
Lopez et Plaza	1993	BOLERO	Planification
Costas et Kashyap	1993	TOTLEC	Planification
Watson et Abdullah	1994	PAKAR	Diagnostic
Mille et Fuchs	1995	PAD'IM	Supervision industrielle
Breslow et Aha	1997	NaCoDAE	Aide à la décision conversationnelle dans la Marine
Lieber et Napoli	1997	Resyn/RàPC	Aide à la conception de plan de synthèse en chimie organique
Badra et al.	2006	KASIMIR	Gestion des connaissances décisionnelles en cancérologie
Mille et Herbeaux	2007	ACCELEBRE	Aide à la conception de caoutchouc cellulaire
Caulier et Delepine	2007	SACRE	Aide à la capitalisation et la réutilisation d'expérience de supervision
Despres	2007	S3A	Aide à l'analyse d'accidents dans le domaine d'assurance

8- Les avantages et les limites du raisonnement à partir de cas (RàPC)

Les systèmes à base de règles (CBR) comportent des insuffisances par rapport aux systèmes à base de raisonnement à partir de cas (RàPC) [56] :

- L'acquisition des connaissances dans les systèmes à base de règles est une tâche particulièrement difficile ;
- Il n'y a pas une base consacrée aux expériences ni à la mémorisation des problèmes rencontrés ;
- L'inférence n'est pas véritablement efficace ;
- La résolution des exceptions n'est pas efficace ;
- Le résultat optimal ou la performance de l'intégralité du système à base de règles sont généralement insuffisants.

Au vu de ces insuffisances des systèmes à base de règles, l'approche du raisonnement à partir de cas apporte quelques remèdes. Selon L. Lamontagne et G. Lapalme, contrairement aux systèmes à base de règles, il est plus facile d'employer les moyens nécessaires à la réalisation de certaines applications basées sur le RàPC. Les problèmes liés à la tâche d'obtention des connaissances sont écartés. L'approche est très appropriée pour les systèmes comportant une intervention des experts du domaine et qui ont une mémoire des problèmes résolus. Aussi, elle est mieux adaptée pour les tâches qui ne nécessitent pas une résolution idéale des problèmes et où la formalisation des principes est insuffisante [33].

D'autres points forts du raisonnement à partir de cas sont cités dans [16] :

- Il fonctionne dans un domaine qui contient un nombre de cas minimes, puis il augmente le nombre de cas (ses connaissances) en mémorisant les cas résolus ;
- Possibilité de persuader l'utilisateur ou prouver la solution donnée par le système, en proposant des problèmes déjà résolus ou l'explication de cette résolution antérieure ;
- Sa flexibilité et la diversité des méthodes et des algorithmes de ses phases permettent son usage dans plusieurs domaines très variés : médecine, génie civil, diagnostic, etc. ;
- Ses similarités avec le raisonnement humain facilitent son acceptation par les utilisateurs et la confiance envers les solutions proposées.

9- Conclusion

Ce chapitre a été consacré à la description de l'approche du raisonnement à partir de cas. Nous avons cerné la globalité de l'approche. Nous pouvons attester que c'est une approche prometteuse très analogue au raisonnement humain et l'une des premières recherches dans l'intelligence artificielle.

Elle est utilisée dans de nombreux pays, colloques, ateliers et domaines d'application ce qui démontre l'efficacité, l'utilité et l'importance de ce type de raisonnement. Les chercheurs algériens s'intéressent à ce raisonnement, mais ils restent minimes en comparaison au nombre de recherches dans l'intelligence artificielle en Algérie et aux travaux de la communauté francophone du raisonnement à partir de cas.

Chapitre 2 :

Annotation des Rôles Sémantiques

Chapitre 2 : Annotation des Rôles Sémantiques

Chapitre 2 : Annotation des Rôles Sémantiques

1- Introduction

Depuis la nuit des temps, l'homme s'intéresse aux formes de communication, signes, images, voix, écriture, etc. Dans l'air actuel de la technologie et avec l'arrivée des ordinateurs et l'informatique, le traitement automatique du langage naturel en général, et la traduction automatique en particulier, sont parmi les premiers domaines de recherches.

Le traitement automatique du langage naturel (TALN) est une discipline qui regroupe plusieurs orientations comme la traduction, le résumé, les questions/réponses, les requêtes, l'opinion, etc.

Dans ces disciplines ou d'autres, comprendre la signification d'un mot ou un groupe de mots d'une manière automatique est une tâche très importante. Étant donné l'importance de la compréhension dans le traitement automatique du langage naturel, nous nous intéressons dans cette section à l'annotation des rôles sémantiques qui est une tâche de compréhension présente dans plusieurs disciplines telles que la traduction automatique.

2- Représentation du mot

"*La lexicographie est la science qui consiste à recenser les mots, les classer, les définir et les illustrer, par des exemples ou des expressions, pour rendre compte de l'ensemble de leurs significations et de leurs acceptions au sein d'une langue, afin de constituer un dictionnaire.*" [W1].

Elle se distingue de la lexicologie, de la sémantique et de l'étymologie. Le traitement du langage naturel (`Natural-language processing - NLP`) peut tirer parti de la lexicographie pour représenter le sens des mots. Mais, par rapport aux dictionnaires récents, des ressources telles que WordNet, Wikipédia et Wiktionnaire sont des ressources libres qui permettent aux utilisateurs un usage à la fois gratuit et sous une licence libre. La grande diffusion de ces ressources est due à leur libre utilisation pour des fins commerciales et de recherche [57]. Également, en comparaison avec la puissance des nouvelles machines, il n'y a pas un grand intérêt à l'utilisation de dictionnaires électroniques afin de faire des recherches plus rapides que celles d'un dictionnaire classique. Ainsi, une contribution telle que WordNet est un regroupement de la lexicographie et la puissance des ordinateurs [58]. Selon Q. Pradet, WordNet est la première représentation graphique d'une ressource lexicale [57].

D'autres contributions sont proposées, parmi elles, OntoNotes [59], [60], qui est l'un des deux projets qui ont conduit à l'organisation d'une compétition sur la désambiguïsation lexicale (`Word Sense Disambiguation`) à SemEval-2007[2] [61].

Dans le cadre de notre travail sur l'annotation des rôles sémantiques, nous avons opté pour l'utilisation d'une large partie de la version la plus récente d'OntoNotes (OntoNotes 5.0) (utilisée dans la célèbre conférence CoNLL-2012[3]), qui est le plus grand corpus sémantique existant pour la langue arabe.

3- Le concept d'annotation des rôles sémantiques

Aux questions comme Qui a fait ?, A Qui ?, Quand ?, Où ?, etc., les réponses sont représentées dans la phrase par différents moyens. Ces questions aident à la compréhension des évènements et cette dernière est une partie fondamentale dans la compréhension du langage naturel. La tâche d'annotation des rôles sémantiques s'intéresse à l'attribution automatique d'un rôle sémantique aux différents arguments d'un attribut pour répondre aux questions précédentes et participer à la compréhension de la phrase [10].

Pour l'anglais deux termes d'annotation des rôles sémantiques sont utilisés : "`Semantic Role Labeling`" si la ressource de référence est PropBank et "`Frame-Semantic Parsing`" dans le cas où la ressource de référence est FrameNet [57]. La communauté francophone utilise plusieurs termes :

- Étiquetage en rôles sémantiques ou reconnaissance de rôles sémantiques [62] ;
- Étiquetage de rôles sémantiques [63] ;
- Prédiction de la structure sémantique [64] ;
- Annotation syntaxico-sémantique des actants [65].

Dans notre travail, nous utilisons PropBank comme ressource de référence pour les rôles sémantiques. Cela, signifie l'utilisation du terme anglais "`Semantic Role Labeling - SRL`" dans la communauté anglophone. Pour cela, nous avons choisi d'utiliser le terme « annotation des rôles sémantiques », car c'est le terme français le plus proche au terme anglais.

[2] http://nlp.cs.swarthmore.edu/semeval/index.php
[3] http://conll.cemantix.org/2012/introduction.html

3.1- Qu'est-ce qu'un rôle sémantique ?

La notion de rôle est apparue, il y a longtemps dans la grammaire du sanskrit par Pāṇini avec le nom Kāraka. Cette notion de rôle est reprise dans les travaux de J. S. Gruber [66] et C. J. Fillmore [67]. Cela constitue le tout début des rôles sémantiques, appelés aussi : cas profonds, relations thématiques, rôles thématiques, thêta-rôles, etc. [68]

Nous prenons l'exemple donné dans [69] sur la phrase « vous cliquez sur le bouton » :

Vous cliquez sur le bouton

[Agent] **[Patient]**

Les rôles sémantiques Agent et Patient représentent chacun un argument de l'attribut (verbe). L'argument pronom personnel "Vous" prend le rôle sémantique Agent et l'argument "le bouton" joue le rôle sémantique Patient [69].

Il y a plusieurs types de rôles sémantiques et cela dépend de la ressource de référence utilisée : FrameNet, PropBank, VerbNet, etc. Le tableau 2.1 présente une liste de quelques rôles sémantiques qui ne sont pas spécifiques à une ressource précise.

Tableau 2.1. Rôles sémantiques avec exemples [68] (avec quelques modifications)

Rôles	Description	Exemples
Agent	Initiateur de l'action, capable de volition	Nadir a posé l'avion sans aucun problème.
Patient	Affecté par l'action, subit un changement d'état	Nadir a cassé la vitre.
Thème	Entité déplacée ou localisée	Nadir a jeté le frisbee. La photo se trouve sur le mur.
Expérienceur	Perçoit l'action, mais ne la contrôle pas	Nadir a vu Khaled sortir.
Bénéficiaire	Pour le bénéfice de qui l'action est accomplie	Nadir a loué un appartement pour Meriem.
Instrument	Intermédiaire ou moyen utilisé pour accomplir une action	Oussama a procédé à l'incision avec un bistouri.
Localisation	Lieu de l'objet ou de l'action	Il y a des monstres sous le lit.
Source	Point de départ	L'avion a décollé d'Annaba. Nous avons entendu la rumeur par un ami.
But	Point d'arrivée	Le ballon est arrivé dans le panier.

3.2- Étapes d'annotation

Généralement, l'annotation automatique des rôles sémantiques passe par quatre (04) étapes, comme la ressource de référence FrameNet [9]:

- **Étape 1 - Identification des prédicats :** reconnaitre les prédicats des autres unités lexicales.
- **Étape 2 - Désambiguïsation du cadre :** dans FrameNet, chaque sens a un cadre, où, pour un mot qui a plusieurs sens, il y aura plusieurs cadres. Cela consiste à lever l'ambiguïté sur le sens de l'attribut identifié, pour avoir un seul cadre.
- **Étape 3 - Détermination de la position des arguments :** principalement basée sur une analyse syntaxique, cette étape fixe la position des arguments évoqués par le cadre.
- **Étape 4 - Donner un rôle à chaque argument :** attribuer un rôle pour les arguments déterminés dans l'étape précédente.

3.2.1- Mesures d'évaluation de l'annotation

Il y a trois (03) mesures de similarité utilisées pour l'évaluation d'un système d'annotation des rôles sémantiques.

La première mesure est la "Précision" qui est un pourcentage des annotations correctes (2.1).

$$\text{Précision} = \frac{\text{étiquetage correct}}{\text{étiquetage correct} + \text{étiquetage incorrect}} \quad (2.1)$$

La seconde mesure est le "Rappel (Recall)" qui est un pourcentage des annotations correctes repérées correctement par le système (2.2).

$$\text{Rappel} = \frac{\text{étiquetage correct repéré}}{\text{étiquetage correct repéré} + \text{étiquetage correct non repéré}} \quad (2.2)$$

La dernière mesure est la "F-mesure" qui est la moyenne de l'accord entre précision et rappel [70] (2.3).

$$F\text{-}mesure=\frac{2Précision\times Rappel}{Précision+Rappel} \quad (2.3)$$

Le tableau 2.2 donne un exemple d'application de ces formules [71] :

Tableau 2.2. Exemple sur les mesures d'évaluation

Phrase référence	Il mange le gâteau avec la cuillère **[Agent] [Patient] [Instrument]**		
		Précision	**Rappel**
Annotation 1	Il mange le gâteau avec la cuillère **[Agent] [Patient]**	**100%**	**66%**
Annotation 2	Il mange le gâteau avec la cuillère **[Patient] [Patient]**	**50%**	**33%**

3.3- Features utilisés

Les features cités dans [72] sont souvent utilisés par les systèmes d'annotation, cependant, il peut y avoir d'autres features [10]. Plusieurs variétés de features peuvent être utilisées, elles sont choisies selon la langue, le lexique de référence, l'algorithme utilisé, etc. Les features et l'exemple donnés par D. Jurafsky et J. H. Martin dans [10], nous donnent un aperçu des features généralement employés dans les systèmes d'annotation. Cet exemple (Figure 2.1) présente une annotation des rôles sémantiques, selon la référence PropBank, la même référence d'annotation utilisée dans notre système d'annotation des rôles sémantiques.

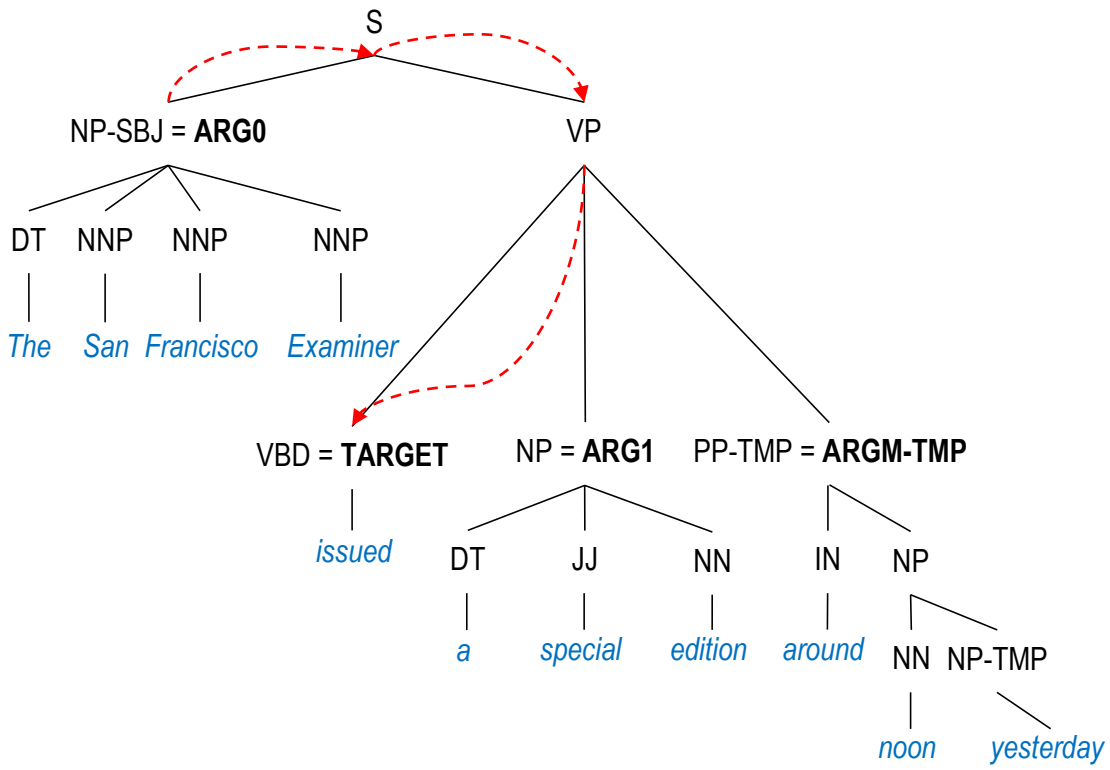


Figure 2.1. Annotation des rôles sémantiques, selon PropBank [10]

La ligne pointillée montre le chemin d'analyse $NP\uparrow S\downarrow VP\downarrow VBD$ pour le rôle sémantique $ARG0 = NP{-}SBJ$: The San Francisco Examiner

Les principaux features cités par D. Jurafsky sont [10] :

- **Le mot clé** : dans l'exemple précédent le mot "`Examiner`" du constituant "`The San Francisco Examiner`";
- **Partie de discours du mot clé** : dans le mot "`Examiner`" la partie de discours est `NNP`;
- **Chemin** : la représentation du chemin entre un constituant et le prédicat. Dans l'exemple précédent, le chemin entre le constituant "`The San Francisco Examiner`" et l'attribut (verbe) "`issued`" est `NP↑S↓VP↓VBD`. Et entre "`a special edition`" et le précédent attribut est `NP↑VP↓VBD`.
- **Voix** : passive ou active.

- **Position du constituant** : le constituant avant ou après l'attribut de la phrase. Dans l'exemple précédent, le constituant "`The San Francisco Examiner`" est avant l'attribut et les autres constituants "`a special edition`" et "`around noon yesterday`" sont après l'attribut.

- **Composants d'un constituant** : un constituant peut être composé de plusieurs mots, le premier et le dernier mot du constituant sont utilisés comme feature.

Par exemple les mots "`The`" et "`Examiner`" du constituant "`The San Francisco Examiner`".

4- Ressources lexicales

Nous citons plusieurs ressources dans le domaine d'analyse linguistique qui ont contribué au développement d'applications de traitement du langage naturel (TALN). Cependant, les plus importantes sont principalement des ressources dans la langue anglaise. Les ressources consacrées à la langue arabe et qui sont l'objet de notre travail seront présentées ultérieurement.

4.1- WordNet

La base de données lexicale WordNet [73] est créée pour l'anglais à l'université de Princeton par le psychologue Georges Miller et ses confrères pour des expérimentations dans la psychologie. Elle est parmi les ressources les plus utilisées pour la compréhension, question/réponse, résumé, etc., et principalement pour la désambiguïsation sémantique [74]. Un `Synset` ou groupement de synonymes dans WordNet est un regroupement de noms, verbes, adjectifs et adverbes. Il y a également des relations sémantiques qui relient les `Synsets` entre elles [68]. Le tableau 2.3 donne des statistiques sur la version 3.0 de WordNet.

Tableau 2.3. Statistiques sur WordNet 3.0 [W2]

Partie de discours	Chaine unique	Synset	Total de la paire mot-sens
Nom	117 798	82 115	146 312
Verbe	11 529	13 767	25 047
Adjective	21 479	18 156	30 002
Adverbe	4 481	3621	5 580
Total	155 287	117 659	206 941

Plusieurs entités composent la base de données WordNet [75]:

- **Synsets** : les lignes de cette table comportent synsetid, définition, parties du discours et lexdomainid. Elle est parmi les tables essentielles de WordNet.
- **Word** : elle contient deux champs (wordid et lemme), tous les lemmes sont mémorisés dans cette table.
- **Sens** : son rôle est la relation entre les mots et leurs définitions.
- **Lexdomains** : donne la définition du domaine lexical d'un mot.
- **Semlinks** : donne les relations sémantiques qui relient les Synsets comme : synonymie, implication, causalité, dérivation, etc.

La version 3.1 de WordNet avec son interface en ligne[4] permet à l'utilisateur de faire des requêtes en ligne. La figure 2.2 donne un exemple d'une requête du mot anglais "`Thesis`".

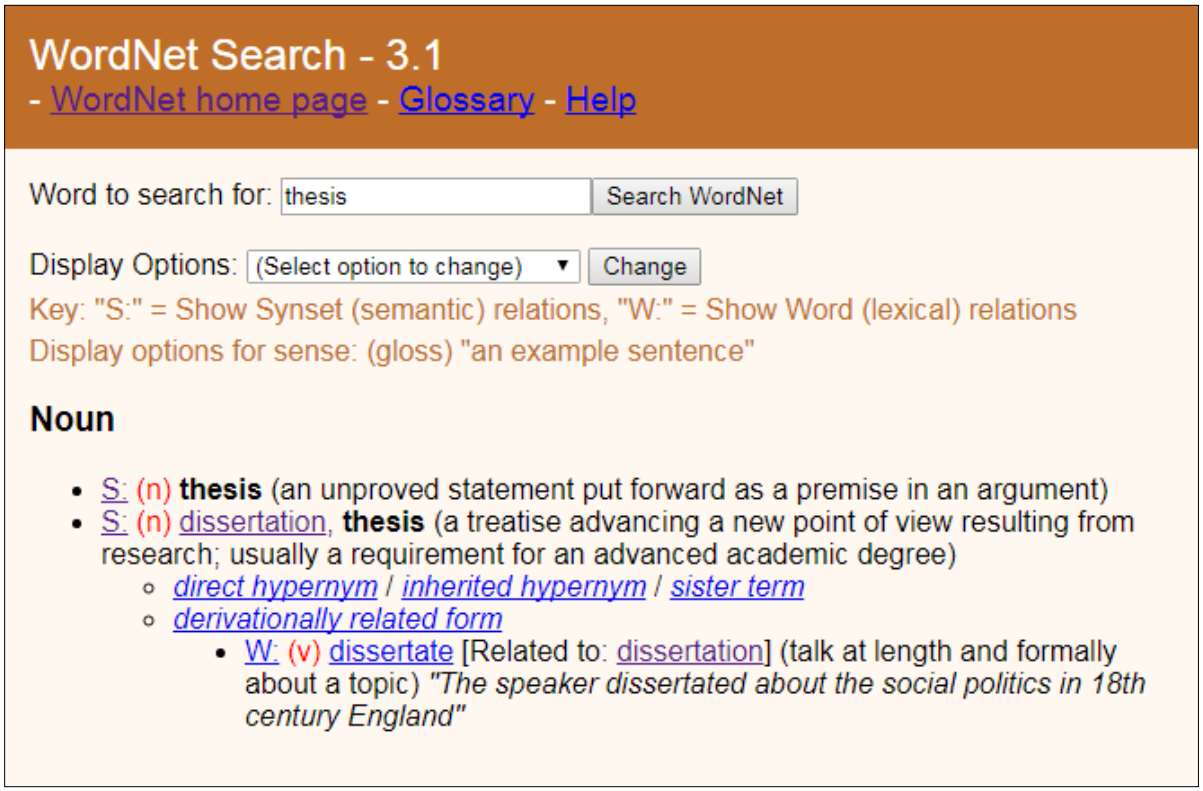


Figure 2.2. Recherche du mot anglais "Thesis" dans la version en ligne de WordNet 3.1 [W3]

[4] http://wordnetweb.princeton.edu/perl/webwn

4.2- FrameNet

Le projet FrameNet [76] est basé sur la théorie de la sémantique des cadres, son nom vient de la ressource lexicale citée précédemment WordNet. Des informations syntaxiques et sémantiques des mots sont analysées à partir du British National Corpus [77].

C'est un projet qui a démarré en 1997 à l'institut international d'informatique de Berkeley. Actuellement, FrameNet contient plus de 13.640 unités lexicales, 202.229 ensembles annotés et 1.224 frames sémantiques [W4]. Il peut être utilisé dans plusieurs applications de traitement du langage naturel comme l'annotation des rôles sémantiques, l'extraction d'information, la traduction automatique, l'analyse des sentiments, etc. [W5].

Une requête du verbe "`Write`" (Figure 2.3) sur le site de FrameNet montre que ce verbe dispose de plusieurs frames. Parmi ces frames, la frame "`Text_creation`".

Inspiré par l'exemple cité dans [63], nous donnons un exemple de la frame "`Text_creation`" (Figures 2.4, 2.5, 2.6 et 2.7) défini dans [W6]: un constituant qui joue le rôle sémantique "`Auteur`" pour produire quelque chose qui est caractérisé par le rôle sémantique "`Texte`", avec une possibilité de destiner le produit de l'auteur vers un destinataire qui est caractérisé par le rôle sémantique "`Destinataire`".

Lexical Unit	Frame	LU Status	Lexical Entry Report	Annotation Report
write down.v	Text_creation	Finished_Initial	LE	Anno
write in.v	Text_creation	Finished_Initial	LE	Anno
write in.v	Contacting	Created	LE	Anno
write out.v	Text_creation	Finished_Initial	LE	Anno
write up.v	Text_creation	Finished_Initial	LE	Anno
write.v	Statement	FN1_Sent	LE	Anno
write.v	Text_creation	Finished_Initial	LE	Anno
write.v	Contacting	Created	LE	Anno
write.v	Spelling_and_pronouncing	Created	LE	Anno
writer.n	People_by_vocation	Created	LE	

Figure 2.3. Frame sémantique du mot anglais "Write" [W7]

La frame "`Text_creation`" contient les informations suivantes [68] :

- **Définition** : donne une clarification de la frame, rôles sémantiques utilisés et des phrases exemples (Figure 2.4).

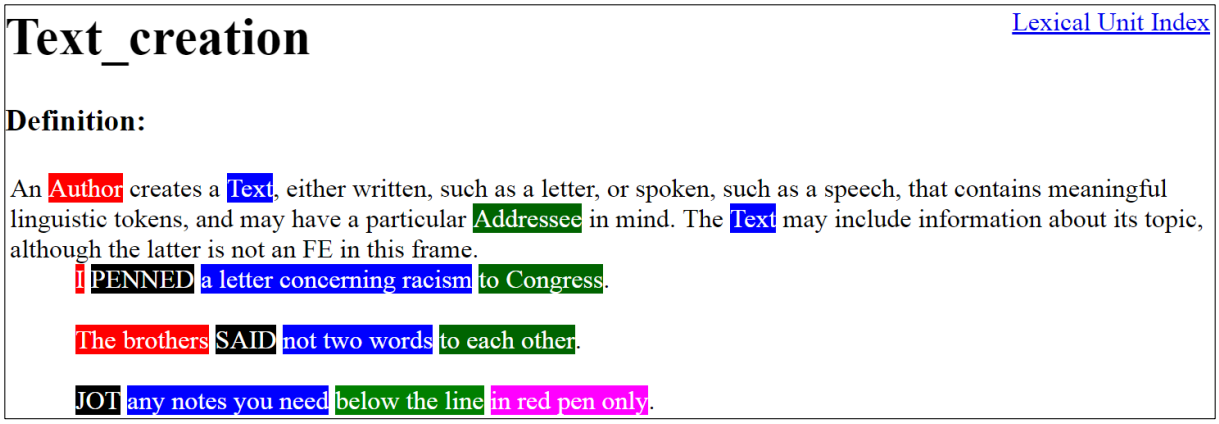


Figure 2.4. Extrait de la définition à partir de la frame "Text_creation" [W6]

- **Éléments de la frame (FEs)** : dans l'exemple des figures 2.5 et 2.6, il y a deux types d'éléments `Core` et `Non-Core`. Le premier regroupe les éléments nécessaires au sens de la frame, dans notre frame d'exemple "`Text_creation`" sont les éléments : "`Author`" et "`Text`". Le deuxième `Non-Core` groupe les éléments qui ne donnent pas une spécification à la frame, pour notre frame d'exemple : `Addressee`, `Beneficiary`, `Components`, `Depictive`, `etc`. [78].Tous ces éléments sont définis et plusieurs d'entre eux possèdent une phrase d'exemple.

- **Relations entre les frames** : contient la liste des frames (Figure 2.7) en relation avec la frame en question et chaque frame est dans une catégorie de relation (`Inherits from`, `Is Inherited by`, `Uses`, `etc.`).

- **Unités lexicales** : la frame "`Text_creation`" peut être appelé par ces unités lexicales (Figure 2.7)

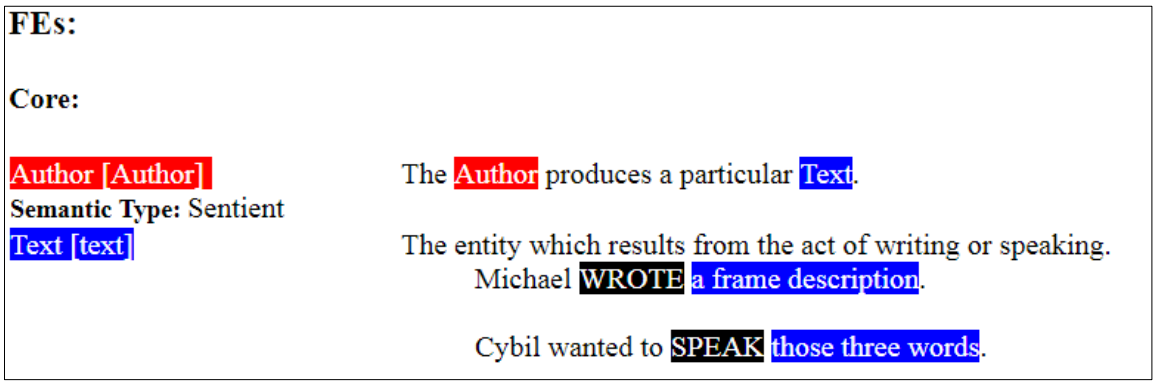
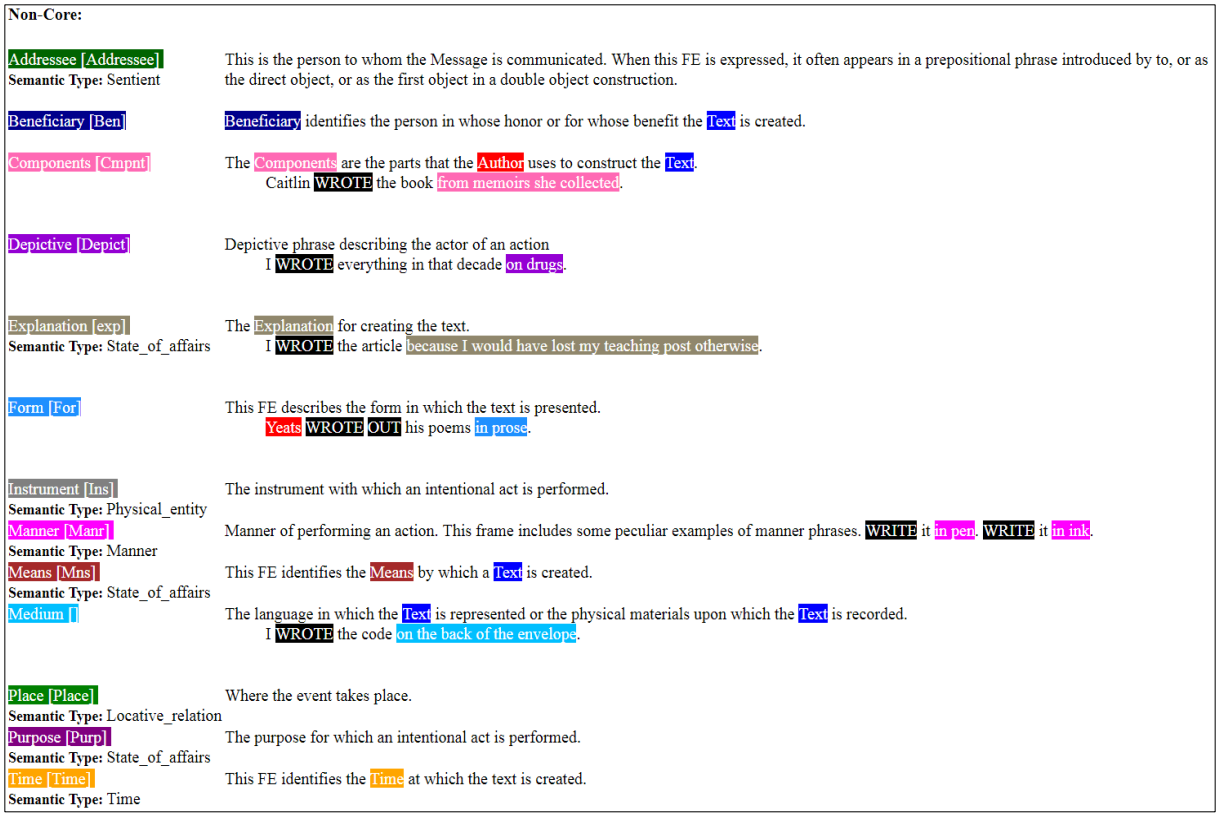


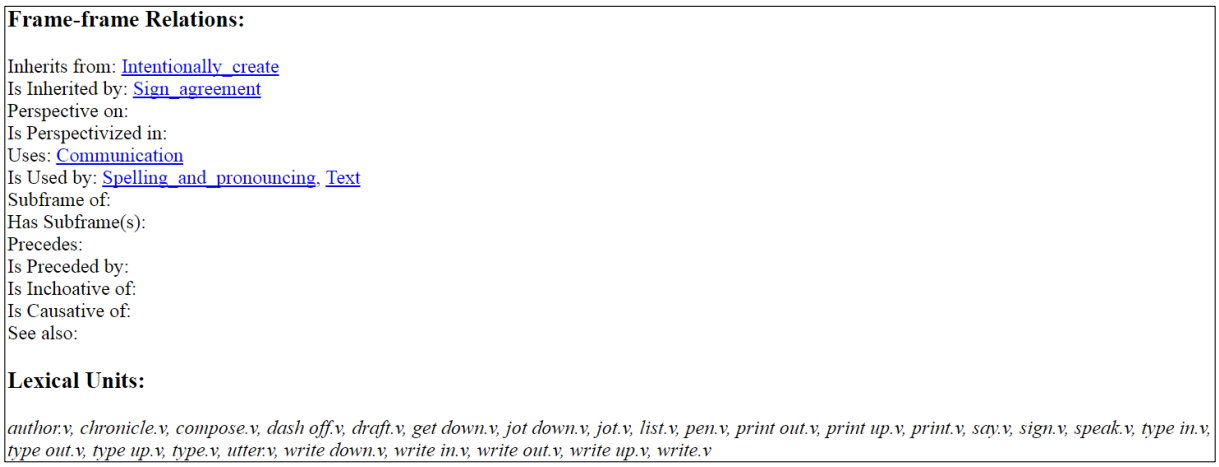
Figure 2.5. La partie "Core" de la frame éléments à partir de la frame "Text_creation" [W6]



Non-Core:

Addressee [Addressee] **Semantic Type:** Sentient	This is the person to whom the Message is communicated. When this FE is expressed, it often appears in a prepositional phrase introduced by to, or as the direct object, or as the first object in a double object construction.
Beneficiary [Ben]	Beneficiary identifies the person in whose honor or for whose benefit the Text is created.
Components [Cmpnt]	The Components are the parts that the Author uses to construct the Text. Caitlin WROTE the book from memoirs she collected.
Depictive [Depict]	Depictive phrase describing the actor of an action I WROTE everything in that decade on drugs.
Explanation [exp] **Semantic Type:** State_of_affairs	The Explanation for creating the text. I WROTE the article because I would have lost my teaching post otherwise.
Form [For]	This FE describes the form in which the text is presented. Yeats WROTE OUT his poems in prose.
Instrument [Ins] **Semantic Type:** Physical_entity	The instrument with which an intentional act is performed.
Manner [Manr] **Semantic Type:** Manner	Manner of performing an action. This frame includes some peculiar examples of manner phrases. WRITE it in pen. WRITE it in ink.
Means [Mns] **Semantic Type:** State_of_affairs	This FE identifies the Means by which a Text is created.
Medium [] 	The language in which the Text is represented or the physical materials upon which the Text is recorded. I WROTE the code on the back of the envelope.
Place [Place] **Semantic Type:** Locative_relation	Where the event takes place.
Purpose [Purp] **Semantic Type:** State_of_affairs	The purpose for which an intentional act is performed.
Time [Time] **Semantic Type:** Time	This FE identifies the Time at which the text is created.

Figure 2.6. La partie "Non-Core" de la frame éléments à partir de la frame "Text_creation" [W6]



Frame-frame Relations:

Inherits from: Intentionally_create
Is Inherited by: Sign_agreement
Perspective on:
Is Perspectivized in:
Uses: Communication
Is Used by: Spelling_and_pronouncing, Text
Subframe of:
Has Subframe(s):
Precedes:
Is Preceded by:
Is Inchoative of:
Is Causative of:
See also:

Lexical Units:

author.v, chronicle.v, compose.v, dash off.v, draft.v, get down.v, jot down.v, jot.v, list.v, pen.v, print out.v, print up.v, print.v, say.v, sign.v, speak.v, type in.v, type out.v, type up.v, type.v, utter.v, write down.v, write in.v, write out.v, write up.v, write.v

Figure 2.7. Relation entre frames et unités lexicales à partir de la frame "Text_creation" [W6]

4.3- VerbNet

C'est un lexique basé sur les classes de Levin [79], il contient des informations de nature syntaxique et sémantique sur les verbes anglais. L'effort pour construire le lexique et la possibilité de faire des erreurs lors de l'ajout de nouveaux verbes sont réduits grâce à l'utilisation des classes de Levin pour regrouper les verbes [80].

Les verbes dans VerbNet contiennent les informations suivantes [77] :

- **Liste des membres** : elle regroupe les verbes d'une classe ou sous classe.
- **Rôles sémantiques** : les membres ont une structure argument-prédicat, cette structure possède une liste de rôles.
- **Restriction** : connaissances sur le rôle.
- **Frames** : englobe les informations syntaxiques, les prédicats et les exemples.

Les figures 2.8, 2.9, 2.10 et 2.11 montrent des parties de la classe "`create-26.4`" selon la version 3.2 de VerbNet[5]. Cette classe contient 28 membres, 4 frames et une sous-classe "`create-26.4-1`" (Figure 2.9). La liste des rôles pour cette frame (Figure 2.10) est `Agent`, `Result`, `Material`, `Beneficiary` et `Attribute`. Parallèlement, la sous-classe "`create-26.4-1`" contient les mêmes informations Membres, Rôles et Frames (figure 2.11).

Le tableau (2.4) présente des statistiques récentes sur VerbNet.

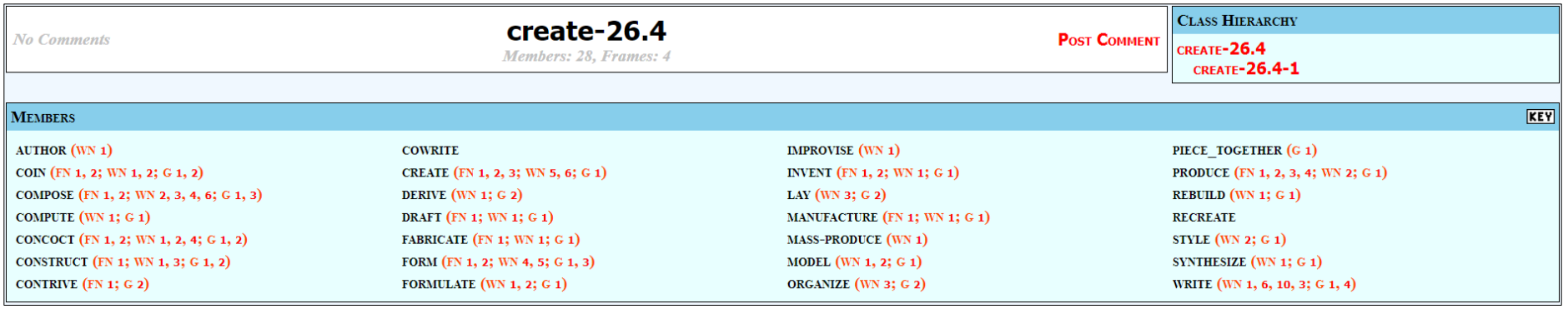
Tableau 2.4. Statistiques sur le lexique VerbNet [W8]

Type	Nombre
Premier niveau de classes	274
Rôles sémantiques	23
Prédicats sémantiques	94
Restrictions syntaxiques	55
Sens verbes	5257
Lemmes	3769

4.4- PropBank

Dans [81], I. Falk mentionne que l'intérêt de PropBank est la construction d'un corpus annoté à des fins d'entrainement pour des systèmes basés sur l'apprentissage supervisé et il précise que l'objectif de PropBank ne réside pas dans la classification des verbes ou la création d'une ressource lexicale [82].

[5] https://verbs.colorado.edu/verb-index/vn/create-26.4.php



No Comments

create-26.4

Members: 28, Frames: 4

POST COMMENT

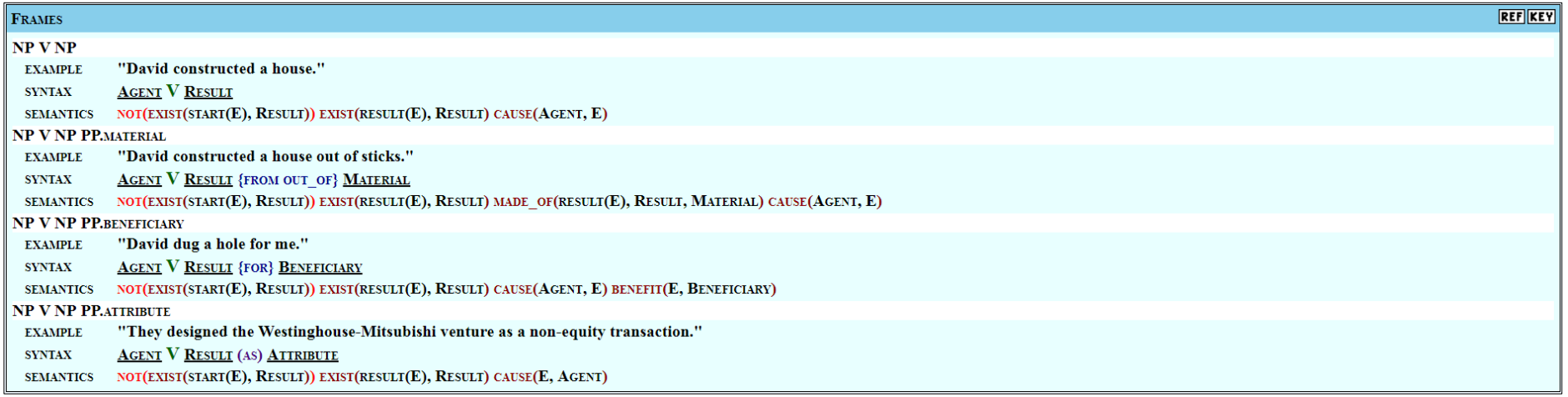
CLASS HIERARCHY

CREATE-26.4
CREATE-26.4-1

MEMBERS KEY

AUTHOR (WN 1)	COWRITE	IMPROVISE (WN 1)	PIECE_TOGETHER (G 1)
COIN (FN 1, 2; WN 1, 2; G 1, 2)	CREATE (FN 1, 2, 3; WN 5, 6; G 1)	INVENT (FN 1, 2; WN 1; G 1)	PRODUCE (FN 1, 2, 3, 4; WN 2; G 1)
COMPOSE (FN 1, 2; WN 2, 3, 4, 6; G 1, 3)	DERIVE (WN 1; G 2)	LAY (WN 3; G 2)	REBUILD (WN 1; G 1)
COMPUTE (WN 1; G 1)	DRAFT (FN 1; WN 1; G 1)	MANUFACTURE (FN 1; WN 1; G 1)	RECREATE
CONCOCT (FN 1, 2; WN 1, 2, 4; G 1, 2)	FABRICATE (FN 1; WN 1; G 1)	MASS-PRODUCE (WN 1)	STYLE (WN 2; G 1)
CONSTRUCT (FN 1; WN 1, 3; G 1, 2)	FORM (FN 1, 2; WN 4, 5; G 1, 3)	MODEL (WN 1, 2; G 1)	SYNTHESIZE (WN 1; G 1)
CONTRIVE (FN 1; G 2)	FORMULATE (WN 1, 2; G 1)	ORGANIZE (WN 3; G 2)	WRITE (WN 1, 6, 10, 3; G 1, 4)

Figure 2.8. Hiérarchie et membres de la classe "create-26.4"



FRAMES REF KEY

NP V NP

EXAMPLE "David constructed a house."
SYNTAX AGENT V RESULT
SEMANTICS NOT(EXIST(START(E), RESULT)) EXIST(RESULT(E), RESULT) CAUSE(AGENT, E)

NP V NP PP.MATERIAL

EXAMPLE "David constructed a house out of sticks."
SYNTAX AGENT V RESULT {FROM OUT_OF} MATERIAL
SEMANTICS NOT(EXIST(START(E), RESULT)) EXIST(RESULT(E), RESULT) MADE_OF(RESULT(E), RESULT, MATERIAL) CAUSE(AGENT, E)

NP V NP PP.BENEFICIARY

EXAMPLE "David dug a hole for me."
SYNTAX AGENT V RESULT {FOR} BENEFICIARY
SEMANTICS NOT(EXIST(START(E), RESULT)) EXIST(RESULT(E), RESULT) CAUSE(AGENT, E) BENEFIT(E, BENEFICIARY)

NP V NP PP.ATTRIBUTE

EXAMPLE "They designed the Westinghouse-Mitsubishi venture as a non-equity transaction."
SYNTAX AGENT V RESULT (AS) ATTRIBUTE
SEMANTICS NOT(EXIST(START(E), RESULT)) EXIST(RESULT(E), RESULT) CAUSE(E, AGENT)

Figure 2.9. Frames de la classe "create-26.4"

ROLES REF

- AGENT [+ANIMATE | +MACHINE]
- RESULT
- MATERIAL
- BENEFICIARY [+ANIMATE]
- ATTRIBUTE

Figure 2.10. Rôles de la classe "create-26.4"

No Comments

create-26.4-1

Members: 11, Frames: 1

POST COMMENT | TOP

MEMBERS KEY

CONJURE (FN 1; WN 1; G 1)	PUBLISH (WN 1, 2, 3; G 1)	STAGE (FN 1; WN 1; G 1, 2, 3)
CRAFT (WN 1)	REARRANGE (WN 1)	
DESIGN (FN 1; WN 2, 3, 4; G 1, 3)	RECONSTITUTE (WN 1)	
DIG (WN 2; G 2)	REORGANIZE (WN 1, 2; G 1)	
MINT (WN 1)	SCHEDULE (WN 1, 2; G 1)	

ROLES REF

NO ROLES

FRAMES REF KEY

NP V NP.BENEFICIARY NP

EXAMPLE "David dug me a hole."

SYNTAX AGENT V BENEFICIARY RESULT

SEMANTICS NOT(EXIST(START(E), RESULT)) EXIST(RESULT(E), RESULT) CAUSE(AGENT, E) BENEFIT(E, BENEFICIARY)

Figure 2.11. Membres, Rôles et Frames de la sous-classe "create-26.4-1"

Dans PropBank, le prédicat (verbe) a un ensemble d'arguments (constituants) et ses arguments possèdent une étiquette sémantique. Les annotations sémantiques sont basées sur la structure syntaxique de Penn Treebank [82]. Chaque sens de l'attribut (verbe) a un ensemble de rôles sémantiques, afin de démunir le problème d'une liste universelle de rôles sémantiques. Par rapport au FrameNet, les rôles dans PropBank sont avec des nombres `ARG0`, `ARG1`, `ARG2`, `etc.`, la signification fréquemment utilisée pour chacun des rôles est donnée dans le tableau (2.5) [10].

Tableau 2.5. Rôles sémantiques numérotés dans PropBank

Rôle sémantique	Signification
ARG0	Agent
ARG1	Patient
ARG2	Bénéficiaire, instrument, attribut ou état final
ARG3	Point de départ, bénéficiaire, instrument ou attribut
ARG4	Point final

Il existe d'autres types de rôles sémantiques pour caractériser le temps, manière, locative, etc., ces rôles sont symbolisés par `ARGM` (`ARGM-TMP`, `ARGM-MNR`, `ARGM-LOC`, `etc.`). L'exemple qui suit donne un aperçu sur l'annotation dans PropBank [83] :

La phrase en anglais :

`Mr. Bush met him privately, in the White House, on Thursday.`
(`M. Bush l'a rencontré en privé, dans la Maison-Blanche, jeudi.`)

est annotée par : Mr. Bush met him privately, in the White House, on Thursday.

[ARG0] [ARG1] [ARGM-MNR] [ARGM-LOC] [ARGM-TMP]

Les ARGMs cités dans un guide récent de PropBank sont [84] : `COM`, `LOC`, `DIR`, `GOL`, `MNR`, `TMP`, `EXT`, `REC`, `PRD`, `PRP`, `CAU`, `DIS`, `ADV`, `ADJ`, `MOD`, `NEG`, `DSP`, `LVB`, `CXN`.

Le tableau (2.6) donne une description de certains ARGMs:

Tableau 2.6. Description générale de quelques ARGMs de PropBank anglais

Argument	Définition	Explication
ARGM-COM	Comitative	Annote la personne ou l'organisation avec qui l'action est effectuée
ARGM-LOC	Locative	Montre le lieu du déroulement de l'action
ARGM-DIR	Directionnel	Indique le mouvement
ARGM-GOL	But	Indique l'objectif de l'action
ARGM-MNR	Manière	Montre comment l'action est effectuée
ARGM-TMP	Temporel	Précise le moment d'exécution de l'action
ARGM-EXT	Ampleur	Annote l'ampleur du changement effectué par une action
ARGM-REC	Réciproque	Annote des constituants tels que himself, itself, etc., ou un constituant qui renvoie à un autre argument

PropBank est disponible pour d'autres langues [W9] : Hindi, Chinois, Finlandais, Portugais et l'Arabe. Nous consacrons une partie du chapitre suivant au PropBank Arabe, objet de notre recherche.

5- Approches et applications d'annotations

Deux catégories de ressources sont exploitées par les systèmes d'annotation des rôles sémantiques. La première appelée les inventaires donne la description des frames, rôles, prédicats, etc., et regroupe des ressources comme FrameNet et PropBank. L'autre catégorie est celle des corpus annotés selon les inventaires. Ces systèmes d'annotation sont supervisés, basés sur la connaissance, semi-supervisée ou non supervisée [57].

La majorité des systèmes d'annotation des rôles sémantiques utilisent un apprentissage supervisé. Ils exploitent FrameNet et PropBank comme corpus [65]. Selon C. Mouton [85], les features proposés par D. Gildea et D. Jurafsky dans [6] sont utilisés dans de nombreux systèmes supervisés.

Nous citons quelques travaux présentés par des chercheurs dans [86]. Un modèle génératif est utilisé dans [87], dans [88], les auteurs utilisent une approche de maximum entropie, une approche basée sur la machine à vecteurs de support est employée par A. Giuglea et A. Moschitti [89]. Ces travaux et autres, comme dans [90], [91] utilisent principalement FrameNet. La célèbre conférence CoNLL a consacré les

années 2004 et 2005 à l'annotation des rôles sémantiques basée sur PropBank [7], [8]. Aussi, certaines tâches de la conférence SemEval-2007 ont été dédiées à l'annotation des rôles sémantiques.

Un grand corpus est nécessaire pour les approches supervisées. Pour l'anglais, il y a deux corpus, FrameNet et PropBank, mais pour d'autres langues, de tels corpus sont rares [85] ou ne contiennent pas un nombre important d'annotations. Des méthodes non supervisées utilisent les informations lexicales pour annoter [65]. Selon C. Mouton [85], la première approche non supervisée a été présentée par R. S. Swier et S. Stevenson [92] et le travail de H. Fürstenau et M. Lapata [93] est l'un des premiers travaux basés sur un apprentissage semi-supervisé.

6- Intérêt des rôles sémantiques pour le TALN

Parmi les avantages de l'annotation des rôles sémantiques, c'est son accord avec d'autres domaines du traitement du langage naturel, d'où l'intérêt de son utilisation dans de nombreux domaines [57]. Elle donne un apport à la recherche d'information, l'extraction d'information, les questions/réponses, le résumé automatique, etc. [65] et aussi dans la traduction automatique.

Des travaux de résumé automatique sont cités dans [65]. Dans [94], l'utilisation des rôles sémantiques vient pour perfectionner le résultat d'une méthode statistique pour le résumé de texte. De même, en 2005 puis en 2006, les travaux de G. Melli emploient les rôles sémantiques pour le résumé automatique [95], [96].

Dans [97], les auteurs ont conclu que l'utilisation des rôles sémantiques est très utile pour l'extraction de réponses. Dans [98], les auteurs utilisent la structure prédicat/argument et les frames sémantiques pour un système de questions/réponses. Également, R. Sun et al. se servent d'un système d'annotation selon PropBank afin d'employer la sémantique dans une application de questions/réponses [99]. Toujours dans le même domaine d'application D. Shen et M. Lapata utilisent les rôles sémantiques de FrameNet [100].

A. Lakhfif et M. T. Laskri utilisent les rôles sémantiques pour la traduction automatique entre l'arabe et le langage des signes [101]. Le projet de FrameNet allemand basé sur le FrameNet anglais montre l'utilité d'un tel lexique pour la traduction automatique [102]. Aussi, certains travaux cités dans [57], montrent l'utilisation des rôles sémantiques dans un système statistique de traduction automatique (`string-to-tree`) [103], [104].

Selon Q. Pradet des domaines d'application moins ordinaires ont tiré profit des rôles sémantiques [57], comme l'évaluation de la traduction [105], [106], la détection de plagiat [107], prévoir le mouvement du cours des actions [108], le marketing pour la recommandation de livres [109], l'interprétation des recettes de cuisine [110].

7- Conclusion

L'annotation des rôles sémantiques est une tâche très importante dans divers domaines du traitement du langage naturel, puisqu'elle participe clairement et essentiellement dans la compréhension. L'importance des rôles sémantiques a poussé informaticiens et linguistes à élaborer des théories, formalismes, lexiques, systèmes, etc. pour mieux comprendre le sens des phrases d'une manière informatisé.

Nous remarquons que les travaux sur l'annotation des rôles sémantiques sont focalisés sur l'anglais comme la majorité des travaux dans le traitement automatique du langage naturel. Le chapitre qui suit est consacré aux travaux d'annotation dans la langue arabe, afin de donner, un panorama satisfaisant sur ce domaine dans la langue arabe.

Chapitre 3 :

Annotation des Rôles Sémantiques dans la Langue Arabe

Chapitre 3 : Annotation des Rôles Sémantiques dans la Langue Arabe

Chapitre 3 : Annotation des Rôles Sémantiques dans la Langue Arabe

1- Introduction

Dans le précédent chapitre, nous avons donné une large description sur l'annotation des rôles sémantiques, les formalismes de représentation de sens, les systèmes d'annotations, etc., concernant la langue anglaise.

Nos travaux de recherche sont orientés vers la langue arabe moderne, nous consacrons ce chapitre aux travaux sur la langue arabe qui essayent d'améliorer les aspects d'annotation de rôles sémantiques, vu le manque considérable de travaux dans ce domaine pour cette langue.

Nous listons les corpus arabes annotés et plusieurs types d'annotations, puis les formalismes de représentation de sens pour l'arabe sont décrits. Nous terminons par la description des deux travaux d'annotation existants.

2- Corpus arabes annotés

W. Zaghouani donne une bonne récapitulation des corpus arabes annotés nécessaire au développement des outils et des systèmes supervisés [111]. Dans ce qui suit, nous citons les corpus donnés par cet auteur qui les divise en trois catégories (entités nommées, erreurs annotées et autres).

2.1- Corpus d'entités nommées

Nous commençons par les corpus de reconnaissance d'entités nommées (Tableau 3.1). Selon cet auteur : les noms de personnes, organisations et lieux géographiques sont les principaux objets couverts par ces corpus, qui varient entre 14K et 230K.

2.2- Corpus d'erreurs annotées

Parmi les utilisations de ce genre de corpus, la construction d'outils pour la correction automatique d'orthographe. Le tableau 3.2 montre trois corpus annotés d'erreurs, dont le nombre de mots varie entre 65K et 2M de mots.

2.3- Autres corpus

Des corpus annotés par des informations syntaxiques, morphologiques ou sémantiques sont présentés dans le tableau 3.3.

Tableau 3.1. Corpus annotés par les entités nommées [111]

Source	Corpus	Mots
R. Steinberger et al. [112]	JRC-Names	230 000
Y. Benajiba et al. [113]	ANERCorp	150 000
B. Mohit et al. [114]	AQMAR Named Entity Corpus[6]	74 000
M. Azab et al. [115]	Named Entity Translation Lexicon[7]	55 000
M. Attia et al. [116]	Named Entities List[8]	45 202
Y. Benajiba et al. [113]	ANERGazet	14 000

Tableau 3.2. Les corpus d'erreurs annotées [111]

Source	Corpus	Mots
N. Habash et al. [117]	Qatar Arabic language Bank(QALB)[9]	2 000 000
A. Alfaifi et al. [118]	Arabic Learner Corpus[10]	282 000
M. Alkanhal et al. [119]	KACST Error Corrected Corpus[11]	65 000

Tableau 3.3. Autres corpus annotés [111]

Source	Corpus	Mots	Type
W. Ralph et al. [120]	OntoNotes Release 5.0[12]	300 000	Sémantique
K. Dukes et N. Habash [121]	The Quranic Arabic Corpus[13]	77 430	Part of Speech / Syntaxique
N. Schneider et al. [122]	AQMAR Arabic Wiki. Supersense Corpus[14]	65 000	Sémantique
K. Shereen et al. [123]	Khoja POS tagged corpus[15]	51 700	POS
E. Mohammed (référence non disponible dans le document source)	Arabic Wikipedia Dependency Corpus[16]	36 000	Syntaxique
S. Hammami et al. [124]	AnATAr Corpus17	18 895	Anaphora

[6] http://www.ark.cs.cmu.edu/ArabicNER/
[7] http://nlp.qatar.cmu.edu/resources/NETLexicon/
[8] https://sourceforge.net/projects/arabicnes/
[9] http://nlp.qatar.cmu.edu/qalb/
[10] http://www.comp.leeds.ac.uk/scayga/alc/corpus%20files.html
[11] http://cri.kacst.edu.sa/Resources/TST_DB.rar
[12] http://catalog.ldc.upenn.edu/LDC2013T19
[13] http://corpus.quran.com/download/
[14] http://www.ark.cs.cmu.edu/ArabicSST/
[15] http://zeus.cs.pacificu.edu/shereen/research.htm#corpora and email the author
[16] http://www.ark.cs.cmu.edu/ArabicDeps/
[17] https://sites.google.com/site/anlprg/outils-et-corpus-realises/AnATArcorpus-BEB.rar?attredirects=0

3- Formalismes de représentation de sens pour la langue arabe

3.1- WordNet

La réussite de Princeton WordNet pour l'anglais donne un encouragement au développement d'un WordNet pour d'autres langues [125]. En se basant sur Princeton WordNet anglais [126], S. ElKateb et al. présentent le challenge de la construction d'un WordNet[18] pour l'arabe standard moderne (`Modern Standard Arabic`) [125]. Pour cela, ils ont suivis la démarche employée pour la construction de l'EuroWordNet [127]. Une approche basée sur les réseaux bayésiens est employée pour l'extension semi-automatique de l'arabe WordNet [128].

3.2- FrameNet

Dans le site consacré au FrameNet [19] de l'université Berkeley en Californie, nous trouvons des informations concernant FrameNet pour d'autres langues que l'anglais: français, chinois, portugais brésilien, allemand, espagnol, suédois et coréen. Cependant, aucune référence pour la langue arabe n'est disponible. Une recherche dans la littérature nous a permis de trouver des contributions récentes pour la construction d'un FrameNet arabe. Cependant, ces contributions n'arrivent pas au stade d'un FrameNet anglais et ne sont pas mises en avant pour être reconnues par la communauté internationale du TALN.

Une première contribution est faite pour la construction d'un FrameNet des verbes arabes du Coran, avec une possible extension vers d'autres prédicats [129]. De même, pour l'élaboration d'un FrameNet arabe qui contient des informations syntaxiques et sémantiques, les auteurs présentent une méthodologie pour la construction d'une telle ressource [130].

Dans leur travail A. Lakhfif et M. T. Laskri ont construit une base de données FrameNet arabe, qui contient 4.006 verbes, 4.113 noms, 230 adjectifs, 12 adverbes, 600 frames lexicales, 10k unités lexicales et 2k phrases annotées [101].

3.3- Treebank

L'importance de Penn TreeBank dans le traitement du langage naturel et la linguistique informatique oblige la réalisation d'un tel type d'annotation pour la langue arabe. Selon M. Maamouri et al., la méthode de développement du Penn TreeBank pour d'autres langues aide profondément le développement d'une telle ressource pour la langue arabe, bien sûr, avec une prise de considération des différences entre

[18] http://globalwordnet.org/arabic-wordnet/
[19] https://framenet.icsi.berkeley.edu/fndrupal/framenets_in_other_languages

l'arabe et les autres langues [131]. Leurs données sont extraites à partir d'agences de presse : Agence France Presse, Ummah (Al-Hayat) et An-Nahar.

Les annotations morphologiques, syntaxiques, parties de discours, etc., sont les informations disponibles dans l'arabe Penn TreeBank [132]. Ces annotations ont un apport considérable pour le développement d'applications de traitement du langage naturel pour la langue arabe. Les outils d'analyse automatisés ont donné un grand avancement à l'équipe de l'arabe TreeBank pour mettre, dans une période d'une année, la première base de données d'informations morphologique et syntaxique [132].

Nous avons organisé dans le tableau ci-dessous (Tableau 3.4) les versions de Penn TreeBank disponibles dans le site du laboratoire "The Linguistic Data Consortium (LDC)[20]" de l'université Pennsylvanie. Nous n'avons pas pris en considération les ressources parallèles entre l'arabe et l'anglais.

Tableau 3.4. Inventaire de l'arabe TreeBank à partir de LDC

Nom	Date	Référence[21]
Arabic Treebank: Part 1 v 2.0	03 Février 2003	LDC2003T06
Arabic Treebank: Part 1 – 10K-word English Translation	25 Février 2003	LDC2003T07
Arabic Treebank: Part 2 v 2.0	30 Janvier 2004	LDC2004T02
Arabic Treebank: Part 3 v 1.0	21 Mai 2004	LDC2004T11
Prague Arabic Dependency Treebank 1.0	19 Novembre 2004	LDC2004T23
Arabic Treebank: Part 1 v 3.0 (POS with full vocalization + syntactic analysis)	15 Février 2005	LDC2005T02
Arabic Treebank: Part 3 (full corpus) v 2.0 (MPG + Syntactic Analysis)	15 Juin 2005	LDC2005T20
Arabic Treebank: Part 1 v 4.1	16 Novembre 2010	LDC2010T13
Arabic Treebank: Part 2 v 3.1	15 Août 2011	LDC2011T09
Arabic Treebank - Broadcast News v1.0	18 Juillet 2012	LDC2012T07
Arabic Treebank - Weblog	18 Janvier 2016	LDC2016T02

[20] https://www.ldc.upenn.edu/
[21] Pour accéder à la ressource en ligne : https://catalog.ldc.upenn.edu/ + REFERENCE

La figure 3.1 montre un exemple d'une arborescence du Penn Treebank arabe de la phrase :

خطت الولايات المتحدة وبريطانيا خطوة جديدة في حربهما...

Les États-Unis et la Grande-Bretagne ont fait un nouveau pas dans leur guerre...

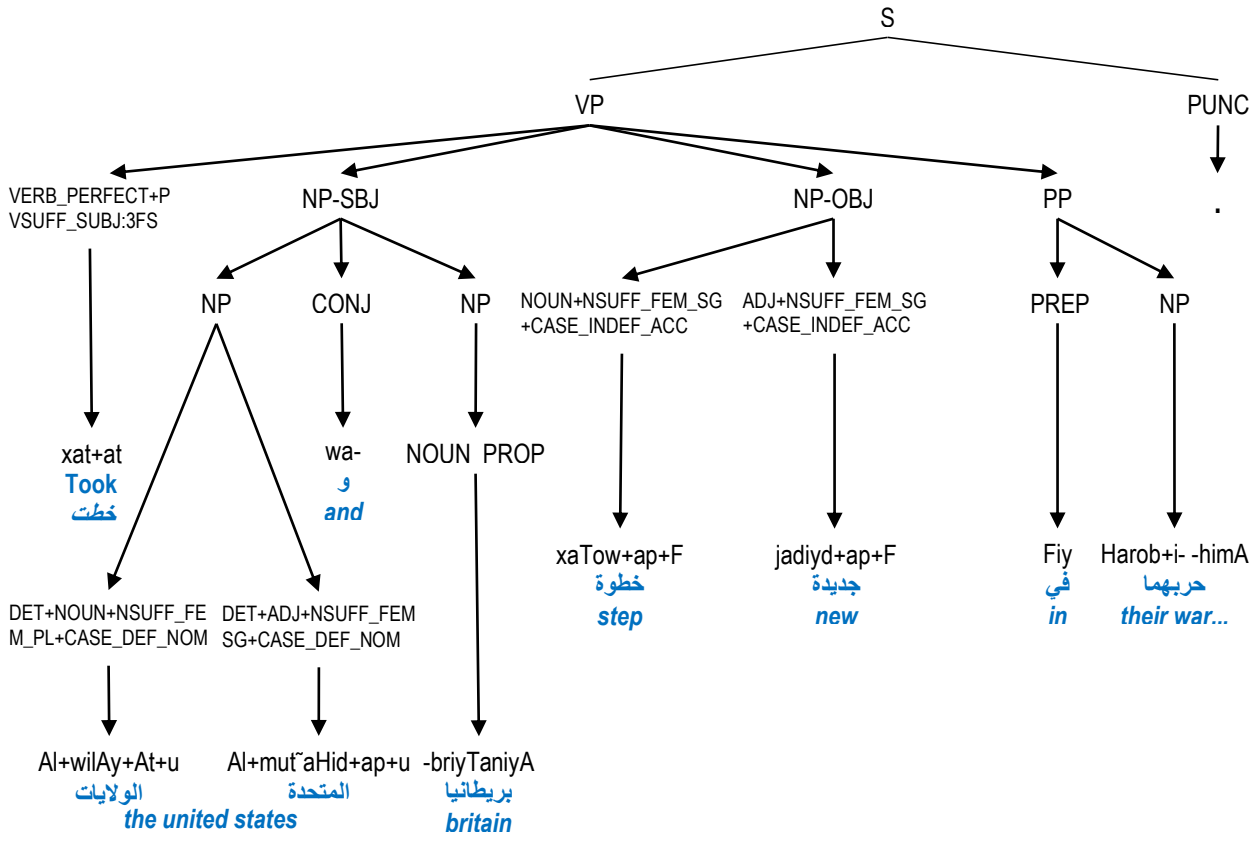


Figure 3.1. Analyse arborescente à partir du Penn TreeBank arabe [133]

Un exemple extrait d'OntoNotes 5.0 (principale ressource du corpus utilisé dans notre travail) de la phrase arabe ci-dessous est présenté dans la figure 3.2.

وَقالَ وَزِيرُ الدِفاعِ أَنْجِيلُو ريس أَنَّ نَحْوَ سِتَّةِ مَسْؤُولِينَ أَمِيرْكِيِّينَ وَصَلُوا إِلَى البِلادِ لِلبَحْثِ فِي التَدابِيرِ اللُوجِسْتِيَّةِ، عَلَى أَن يَلْحَقَ بِهِم الآخَرُونَ تِباعاً، بَدْءاً مِن اليَوْم.

Et le secrétaire de la défense, Angelo Reyes, a déclaré que six responsables américains sont arrivés au pays pour discuter des mesures logistiques, et ils seront suivis successivement par d'autres à partir d'aujourd'hui.

```
(TOP (S (CONJ وَ-)
        (VP (PV+PVSUFF_SUBJ:3MS -قَالَ)
            (NP-SBJ (NP (NOUN+CASE_DEF_NOM وَزِيرُ)
                        (NP (DET+NOUN+CASE_DEF_GEN الدِفَاعِ)))
                    (NP (NOUN_PROP أُنْجِيلُو)
                        (NOUN_PROP رِيس)))
            (SBAR (SUB_CONJ أَنْ)
                  (S (NP-TPC-1 (NOUN+CASE_DEF_ACC نَحْوَ)
                               (NP (NOUN_NUM+NSUFF_FEM_SG+CASE_DEF_GEN سِتَّةِ)
                                   (NP (NOUN+NSUFF_MASC_PL_GEN مَسْؤُولِينَ)
                                       (ADJ+NSUFF_MASC_PL_GEN أَمِيرْكِيِّينَ))))
                     (VP (PV+PVSUFF_SUBJ:3MP وَصَلُوا)
                         (NP-SBJ (-NONE- *T*-1))
                         (PP-DIR (PREP إِلَى)
                                 (NP (DET+NOUN+CASE_DEF_GEN الْبِلَادِ)))
                         (PP-PRP (PREP لِ-)
                                 (NP (NP (DET+NOUN+CASE_DEF_GEN -الْبَحْثِ))
                                     (PP (PREP فِي)
                                         (NP (DET+NOUN+CASE_DEF_GEN الْتَّدَابِيرِ)
                                             (DET+ADJ+NSUFF_FEM_SG+CASE_DEF_GEN الْلُّوجِسْتِيَّةِ)))))
                         (PUNC ,)
                         (PP (PREP عَلَى)
                             (SBAR (SUB_CONJ أَن)
                                   (S (VP (IV3MS+IV+IVSUFF_MOOD:S يَلْحَقَ)
                                          (PP-CLR (PREP بِ-)
                                                  (NP (PRON_3MP -هِم)))
                                          (NP-SBJ (DET+NOUN+NSUFF_MASC_PL_NOM الْآخَرُونَ))
                                          (NP-MNR (NOUN+CASE_INDEF_ACC تِبَاعاً))
                                          (PUNC ,)
                                          (NP-TMP (NP (NOUN+CASE_INDEF_ACC بَدْءاً))
                                                  (PP (PREP مِن)
                                                      (NP (DET+NOUN+CASE_DEF_GEN الْيَوْمِ))))))))))))
        (PUNC .)))
```

Figure 3.2. Phrase arabe extraite d'OntoNotes 5.0

Répertoire : ontonotes-release-5.0_LDC2013T19\ontonotesrelease5.0\data\files\data\arabic\annotations\nw\ann\00\ann_0061.onf

Le Penn TreeBank arabe est en relation directe avec notre travail d'annotation. Pour cela, nous avons cité principalement le TreeBank de l'université de Pennsylvanie, cependant, il y a d'autres contributions TreeBank.

Le travail sur l'arabe TreeBank à l'université de Leeds au Royaume-Uni est un projet unique dans la littérature, car il se concentre sur la langue du saint Coran et vise la réalisation d'un TreeBank pour la grammaire arabe traditionnelle [134]. Le site web « http://corpus.quran.com » consacré au corpus coranique contient la contribution syntaxique TreeBank et deux autres niveaux : annotation morphologique et ontologie sémantique. Aussi, nous citons un TreeBank [135] de l'université Columbia à New York (Figure 3.3), un TreeBank pour l'arabe égyptien [136], un TreeBank arabe de dépendance universelle [137] et le travail de D. Halabi et al. est une contribution récente qui montre les premières démarches pour la construction d'une nouvelle ressource TreeBank [138].

Dans [139], les auteurs discutent de différents TreeBanks arabes et leurs caractéristiques, avec l'objectif de construire leur propre ressource.

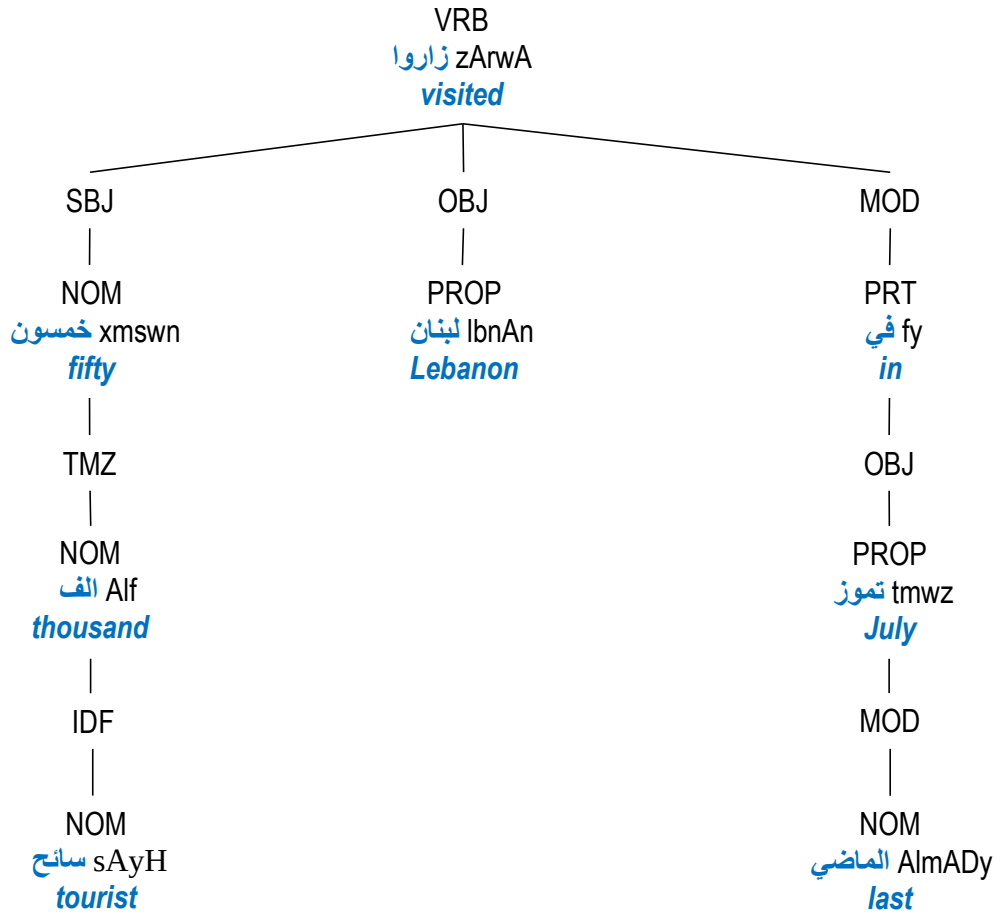


Figure 3.3. Annotation de la phrase arabe [135]

خمسون ألف سائح زاروا لبنان في تموز الماضي

Cinquante mille touristes ont visité le Liban en juillet dernier

3.4- PropBank

La première contribution pour la réalisation d'un PropBank arabe est celle de M. Palmer et al. [140]. Ils suivent le même procédé du développement de PropBank pour d'autres langues que l'arabe. Sachant que l'arabe est une langue sémitique qui diffère des autres langues non sémitiques comme l'anglais, il est nécessaire de porter attention à ses différences lors du suivi du procédé de développement. Sans l'arabe TreeBank [131] et l'analyse morphologique de la langue arabe [141], la construction d'une telle ressource est impossible [140]. Les fichiers de frames et le corpus annoté sont des constituants de PropBank arabe [142]. Il consiste à la construction des framesets pour les verbes, puis ces framesets sont utilisés par les annotateurs. Le prédicat et ses éventuels arguments sont distingués par le frameset [140].

Des changements ont été portés sur le Penn TreeBank arabe [143], ce qui a nécessité une adaptation de PropBank arabe [144]. Lors de cette adaptation, les auteurs utilisent l'outil Cornerstone [145] pour la création des framesets. Auparavant, la tâche de création de framesets en format XML était manuelle et prends un temps considérable. Avec cet outil, la création de framesets est plus rapide et ne nécessite pas des connaissances en XML. Aussi, l'outil Jubilee [146] est utilisé pour l'amélioration de l'annotation, le travail donne un total de 1.955 fichiers de frame et 2.446 framesets.

Les deux outils précédents sont utilisés pour un PropBank coranique [147]. Ce travail montre la possibilité de créer un lexique, annoter le sens des verbes et mettre des informations sémantiques en suivant un modèle PropBank. Dans ce travail, les auteurs créent les fichiers de frames de 50 verbes du TreeBank coranique de Dukes et Buckwalter, cité précédemment [134]. C'est une ressource importante pour les recherches du traitement du langage pour la langue arabe, car elle traite la sémantique d'un texte classique de grande importance [147].

W. Zaghouani auteur dans les trois projets PropBank [140], [144], [147] donne une liste d'arguments de PropBank arabe (Tableau 3.5). La figure 3.4 montre un exemple d'annotation PropBank de la phrase arabe :

بدأ رئيس الوزراء الصيني زو رونجي زيارة رسمية للهند الاحد الماضي

Le Premier ministre chinois Zhu Rongjy a entamé une visite officielle en Inde dimanche dernier

Tableau 3.5. Argument PropBank arabe [148]

Arguments	Fonction	Arguments	Fonction
ARG-0	Agent	ARGM-BNF	Bénéficiaire
ARG-1	Patient	ARGM-CAU	Causale
ARG-2	Instrument, bénéficiaire ou attribut	ARGM-CND	Condition
ARG-3	Point de départ	ARGM-DIR	Direction
ARG-4	Point d'arrivée	ARGM-DIS	Discours
ARGM-ADV	Adverbiale	ARGM-EXT	Degré
ARGM-LOC	Lieu	ARGM-PRD	Prédicatif
ARGM-MNR	Manière	ARGM-PRP	But
ARGM-NEG	Négation	ARGM-REC	Réciproque
ARGM-TMP	Temporel		

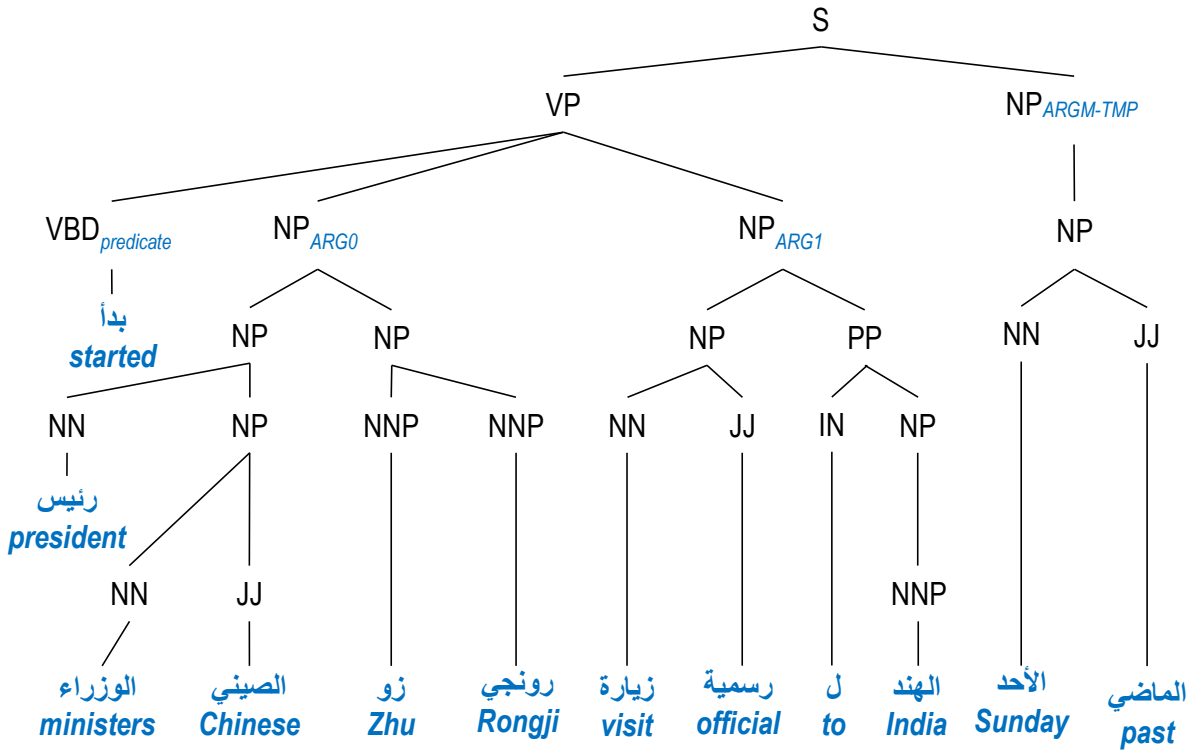


Figure 3.4. Exemple d'annotation de PropBank arabe [14]

Les figures 3.5 (a, b, c et d) montrent la frameset de l'attribut (prédicat) «بَدَأ» «a commencé» "badaO-a [22] ". La figure 3.6 montre l'arbre de la phrase arabe ci-dessous et la figure 3.7 montre le verbe (attribut) «تَبْدَأ » et ses rôles sémantiques (arguments) PropBank.

و من المقرر ان تبدأ التدريبات هذا الشهر، و قد تستمر حتى نهاية هذه السنة في مدينة زامبوانغا و جزيرة باسيلان المجاورة لها حيث تشن القوات الفيليبينية حملة واسعة على جماعة " ابو سياف " التي تحتجز رهائن، بينها أميركيان منذ حزيران الماضي.

Les exercices devraient commencer ce mois-ci et peuvent se poursuivre jusqu'à la fin de cette année dans la ville de Zamboanga et son île voisine de Basilan, où les forces philippines mènent une vaste répression contre "Abu Sayyaf", qui détient des otages, dont deux Américains, depuis juin dernier.

[22] https://verbs.colorado.edu/propbank/framesets-arabic/badaO-a-v.html

Roleset id : 01, *to start, begin action (agent known)*

Arg0 : *agent – starter*
Arg1 : *thing started*
Arg2 : *instrument, manner – frequently marked by* ب

Figure 3.5.a. Frameset du prédicat badaO-a

Frame:

SBJ is ARG1

```
(S
  (CONJ و)
  (PP-PRD
    (PREP من)
    (NP
      (DET+NOUN+CASE_DEF_GEN المقرر)))
  (SBAR-SBJ
    (SUB_CONJ ان)
    (S
      (S
        (VP
          (IV3FS+IV+IVSUFF_MOOD:S تبدأ)
          (NP-SBJ
            (DET+NOUN+NSUFF_FEM_PL+CASE_DEF_NOM التدريبات))
          (NP-TMP
            (NP
              (DEM_PRON_MS هذا))
            (NP
              (DET+NOUN+CASE_DEF_ACC الشهر))))))
      (PUNC .))
```

Arg1 : التدريبات
Gloss: the trainings
Argm-tmp : هذا الشهر
Gloss: this month

Figure 3.5.b. Frameset du prédicat badaO-a

```
SBJ is ARG0


(S
  (CONJ و)
  (VP
    (PV+PVSUFF_SUBJ:3MS كان)
    (NP-1
      (NOUN_PROP عارف))
    (VP
      (PV+PVSUFF_SUBJ:3MS بدأ)
      (NP-SBJ
        (-NONE- *T*-1))
      (NP-TMP
        (DET+NOUN+CASE_DEF_ACC الاربعاء))
      (NP-OBJ
        (NP
          (NOUN+NSUFF_FEM_SG+CASE_INDEF_ACC زيارة))
        (PP-LOC
          (PREP ل)
          (NP
            (NP
              (NOUN_PROP سوريا))
            (SBAR
              (WHNP-2
                (-NONE- 0))
              (S
                (VP
                  (PV+PVSUFF_SUBJ:3FS استمرت)
                  (NP-SBJ
                    (-NONE- *T*-2))
                  (NP-TMP
                    (NOUN_NUM+NSUFF_FEM_SG+CASE_DEF_ACC ثلاثة)
                    (NP
                      (NOUN+CASE_INDEF_GEN أيام)))))))))))
  (PUNC .))
```

Arg0 : *T*-1 < عارف
Gloss: -NONE- < 'Aarif
Argm-tmp : الاربعاء
Gloss: Wednesday
Arg1 : زيارة لسوريا استمرت ثلاثة أيام .
Gloss: a visit to Syria that lasted 3 days

Figure 3.5.c. Frameset du prédicat badaO-a

```
(S
  (CONJ و)
  (NP-TPC-1
    (PRON_3MS هو))
  (VP
    (PV+PVSUFF_SUBJ:3MS كان)
    (VP
      (PV+PVSUFF_SUBJ:3MS بدأ)
      (NP-SBJ
        (-NONE- *T*-1))
      (NP-OBJ
        (NOUN+CASE_DEF_ACC يوم)
        (NP
          (POSS_PRON_3MS ه)))
      (PP-MNR
        (PREP ب)
        (NP
          (NOUN+CASE_DEF_GEN لقاء)
          (NP
            (NP
              (NP
                (NOUN+NSUFF_MASC_PL_GEN ناشطين))
              (PP
                (PREP في)
                (NP
                  (NOUN+CASE_DEF_GEN حقوق)
                  (NP
                    (DET+NOUN+CASE_DEF_GEN الانسان)))))
            (SBAR
              (WHNP-2
                (-NONE- 0))
              (S
                (VP
                  (PV+PVSUFF_SUBJ:3MD امضيا)
                  (NP-SBJ
                    (-NONE- *T*-2))
                  (NP-OBJ
                    (NOUN+NSUFF_FEM_SG+CASE_INDEF_ACC فترة))
                  (PP-LOC
                    (PREP في)
                    (NP
                      (DET+NOUN+CASE_DEF_GEN السجن)))))))))))
  (PUNC .))
```

Arg0 : *T*-1 < هو
Gloss: -NONE- < he
Arg1 : يومه
Gloss: his day
Arg2 : بلقاء ناشطين في حقوق الانسان امضيا فترة في السجن
Gloss: meeting two human rights activists who had spent some time in prison

Figure 3.5.d. Frameset du prédicat badaO-a

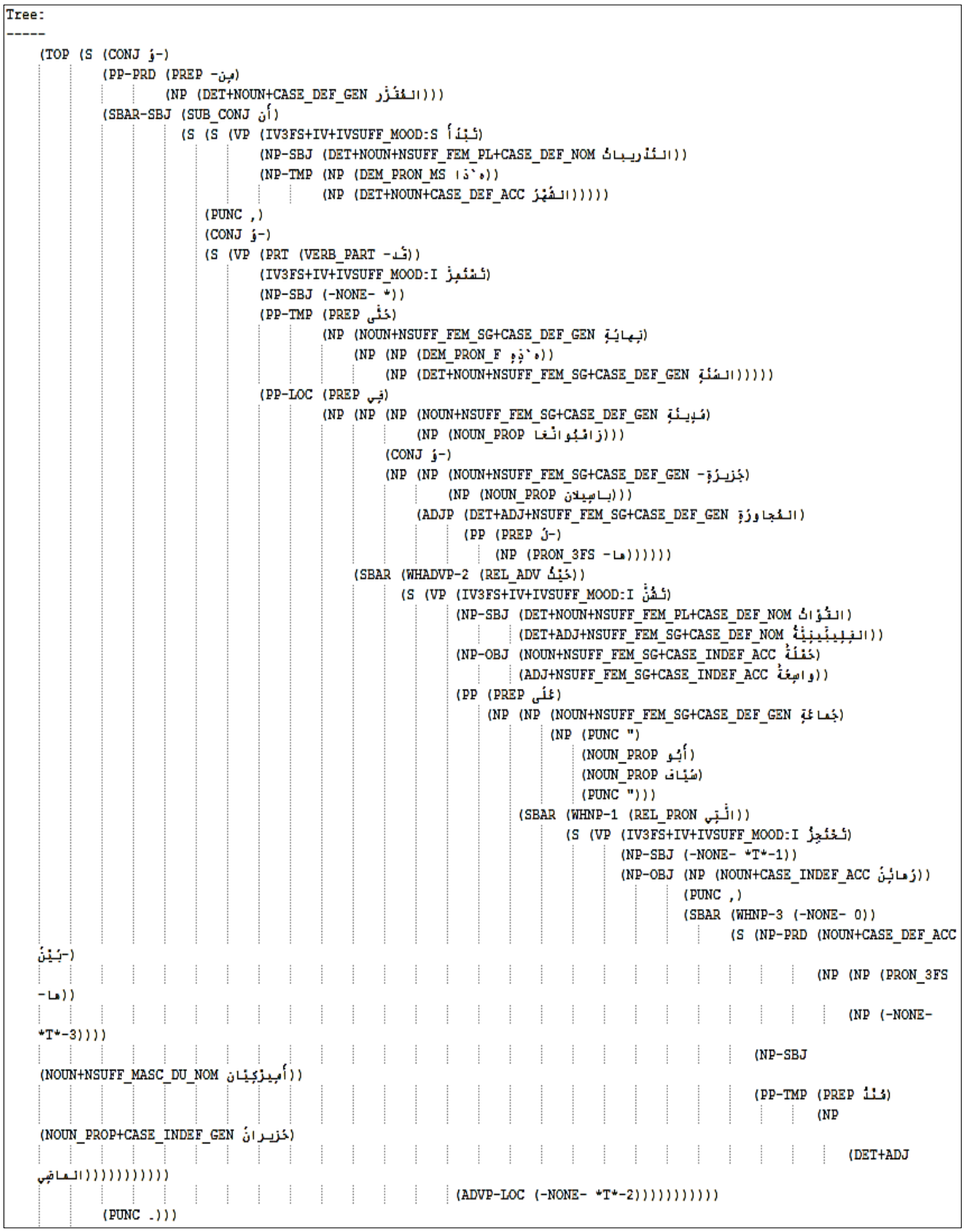


Figure 3.6. L'arbre d'une phrase arabe qui contient le verbe تُنْبِئَ

Extraite de OntoNotes 5.0

Répertoire : ontonotes-release-5.0_LDC2013T19\ontonotes-release5.0\data\files\data\arabic\annotations\nw\ann\00\ann_0001.onf

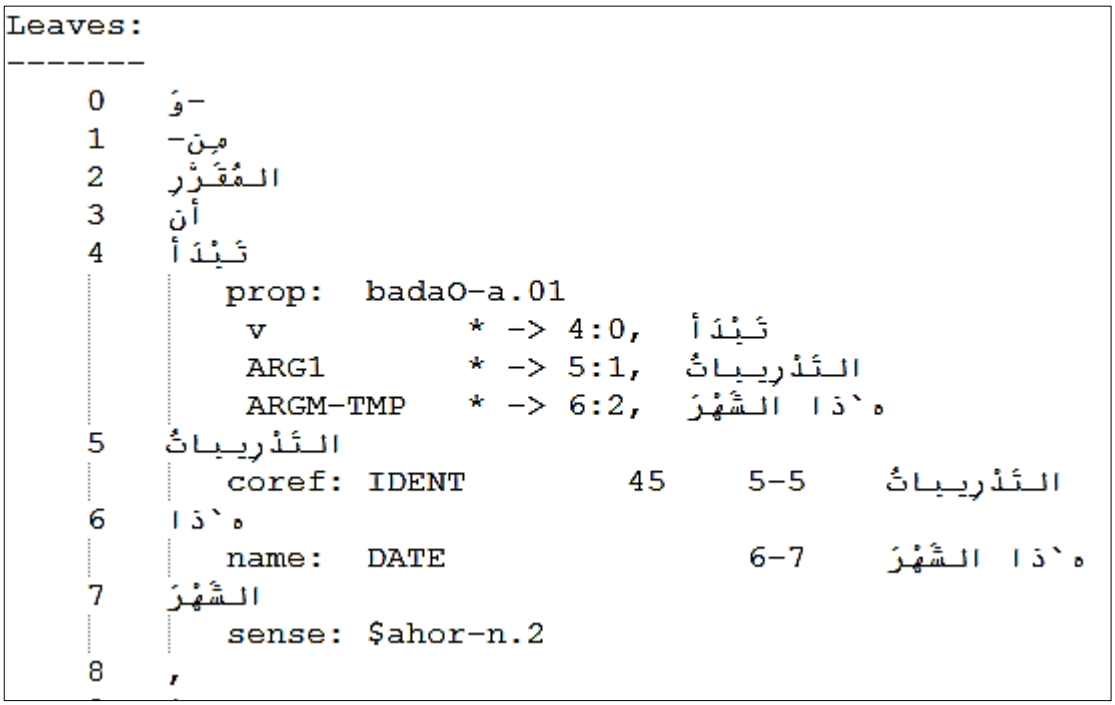


Figure 3.7. L'attribut تَبْدَأ et ses arguments

Extraits de OntoNotes 5.0

Répertoire : ontonotes-release-5.0_LDC2013T19\ontonotes-release5.0\data\files\data\arabic\annotations\nw\ann\00\ann_0001.onf

3.5- VerbNet

Le travail de J. Mousser [149] sur l'arabe VerbNet adapte à la langue arabe, la classification des verbes en anglais par les classes de Levin dans [79] et [150], en suivant les étapes de construction de K. K. Schuler [80].

Dans son travail, les caractéristiques sémantiques et syntaxiques communes entre les verbes sont exploitées pour classer les verbes dans des classes et certains verbes sont organisés dans des sous-classes. Il fait usage des 23 rôles thématiques utilisés dans le VerbNet anglais. Pour essayer d'enrichir la description sémantique, l'auteur associe VerbNet arabe à l'arabe WordNet [151]. Une extension de ce travail est présentée dans [149].

Actuellement, selon le site officiel de cette ressource, le VerbNet arabe[23] contient 336 classes, 7.748 verbes et 1.399 frames. La figure (3.8) montre un exemple de la classe "badaOa-1" à partir d'une recherche du verbe « بدأ » dans VerbNet arabe disponible en ligne et la figure 3.9 montre la sous-classe

[23] http://ling.uni-konstanz.de/pages/home/mousser/files/Arabic_verbnet.php

"badaOa-1.1" de la classe "badaOa-1". Récemment, il y a la possibilité de télécharger ces classes en format XML à partir du site VerbNet arabe.

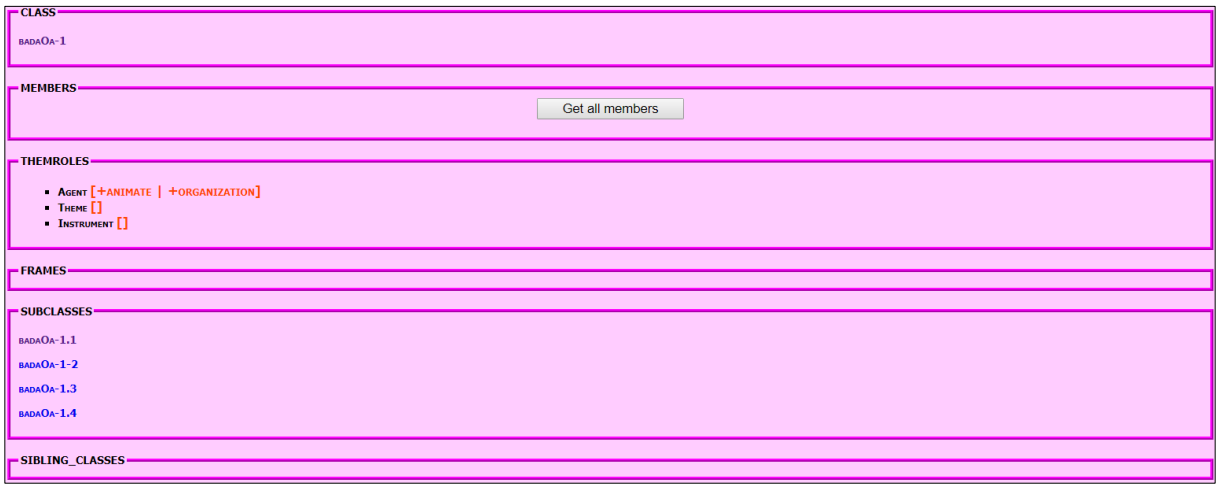


Figure 3.8. La classe "badaOa-1" dans le VerbNet arabe

Figure 3.9. La sous-classe "badaOa-1.1" dans le VerbNet arabe

4- Système d'annotation des rôles sémantiques pour l'arabe

4.1- Difficultés pour les systèmes d'annotation de la langue arabe

Deux points font du traitement de la langue arabe une tâche difficile. Premièrement, les techniques de traitement du langage naturel actuelles sont élaborées pour le traitement de la langue anglaise. Le deuxième point est que les caractéristiques syntaxiques et morphologiques entre la langue sémitique arabe et la langue anglaise sont très distinctes [11].

L'arabe est une langue riche morphologiquement. Par exemple, un mot dans la langue arabe est composé de racines et d'affixes, le mot arabe « وبحسناتهم » « et par leurs vertus » est composé de quatre parties [12].

هم	حسنات	بـ	و
leurs	vertus	par	et
Pronom possessif	Tige	Préposition	Conjonction

Un verbe arabe ou anglais exprime le temps, la voix et la personne, tandis que dans l'arabe un verbe exprime aussi des marqueurs d'humeur : subjonctives, indicatives et jussif.

Nous citons quelques difficultés données par M. Diab et Y. Marton [142] auxquelles se confronte le développement d'une méthode d'annotation des rôles sémantiques pour la langue arabe en particulier et les langues sémitiques en général :

- **Pro-drop (تسمح بسقوط الفاعل** [W10]**):** dans la langue arabe, le sujet d'un verbe peut être marqué morphologiquement sur le verbe et n'apparait pas comme constituant de la phrase. Dans la phrase « اكلوَا البرتُقَال » « Ils ont mangé les oranges », le pronom "Ils" n'est pas un constituant de la phrase arabe. Vu le manque d'un argument caractérisé par un mot, cette particularité complique la tâche du système d'annotation des rôles sémantiques.

- **Ordre libre des mots :** l'organisation d'une phrase peut être "Verbe Sujet Objet", "Sujet Verbe Objet", "Objet Verbe Sujet", etc., mais habituellement l'ordre des mots dans la langue arabe est "Verbe Sujet Objet".

- **Construction possessive :** l'une de ces manifestations dans la langue arabe est (الاضافة). Cette forme linguistique peut contenir plusieurs mots et modificateurs dans les langues sémitiques, ce qui pose problème d'identification des limites d'un argument. Exemple : « صديق إبن أخت رئيس الشركة » « Ami du neveu du président de l'entreprise ».

- **Diacritiques :** dans la phrase « رجل البيت الكبير » peut avoir deux significations « l'homme de la grande maison » ou « le grand homme de la maison ». Les diacritiques jouent un rôle important pour donner la bonne signification. Pour la première signification de la phrase précédente, les diacritiques de la phrase devront être « رجلُ البيتِ الكبيرِ » le modificateur « الكبيرِ » « grande » se porte sur « البيتِ » « maison ». Cependant pour la deuxième signification la phrase arabe sera : « رجلُ البيتِ الكبيرُ ».

Pour les langues sémitiques, selon M. Diab et Y. Marton [142], il y a que certaines contributions de systèmes d'annotation des rôles sémantiques pour la langue arabe et qu'ils utilisent l'arabe PropBank. Dans les recherches menées dans le cadre de notre thèse, nous avons recensé deux travaux seulement sur l'annotation des rôles sémantiques dans la langue arabe.

4.2- Systèmes d'annotation pour la langue arabe

La première contribution pour l'annotation des rôles sémantiques dans la langue arabe est présentée lors de SemEval-2007[24], appliquée sur l'arabe standard moderne sans les diacritiques. Elle est décrite dans [11] et elle se compose de deux sous-tâches, l'une d'elles est l'annotation des rôles sémantiques et elle est divisée en deux parties : la détection des limites des arguments et la classification de ces arguments. Leur travail est basé sur l'arabe TreeBank [131] , l'arabe WordNet [125] et l'arabe PropBank. La figure 3.10 montre un arbre syntaxique avec les annotations selon l'arabe PropBank.

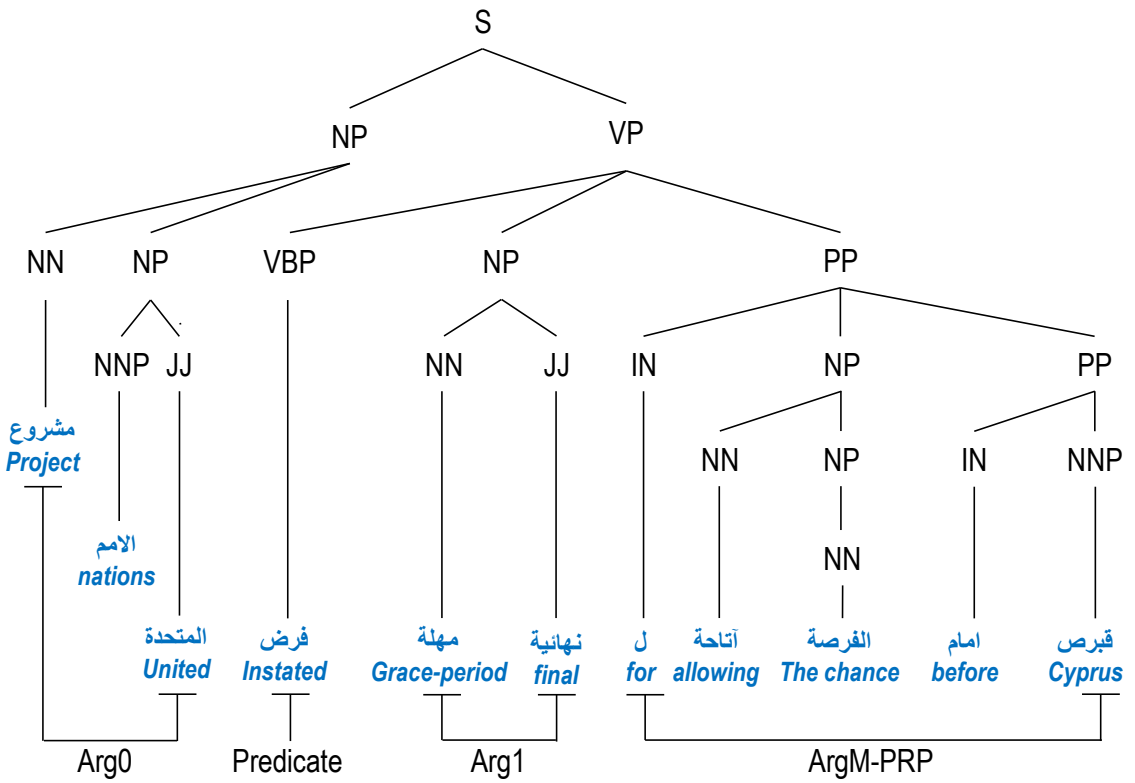


Figure 3.10. Arbre syntaxique annotée en rôles sémantiques [12]

مشروع الامم المتحدة فرض مهلة نهائية لإتاحة الفرصة امام قبرص

Projet de Nations Unies a imposé un délai final pour permettre une opportunité à Chypre

[24] http://web.archive.org/web/20080727062358/http://nlp.cs.swarthmore.edu/semeval/index.php

Ce système d'annotation des rôles sémantiques est présenté dans [13]. Il est basé sur les machines à vecteur de support (`Support Vector Machines – SVM`). Ils furent inspirés par d'autres travaux [12], [152], [153] pour le choix des features du système.

Les données sont celles de l'Arabe PropBank et présentées dans [11], elles sont réparties en trois parties : développement, entrainement et test. La partie développement contient 886 phrases et 1 725 arguments, la partie entrainement contient 8 402 phrases et 21 194 arguments et la partie test contient 902 phrases et 1 661 arguments, et d'un nombre total de 700k de données d'entrainement. Cependant, ils n'utilisent que 350k pour la détection des limites d'arguments [13]. Pour un total de 26 rôles sémantiques. Ils obtiennent un F_1 (F-mesure) de 94.06% pour la détection et 81.43% pour la classification. Un rappel de 78.3% et une précision de 84.71%. Les résultats sont donnés en détail dans [13].

Une autre contribution sur l'annotation des rôles sémantiques arabe est présentée par les mêmes auteurs de la précédente contribution [14]. Ils visent à exploiter les caractéristiques morphologiques de la langue arabe pour améliorer leur système d'annotation. Pour cela, ils utilisent machine à vecteur de support [154] et astuce de noyau[25] (`Kernel Methods`) [155].

L'idée de ce travail est basée sur l'apport des relations entre la syntaxe et la morphologie dans les langues riches morphologiquement, afin de perfectionner les systèmes d'annotation des rôles sémantiques dans ce type de langues [14]. Certains features en relation avec la morphologie sont ajoutés tels que nombre, genre, humeur, forme de Lemme, etc., ils ajoutent en tout dix nouveaux features. Les données utilisées sont celles de [11], [13], ils obtiennent une F_1 (F-mesure) de 82.17% meilleure par rapport à leur précédent travail.

5- Conclusion

Parmi les nombreuses difficultés auxquelles est confronté le chercheur qui travaille sur la langue arabe, nous trouvons : corpus existants, l'importance des corpus, types d'annotation dans un corpus, etc.. Pour cela, nous avons présenté à travers ce chapitre un panorama essentiel sur l'annotation en général et l'annotation des rôles sémantiques, en particulier, ce qui aidera les chercheurs intéressés pour le développement et l'amélioration de l'annotation dans la langue arabe.

Durant nos travaux, nous avons remarqué qu'il y a un grand écart entre les travaux sur l'arabe et sur l'anglais. Nous comparons l'arabe à l'anglais et pas au français, car le classement de l'arabe vient dans les premiers rangs des langues les plus utilisées dans le monde avec l'anglais et au cours de nos recherches nous n'avons pas trouvé assez de travaux sur le français par rapport à l'anglais.

[25] https://fr.wikipedia.org/wiki/Astuce_du_noyau

Chapitre 4 :

Système d'Annotation des Rôles Sémantiques

Chapitre 4 : Système d'Annotation des Rôles Sémantiques

Chapitre 4 : Système d'Annotation des Rôles Sémantiques

1- Introduction

Nous avons vu dans le chapitre précédent le manque considérable de travaux sur la notion des rôles sémantiques pour la langue l'arabe, les corpus annotés et l'absence totale d'un système qui travaille sur un grand et récent corpus. Ainsi, vu l'importance des rôles sémantiques dans de nombreux domaines du traitement du langage naturel (traduction, questions/réponses, résumé, etc.), nous abordons dans ce chapitre, une méthode d'annotation des rôles sémantiques pour la langue arabe basée sur une approche de raisonnement à partir de cas qui évite de faire des annotations similaires ou faire deux fois la même annotation.

Nous donnons une description générale de la méthode proposée, nous décrivons ensuite le mode de représentation des cas et ses différentes phases. Nous donnons ensuite une vue d'ensemble de cette méthode d'annotation.

2- Approche proposée

Le manque d'applications comme pour la traduction automatique pour la langue l'arabe à la hauteur de celles pour la langue anglaise n'est pas dû au manque de données, mais au manque de données annotées avec les informations nécessaires au traitement du langage naturel [140]. C. Lo et D. Wu montrent l'intérêt des rôles sémantiques de type PropBank dans la traduction automatique pour la paire de langues chinoise/anglaise [156]. Pour la même paire de langues, une autre étude est présentée par D. Wu et P. Fung dans le but de montrer l'apport des rôles sémantiques dans la traduction automatique statistique [157]. Nous avons abordé dans le chapitre deux, certains travaux qui relient les rôles sémantiques à la traduction automatique telle que les travaux de A. Lakhfif et M. T. Laskri [101], M. Bazrafshan et D. Gildea [103], etc.

Notre approche d'annotation des rôles sémantiques est basée sur deux idées. La première est qu'il n'est pas nécessaire par l'utilisateur d'annoter une nouvelle phrase similaire à une ou plusieurs phrases déjà annotées et la deuxième idée est qu'il n'est pas efficace de ne pas mémoriser une phrase déjà annotée ou de faire deux fois l'annotation de la même phrase par le même système. Ainsi, le principe de notre approche est d'utiliser des phrases déjà annotées pour annoter de nouvelles phrases, puis

mémoriser ces nouvelles phrases annotées pour permettre leurs utilisations dans de nouvelles annotations.

Nous nous basons sur le modèle de processus du cycle de raisonnement à partir de cas, les K plus proches voisins et l'apprentissage approfondi pour le développement d'une approche d'annotation de rôles sémantiques pour la langue arabe [158]. Les corpus annotés qui ont une proportion importante sont presque inexistants pour la langue arabe.

2.1- Données utilisées

Nous avons préféré de ne pas utiliser un corpus élaboré seulement pour nos objectifs de recherches, car :

- Il sera forcément de petite taille comparé aux corpus reconnus par la communauté du TALN ;
- Il y a un manque d'annotateurs, linguistes, une collaboration de longue durée entre annotateurs et linguistes, systèmes dédiés à cette tâche d'annotation, etc. ;
- Logiquement, il ne sera pas à la hauteur d'un corpus construit par plusieurs équipes et qui a nécessité des années d'élaboration ;
- Il ne donnera pas de valeur à l'étape d'expérimentation.

Nous trouvons que certaines exigences sur le corpus à utiliser pour le développement et les tests d'une approche sont évidentes pour valoriser n'importe quel travail dans la communauté internationale du TALN :

- Reconnus par la communauté internationale ;
- Subvention de taille, car elle aide dans l'efficacité du travail ;
- Corpus de renommé mondiale ;
- Coordination entre plusieurs universités, laboratoires, sociétés, etc.

Ces exigences et autres attestées par notre corpus mettent en valeur le travail de recherche. C'est un corpus qui utilise Propbank comme formalisme de représentation de sens, ainsi, il est évident d'utiliser le PropBank arabe comme formalisme de représentation de sens pour notre travail.

3- Représentation des cas dans notre approche

La première étape dans le raisonnement à partir de cas est la détermination d'une représentation de cas adéquate au domaine d'étude, les connaissances utilisées, les objectifs du système, etc. Dans l'annotation des rôles sémantiques, les constituants importants sont le prédicat et les arguments de ce prédicat qui doivent être annotés. Nous représentons chaque argument par un cas.

Nous définissons un cas par la paire "prob" et "sol_prob" (4.1).

$$Cas = (prob, sol\_prob) \quad (4.1)$$

Où :

prob : problème du cas

sol_prob : solution du problème "prob"

La partie problème (prob), regroupe un ensemble fini de caractéristiques pour décrire le problème, généralement appelées features (4.2).

$$prob = \{f_1, f_2, f_3, \ldots f_n\} \quad (4.2)$$

Nous nous sommes inspirés de certains travaux sur l'annotation des rôles sémantiques [12], [14] pour la sélection d'un ensemble de features adaptés à notre approche. Le tableau 4.1 représente les features utilisés dans notre approche.

Tableau 4.1. Features sélectionnés dans notre approche [158]

Features	Description
Prédicat / attribut	Le verbe de la phrase
Lemma	Lemma de prédicat
Prédicat	Frameset ID du prédicat selon PropBank
Position dans la phrase	Position de l'argument dans la phrase par rapport à l'attribut
Mot	Les mots qui composent l'argument (le premier mot)
Parties de discours	La partie du discours du mot. Elle contient des informations morphologiques sur le mot arabe (feature précédent).
Analyse	La partie de l'analyse après la première parenthèse ouvrante du format conll 2012

La deuxième partie d'un cas est appelée solution du problème (`sol_prob`). Elle contient une seule information qui représente l'un des rôles sémantiques selon l'arabe PropBank (`ARG0, ARG1, ARG2, ARGMTMP, ARGMADV, etc.`).

Nous donnons la définition générale d'un cas, cependant, dans le RàPC, il n'y a pas un cas sans une étiquette cible ou source. Nous donnons par la suite, la définition des cas sources et cibles dans notre approche.

3.1- Description des cas sources

Un cas source **"S"** représente un argument (constituant) dans une phrase annotée. Comme dans la représentation générale d'un cas, le cas source est composé de la paire (`probS, sol_probS`) (4.3).

Argument phrase déjà annotée = Cas source (S) = (*prob*S, *sol_prob*S)	(4.3)

Où :

`probS` : problème du cas source

`sol_probS` : solution du problème source **"`probS`"**

La partie problème source **"`probS`"** regroupe les features du problème source, ces features représentent un argument d'une phrase annotée. La partie solution du problème source **"`sol_probS`"** représente le rôle sémantique de l'argument (4.4).

probS = { fS_1, fS_2, fS_3,... fS_n } $fS_i = aS_i / vS_i$ sol_probS = rôle sémantique	(4.4)

Où :

fS_i : feature **i** d'un problème source

aS_i : attribut du feature **i**

vS_i : valeur du feature **i**

La base de cas dans le RàPC contient un ensemble fini de cas sources (4.5).

$$\text{Base de cas} = \{S_1, S_2, S_3, \ldots S_n\} \quad (4.5)$$

Où :

s_i : cas source **i**

Elle contient un ensemble d'arguments annotés en rôles sémantiques. Le tableau 4.2 montre un cas source d'un argument dans une phrase déjà annotée.

Tableau 4.2. Cas source de l'argument أَحَدُ اللاجِئِينَ الفِلَسْطِينِيِّينَ

.

Problème source (probS)	
Attribut	رَدَّدَ-#rad~ad#rdd#rad~ad+a-
Lemma	رَدَّدَ-#rad~ad#rdd#rad~ad+a-
Prédicat Frameset ID	01
Position dans la phrase	+06
1er Constituant	أَحَدُ#>aHad#AHd#>aHad+u
Partie de discours	NOUN+CASE_DEF_NOM
Analyse	(NP*
Solution du problème source (sol_probS)	
Rôles sémantiques	ARG0

Élaboré à partir de CoNLL-2012 train data

Data CoNLL-2012 Shared Task\1- conll-2012-train.v4.tar\conll-2012\v4\data\train\data\arabic\annotations\nw\ann\04

Fichier : ann_0401

phrase :

رَدَّدَها أَمْسِ أَحَدُ اللاجِئِينَ الفِلَسْطِينِيِّينَ فِي مُخَيَّمِ عَيْنِ الحُلْوَةِ فِي صَيْدا تَعْبِيراً عَن رَأْيِهِ فِي الوَضْعِ داخِلَ المُخَيَّمِ بَعْدَ الاشْتِباكاتِ الَّتِي حَصَلَت يَوْمَ الثُّلثاءِ بَيْنَ مُسَلَّحِينَ إِسْلامِيِّينَ وَ عَناصِرِ حَرَكَةِ فَتْح وَ الكِفاحِ المُسَلَّحِ فِي المُخَيَّمِ.

Répétée hier par l'un des réfugiés palestiniens du camp d'Ain al-Hilweh à Saïda, exprimant son opinion sur la situation à l'intérieur du camp après des affrontements qui ont eu lieu mardi entre des islamistes armés et des éléments du mouvement Fatah et la lutte armée dans le camp

3.2- Description du cas cible

Le cas cible **"C"** représente un argument dans une nouvelle phrase à annoter. Pareillement à la représentation d'un cas en général ou d'un cas source, le cas cible est aussi composé de la paire `(probC, sol_probC)` (4.6).

Argument d'une phrase à annoter = Cas cible (C) = (probC, sol_probC)	(4.6)

Où :

`probC` : problème du cas cible

`sol_probC` : solution du problème cible **"**`probC`**"**

La partie problème cible **"**`probC`**"** regroupe les features du problème cible, similaires aux features du problème source. Les features caractérisent un argument d'une nouvelle phrase à annoter. La partie solution du problème cible **"**`sol_probC`**"** représente le rôle sémantique assigné par notre approche d'annotation des rôles sémantiques (4.7).

$probC = \{ fC_1, fC_2, fC_3, \ldots fC_n \}$ $fC_i = aC_i / vC_i$ sol_probC = rôle sémantique à identifier	(4.7)

Où :

fC_i : feature *i* d'un problème cible

aC_i : attribut du feature *i* (les attributs d'un problème source et cible sont similaires)

vC_i : valeur du feature *i*

La solution du problème cible est donnée par notre méthode de raisonnement à partir de cas au moyen des cas sources stockés dans la base de cas, mesures de similarités, méthode de classification, etc.

4- Cycle de l'approche

Le cycle de raisonnement à partir de cas est composé de cinq (05) étapes :

- Étape d'élaboration ;
- Étape de remémoration ;
- Étape d'adaptation ;
- Étape de révision ;
- Étape d'apprentissage.

Par priorité, nous avons consacré la section précédente à la description des cas sources et cibles dans notre approche indépendamment des autres étapes du RàPC, car la représentation des cas est une étape préliminaire très importante pour la suite de notre travail. Ainsi, nous considérons cette description comme la première partie de l'élaboration des cas, puisqu'elle aborde la structure des cas dans notre approche.

Nous nous sommes inspiré du travail de B. Fuchs et al [38] pour définir l'élaboration qui est décrite dans la description des cas dans notre approche (4.8) :

Élaboration : argument ∈ phrase non annotée → problème cible ∈ phrase non annotée	(4.8)

Dans ce qui suit, nous abordons les autres étapes du raisonnement à partir de cas de notre approche.

4.1- Étape de remémoration

La principale tâche de cette étape est l'extraction à partir de la base des cas, d'un ou de plusieurs cas sources les plus similaires au cas cible. Pour cela, il est nécessaire de calculer la similarité entre le problème cible et les problèmes sources de la base de cas.

La méthode non-paramétrique des K plus proches voisins `(K near neighbors – KNN)` est basée sur la classification d'un nouvel état dans la classe majoritaire des **K** plus proches voisins du même état [159]. La méthode des K plus proches voisins (K-PPV) est utilisée dans des travaux de RàPC [16], [160], et également dans certains travaux d'annotation des rôles sémantiques [161], [162].

Nous nous sommes inspirés du principe de la méthode des K-PPV, dite **wKNN** utilisée dans [159], [163]. Cependant, nous avons apporté des modifications. Les phases de la méthode **wKNN** dans notre travail sont décrites ci-dessous.

D'abord, nous utilisons l'équation (4.9) pour mesurer la similarité entre le problème cible `(probC)` et un problème source `(probS)`. Cette équation utilise les features du `probC` et des `probS`, pour calculer le pourcentage de la similarité entre le `probC` et le `probS`.

$$d(probC, probS_i) = \sum_{y=1}^{n} w_y \times d_y(fC_y, fS_{iy}) \quad (4.9)$$

Où :

i : cas source i

y : feature y

n : nombre de features

w_y : poids du feature y

d_y : similarité du feature y entre le problème cible et source

fC_y : feature du problème cible

fS_{iy} : feature du problème source

Supposons la liste **L** qui contient un nombre **K** de voisins, `L = {v1, v2, v3,...vk}` et soit deux sous-listes de la liste **L**, appelées **l_1** et **l_2** (4.10).

$$l_1 = \{v_1, v_2, v_3, \ldots v_x\}$$
$$l_2 = \{v_1, v_2, v_3, \ldots v_y\} \quad (4.10)$$

Où :

x : le nombre de voisins de la liste **l_1**
y : le nombre de voisins de la liste **l_2**
} **x + y = k**, **x > y**

l_1 : rassemble les **x** voisins qui représentent la classe `RôleX`

l_2 : rassemble les **y** voisins qui représentent la classe `RôleY`

Si la détermination de la classe est adoptée selon la classe majoritaire, alors la classe `RôleX` sera choisie comme une classe du cas cible, puisque **x** est supérieurs à **y**. Cela signifie que la classe qui est représentée par un grand nombre de voisins est choisie, indépendamment du degré de similarité entre les voisins de cette classe et le cas cible. Ainsi, un groupe majoritaire de voisins qui représentent la même classe et avec des degrés de similarités faibles peut brouiller la détermination d'une classe (solution) adéquate au problème cible.

Pour remédier à ce problème de la non-considération de la similarité des **K** voisins par rapport au `probC`, nous pondérons les similarités. Ainsi, les similarités de l'équation (4.9) sont inversées par l'équation (4.11).

$$d(probC, probS_i) = \frac{1}{d(probC, probS_i)} \tag{4.11}$$

L'équation (4.12) standardise les degrés de similarité des **K** voisins par rapport au degré de similarité du dernier problème source de ces **K** voisins (`probSk)` **.**

$$D_{(i)} = D(probC, probS_i) = \frac{d(probC, probS_i)}{d(probC, probS_k)} \tag{4.12}$$

Ensuite, pour `d = 0`, l'équation (4.13) dite fonction noyau inverse `Q(.)` est dans le maximum **.**

$$Q = \frac{1}{|D|} \tag{4.13}$$

L'équation (4.14) transforme les distances `D(i)` de l'équation (4.12) en poids.

$$K = \frac{1}{D_{(i)}} \tag{4.14}$$

À la fin de cette phase, nous obtenons un nombre **K** de cas source, ces derniers sont les cas les plus similaires au cas cible. Dans la phase suivante d'adaptation, ces cas sources sont utilisés pour proposer une solution au problème cible.

4.2- Étape d'adaptation

Dans cette phase, nous utilisons l'équation (4.15) dont l'objectif est d'avoir une similarité globale pour chaque classe. Cette similarité globale est calculée par la sommation des similarités des voisins de la

même classe. La classe avec une grande similarité globale est proposée comme solution au problème cible `sol_probC`.

$$sol\_probC = max_r \left[\sum_{i=1}^{k} W_i I\left(sol\_probC_i\right) = r \right] \quad (4.15)$$

Où :

$\sum_{i=1}^{k} W_i$: représente la cumulation des poids des voisins parmi les K-PPV qui appartiennent à la classe r.

En d'autres termes, la cumulation des similarités des arguments (voisins) du même rôle sémantique (classe) permet de déterminer la plus grande similarité, le rôle sémantique (classe majoritaire) de celle-ci est proposé comme rôle sémantique `(sol_probC)` du nouvel argument cas cible.

4.3- Étape de révision

Souvent dans les systèmes du raisonnement à partir de cas, la phase de révision est manuelle. Par exemple, dans notre situation un humain (linguiste, annotateur, chercheur en TLN, etc.) valide ou corrige, si selon lui, le rôle sémantique proposé comporte une erreur. L'importance de cette phase est d'améliorer les résultats du système dans les prochaines annotations, puisque le résultat est mémorisé dans la phase qui suit. Également, elle est mieux concrétisée pour un système d'aide à l'annotation ou pour la construction d'un corpus annoté, car elle optimise les efforts des annotateurs.

4.4- Étape d'apprentissage

La dernière phase du cycle de RàPC est la phase d'apprentissage, appelée aussi mémorisation. À la fin de la phase précédente, la deuxième partie du cas cible `sol_probC` est rempli, donc nous avons un cas cible (argument) avec une classification (rôle sémantique). Ce cas cible `(probC, sol_probC)` est mémorisé dans la base des cas et considéré comme un cas source à utiliser dans de prochaines annotations.

5- Le cycle du raisonnement à partir de cas dans notre système

La figure 4.1 décrit le cheminement de notre approche depuis l'élaboration d'un cas cible jusqu'à la mémorisation de ce cas dans la base.

- Premièrement, un `probC` est créé à partir d'un argument dans une phrase donnée. Cette création est considérée comme une phase d'`élaboration`. Ainsi, nous aurons la première

partie (`probC`) d'un cas cible. L'objectif des prochaines étapes est de donner une solution à ce problème cible (`sol_probC`).

– Dans une seconde étape, une étape de `remémoration` est dans le cœur du RàPC pour extraire de la base de cas, un nombre **n** de cas sources les plus similaires au cas cible. Cette similarité est calculée entre le `probC` et la partie `probS` des cas sources. Dans cette similarité un algorithme de K-PPV ou l'apprentissage approfondi est utilisé et il résulte un nombre **K** de cas sources (voisins).

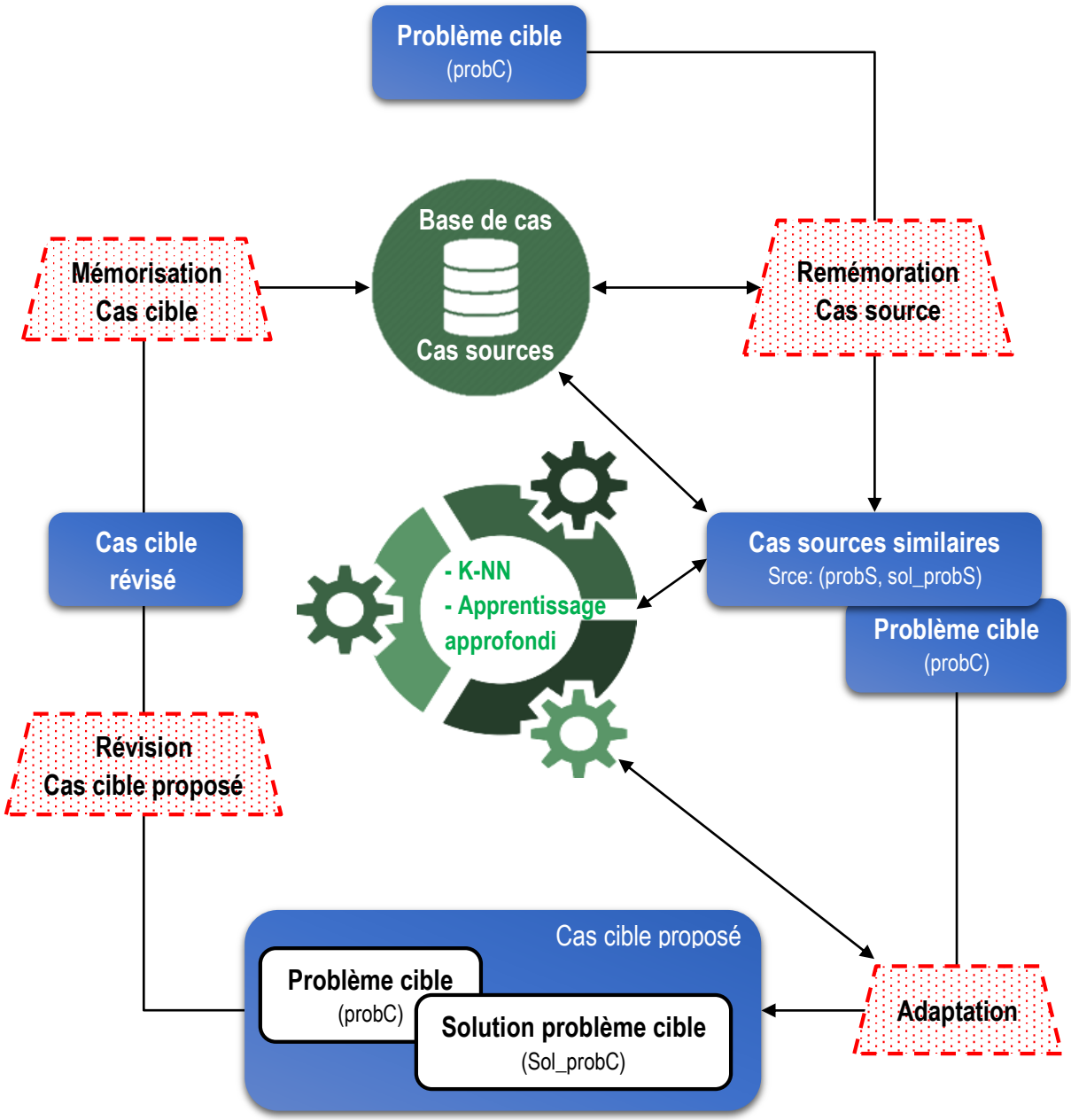


Figure 4.1. Cycle de raisonnement à partir de cas dans notre méthode

- Les cas sources issus de la précédente étape sont utilisés dans une troisième étape d'`adaptation`. Le but est d'avoir une solution au cas cible à partir de la partie `sol_probS` des **K** cas sources remémorés. Ainsi, cette étape propose une solution au problème cible (`sol_probC`).

- Ensuite vient l'étape de `révision` qui consiste à vérifier la solution `sol_probC` proposée par la précédente étape, afin d'utiliser une solution valide dans l'étape suivante.

- La dernière étape de `mémorisation` enregistre les deux parties d'un cas cible `probC` et `sol_probC` dans la base de cas. Ce cas cible mémorisé avec les cas sources est utilisé comme un cas source pour les prochaines annotations.

Ainsi, nous avons une annotation à un argument non annoté à partir des annotations précédentes mémorisées dans la base de cas. Le nombre de cas dans la base augmente au fur et à mesure d'annotation et mémorisation des nouveaux arguments.

6- Conclusion

Nous avons vu dans ce chapitre les différentes phases de notre démarche basée sur une approche de raisonnement à partir de cas et le fonctionnement d' un algorithme des K plus proches voisins.

Les différentes phases et la représentation des cas ont été détaillées dans ce chapitre pour permettre une meilleure concordance entre ce chapitre et le côté technique de la méthode, présenté dans le prochain chapitre.

Chapitre 5 :

Réalisation et Implémentation

Chapitre 5 : Réalisation et Implémentation

Chapitre 5 : Réalisation et Implémentation

1- Introduction

Dans le présent chapitre, nous allons expliquer l'aspect technique d'implémentation de notre approche.

Nous présentons quelques définitions et des statistiques, puis nous présentons la ressource de nos données OntoNotes 5.0, pour montrer son rôle dans la communauté du TALN, ses aspects et ses caractéristiques. Nous terminons par la présentation du système réalisé.

2- Le langage Python

Pour le développement de notre système, nous avons opté pour le langage Python. Créé en 1991 par Guido van Rossum, c'est un langage très complet. Il a la possibilité de définir des scripts, des jeux, des suites bureautiques, des progiciels, etc. Contrairement aux langages compilés, Python est un langage interprété (Figure 5.1), ainsi, il a les avantages de la simplicité et la portabilité sans aucune modification de compilateurs [W11]. Il ne demande pas un nombre important de lignes de code, comparé au langage Java, il est 5 fois plus court (lignes de code), ce qui engendre l'augmentation du rendement des développeurs et abaisse le nombre de bugs [W12]. La dernière version 3.7 date de janvier 2018 [W13].

L'utilisation de ce langage est justifiée à travers les statistiques de la grande institution de la recherche scientifique "IEEE". Depuis 2014, elle donne un classement des langages de programmation suivant plusieurs métriques et à partir de nombreuses sources [W14]. Elle utilise des sources comme : Google Search, Google Trends, Twitter, GitHub, Hacker News, IEEE Xplore Digital Library, etc., et certaines mesures pour chaque source, par exemple, ses deux mesures dans GitHub sont le nombre de nouveaux ajouts et de modification de code pour chaque langage [W15].

Un classement interactif a été publié en juillet 2017 [W16] [W17], les classements des figures 5.2 jusqu'à 5.4 sont extraits à partir de ce classement [W17]. La figure 5.2 montre le classement de 2017 des dix (10) premiers langages, selon les paramètres standards d'IEEE.

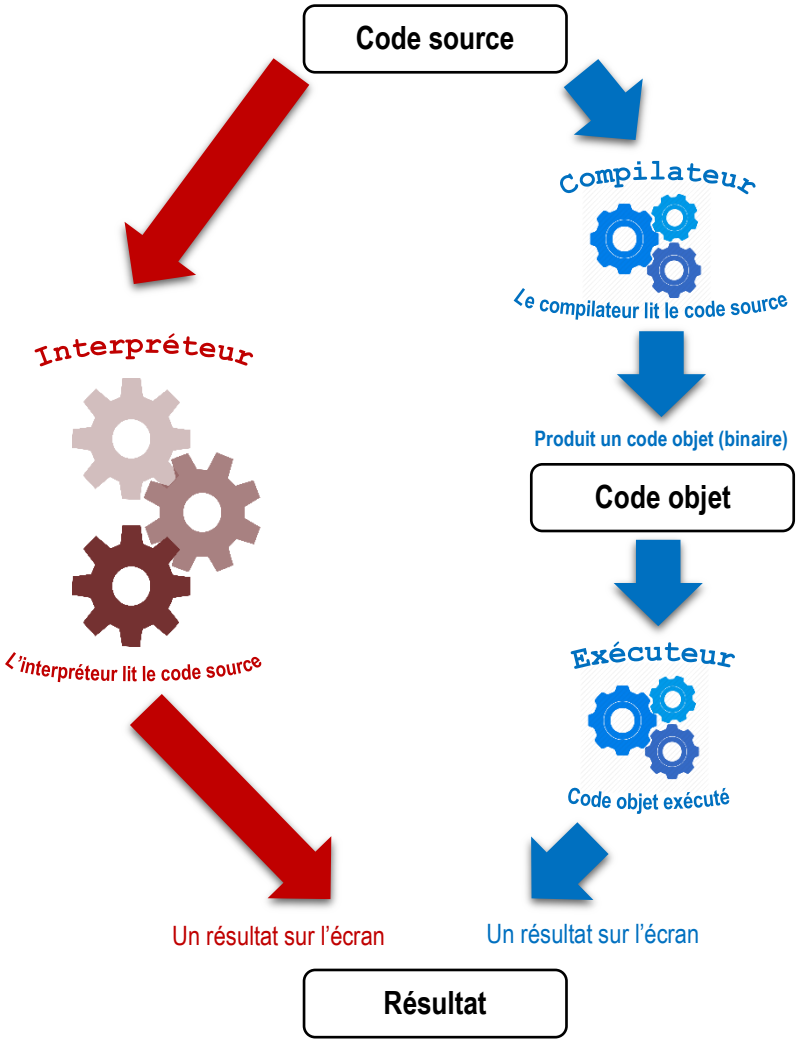


Figure 5.1. Langage compilé et langage interprété

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		99.7
3. Java		99.4
4. C++		97.2
5. C#		88.6
6. R		88.1
7. JavaScript		85.5
8. PHP		81.4
9. Go		76.1
10. Swift		75.3

Figure 5.2. Classement IEEE des langages de programmation

Cependant, comme notre travail est scientifique, nous paramétrons le classement sur la source IEEE Xplore (Figure 5.3), ce qui montre une hausse entre 2014 et 2017 pour le langage Python dans le côté scientifique.

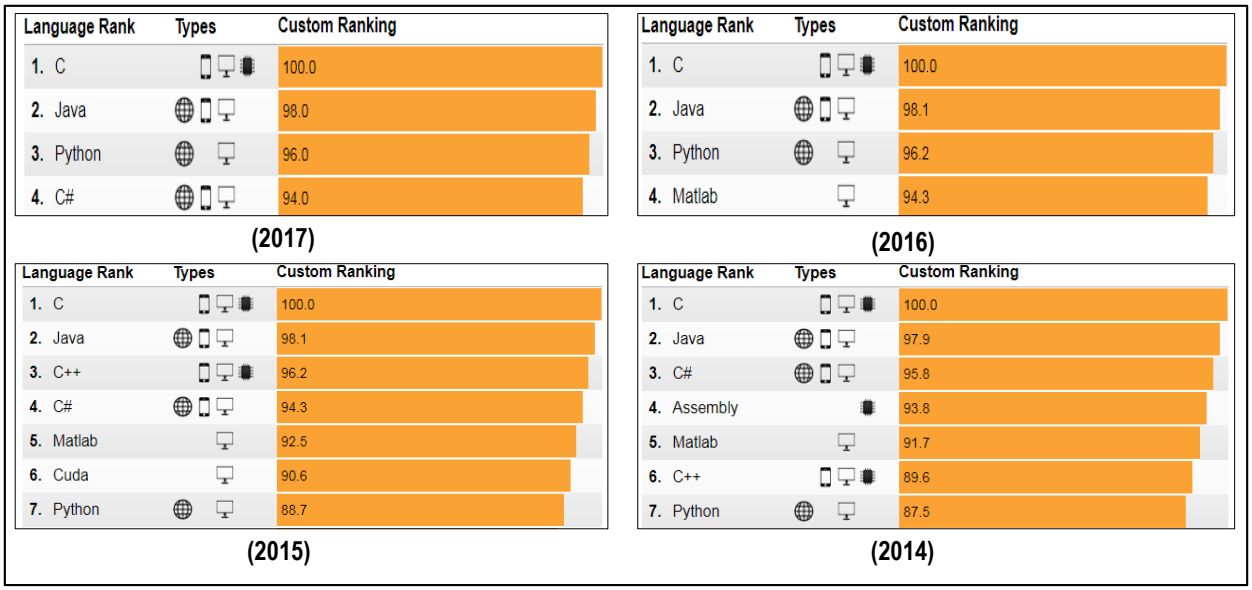


Figure 5.3. Classement utilise IEEE Xplore comme principale source

Dans le même côté scientifique, la figure 5.4 montre que Python est le premier langage utilisé pour les entreprises, les bureaux et les applications scientifiques.

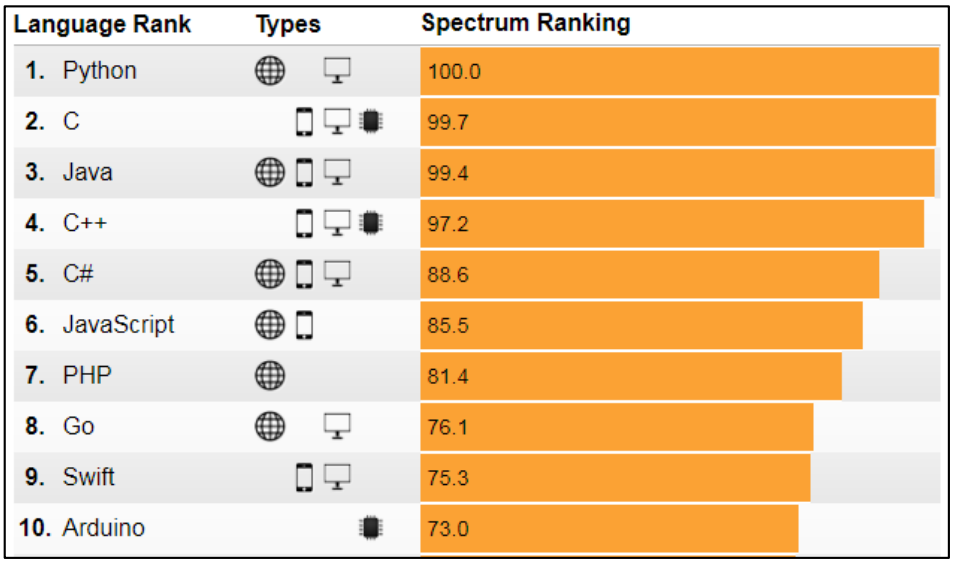


Figure 5.4. Classement IEEE langages utilisés pour les entreprises, bureau et applications scientifiques

De nombreux domaines tels que : la génétique, l'apprentissage automatique, la chimie, le langage naturel, etc., peuvent être exploités grâce aux librairies disponibles dans ce langage [W18]. Dans le cas de notre étude d'apprentissage automatique, il contient des librairies comme : Caffe (Deep learning framework), NuPIC (algorithme d'apprentissage HTM), TensorFlow (réseau de neurones), Scikit-Learn, etc. [W19]

Par exemple, Scikits-Learn regroupe l'apprentissage supervisé, non supervisé, arbres de décision, machines à vecteur de support, K plus proches voisins, réseaux de neurones, etc. [W20]. Côté traitement du langage naturel, il y a une grande boite à outils appelée NLTK (`Natural Language Toolkit`) libre et open source qui contient 50 corpus et ressources lexicales, librairies de traitement de texte et bien d'autres fonctionnalités [W21]. La large utilisation de ce langage en général et dans la direction scientifique en particulier atteste l'intérêt que nous attribuons à son utilisation. Aussi, nous allons voir que la première phase de construction du corpus utilisé doit passer par un script Python (disponible avec les données initiales).

Pour notre système, basé sur les K plus proches voisins, nous avons programmé notre propre algorithme sans l'utilisation de librairies d'apprentissage automatique. Le but est de mieux personnaliser l'algorithme à notre tâche, bon choix des paramètres, faire plusieurs expérimentations et éviter de travailler dans une boite noire comme dans le cas des librairies d'apprentissage automatique.

Pour le système basé sur un apprentissage approfondi (Deep Learning), nous utilisons la librairie Scikits-Learn afin d'exploiter les avantages de ce type d'apprentissage très puissant par rapport aux approches classiques de l'IA, via cette célèbre librairie.

3- OntoNotes 5.0

OntoNotes 5.0[26] est la principale source de notre base de cas. Selon S. Pradhan et L. Ramshaw, les types de représentation syntaxiques et sémantiques variés et riches rendent cette ressource première dans ce genre de travaux sur l'annotation [164]. Il constitue l'une des contributions du Linguistic Data Consortium (LDC)[27] de l'université de Pennsylvanie. Ce corpus est le résultat d'importants collaborateurs : BBN Technologies[28], les universités du Colorado, Brandeis [120], Pennsylvanie et Southern Californias Information Sciences Institute. Il comporte trois langues : l'Anglais, le Chinois et l'Arabe, de différents genres : les conversations téléphoniques, les weblogs, les journaux, etc. [W22].

C'est une ressource d'une taille de 1.5 million de mots anglais, 800k chinois et 300k pour l'arabe. Le tableau 5.1 indique le nombre et les types de données dans OntoNotes 5.0. La partie arabe de cette ressource comporte cinq (05) types d'annotations : **TreeBank** (syntaxique), sens du mot, **PropBank** (proposition), coréférence et entité nommée (`named entity`) [120].

[26] https://catalog.ldc.upenn.edu/ldc2013t19
[27] https://www.ldc.upenn.edu/
[28] Importante société dans le domaine d'informatique, parmi ses contributions le réseau ARPANET (https://en.wikipedia.org/wiki/BBN_Technologies)

Tableau 5.1. Nombre et types de ressources OntoNotes 5.0 [120]

	Anglais	Chinois	Arabe
Journal	625 k	250 k	300 k
Broadcast	200 k	250 k	–
Broadcast conversation	200 k	150 k	–
Texte web	300 k	150 k	–
Conversation téléphonique	120 k	100 k	–
Pivot Corpus	300 k	–	–

Le fichier des annotations de la langue arabe dans OntoNotes 5.0 contient six (06) dossiers (Figure 5.5) avec sept (07) extensions (coref, lemma, name, onf, parse, prop, sens et source). Chacune de ces extensions est consacrée à un niveau d'annotation (certaines de ces extensions ne sont pas disponibles dans tous les dossiers).

Par exemple, pour l'extension "`.onf`" (`OntoNotes Normal Form`), elle facilite la compréhension des annotations des fichiers OntoNotes 5.0 et l'extension "`.source`" contient les phrases originales [120].

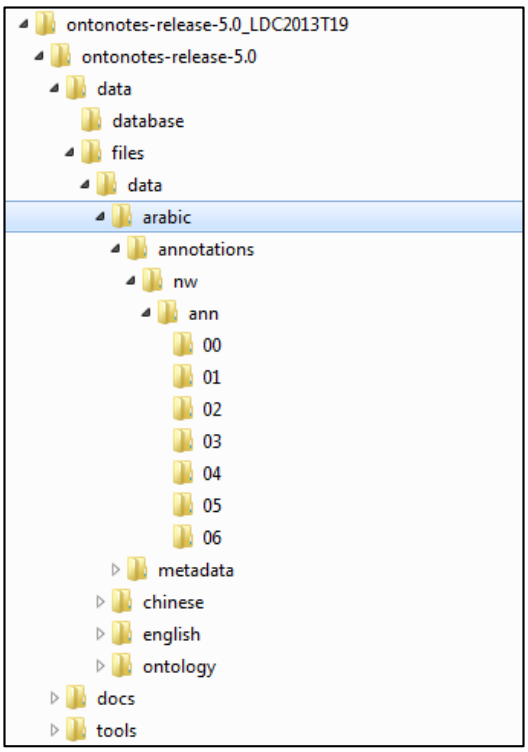


Figure 5.5. Dossiers dans OntoNotes 5.0

Le texte arabe ci-dessous est extrait du corpus OntoNotes 5.0, dossier arabe '03', le fichier 'ann_0300.source' et identifié par le texte numéro '04'. Ce texte est considéré comme une seule phrase (entrée) TreeBank dans le fichier 'ann_0300.onf'.

اننا ضد الاضرابات والتظاهرات، وضد أي اهتزاز يمكن ان تكون انعكاساته سلبية على البلاد والعباد، لان ما يحيط بنا الآن من اوضاع اقليمية وخارجية يحتم علينا وعيا في التصرف، وادراكا في ابعاد كل ما نقوم به من خطوات، لذا علينا ان نكون منتبهين جدا، ويقظين، فالمناخات غير الصافية التي تهب علينا في استمرار، والضغوط التي تمارس من كل حدب وصوب وفي كل الاتجاهات السياسية والاقتصادية والاجتماعية يجب ان تكون حافزا يدفعنا الى المزيد من التعاون والتوافق لتفويت الفرص امام كل من يضمر شرا لهذا الوطن الذي دفع شعبه الاثمان باهظة، ولا يزال يكابد آثار الارهاب الصهيوني الذي لم يطاول الفلسطينيين وحدهم، بل لبنان وسوريا وكل شعوب المنطقة، وبالتالي فهو يهدد استقرار العالم بأسره .

Nous sommes contre les grèves, manifestations, et contre toute instabilité qui peut avoir un impact négatif sur le pays et le peuple. Parce que ce qui nous entoure maintenant des situations régionales et externes, nous obliger à acter consciemment et une conscience dans les dimensions de tous les pas que nous faisons. Donc nous devons être très attentif et vigilant, les climats non nets qui soufflent sur nous continuellement et les pressions exercées de tous les côtés et dans toutes les directions politiques, économiques et sociales doivent être une incitation à nous poussent à plus de coopération et consensus pour rater les occasions à ceux qui dissimule le mal pour ce pays, que son peuple a payé les prix chers et qu'il subit encore les effets de la terreur sioniste non seulement sur les Palestiniens, mais aussi le Liban, la Syrie et tous les peuples de la région, ainsi il menace la stabilité du monde entier.

Comme nous sommes intéressés par la structure sémantique, l'annotation des rôles sémantiques du texte arabe ci-dessous (extrait du texte précédent) est présentée dans la figure 5.6.

...ما يحيط بنا الآن من اوضاع اقليمية وخارجية يحتم علينا وعيا في التصرف،...

…ce qui nous entoure maintenant des situations régionales et externes, nous obliger à acter consciemment …

Figure 5.6. Annotation d'un texte arabe

Extrait de OntoNotes 5.0

Répertoire : …\ontonotes-release-5.0\data\files\data\arabic\annotations\nw\ann\03

Fichier : ann_0300.onf

3.1- Donnée CoNLL

Dans la conférence CoNLL-2012, les données utilisées pour l'anglais, le chinois et l'arabe sont extraites à partir d'OntoNotes 5.0, ils sont utilisés dans une tâche de coréférence. Un algorithme particulier est utilisé par les organisateurs pour la création des parties de développement, d'entrainement et de test [165].

Les données CoNLL sont organisées dans des dossiers pour chaque langue, ces dossiers contiennent des fichiers qui regroupent les niveaux d'annotation. Ces données ont une autre extension que les sept (07) extensions abordées précédemment, et elle a le même nom de la conférence "`.conll`".

Cependant, les fichiers de cette extension ne sont pas disponibles avec les données, un traitement est nécessaire pour leur création (Section 4.1).

Cette extension est en forme spécialisée, similaire à un tableau de **n** lignes et **n** colonnes. Chaque ligne représente un constituant et chaque colonne représente une information : fichiers, numéro du mot, racine, partie de discours (`part of speech`), etc. [165]. Le tableau 5.2 donne la définition de chaque colonne.

Tableau 5.2. Les colonnes du format colonne [165]

Numéro de colonne	Représentation[29]	Description
1	Identification dossier	Les dossiers du fichier
2	Numéro fichier	Numéro du fichier
3	Numéro du constituant	Numéro de ce constituant dans la phrase
4	Constituant	Le constituant lui-même
5	Partie de discours	Partie de discours du constituant
6	Parse bit	Partie de l'analyse après la première parenthèse
7	Lemma	C'est comme une racine du constituant.
8	Frameset du prédicat	Identification du frameset du Lemma selon PropBank, s'il a un frameset.
9	Sens constituant	Le sens de ce constituant
10	Orateur/auteur	Le nom de l'orateur ou l'auteur, il n'est pas disponible pour la langue arabe
11	Entité nommée	Représentation des entités nommées
12 : N	Argument du prédicat	Le nombre de colonnes n'est pas fixe, car chaque colonne représente un seul prédicat (Lemma) et ses arguments.
N+1	Coréférence	Information sur les coréférence

[29] Pour des raisons de traduction et d'adaptation, nous n'avons pas respecté les noms des colonnes et les descriptions exactes mentionnées dans le document original de ce tableau.

La figure 5.7 illustre un exemple depuis un fichier OntoNotes 5.0 (`.onf`) et le même exemple (Figure 5.8) dans un fichier au format conll (`.conll`).

```
Plain sentence:
---------------
    لكن ه لم يشر الى اقفال الطرقات اليوم و غدا ك ما تعهد سابقا .

Treebanked sentence:
--------------------
    ل`كِنْ- -ه لَم يُشِرْ *-1 إلَى إقْفالِ الطُّرُقاتِ اليَوْمَ وَ- -غَداً كَ- -ما تَعَهَّدَ * سابِقاً .

Tree:
-----
    (TOP (S (VP (PSEUDO_VERB ل`كِنْ-)
                (NP-SBJ-1 (PRON_3MS -ه))
                (S (VP (PRT (NEG_PART لَم))
                       (IV3MS+IV+IVSUFF_MOOD:J يُشِرْ)
                       (NP-SBJ (-NONE- *-1))
                       (PP-CLR (PREP إلَى)
                               (NP (NOUN+CASE_DEF_GEN إقْفالِ)
                                   (NP (DET+NOUN+NSUFF_FEM_PL+CASE_DEF_GEN الطُّرُقاتِ))))
                       (NP-TMP (DET+NOUN+CASE_DEF_ACC اليَوْمَ)
                               (CONJ وَ-)
                               (NOUN+CASE_INDEF_ACC -غَداً))
                       (PP-MNR (PREP كَ-)
                               (SBAR (SUB_CONJ -ما)
                                     (S (VP (PV+PVSUFF_SUBJ:3MS تَعَهَّدَ)
                                            (NP-SBJ (-NONE- *))
                                            (NP-TMP (NOUN+CASE_INDEF_ACC سابِقاً)))))))))
            (PUNC .)))

Leaves:
-------
    0   ل`كِنْ-
    1   -ه
        coref: IDENT        70817 1-1    ه
    2   لَم
    3   يُشِرْ
        prop:  Oa$Ar.01
         v          * -> 3:0,  يُشِرْ
         ARG0       * -> 1:1,  -ه
                    * -> 4:0,  *-1 -> 1:1, -ه
                    * -> 14:0, *
         ARG1       * -> 5:1,  إلَى إقْفالِ الطُّرُقاتِ
         ARGM-TMP   * -> 8:1,  اليَوْمَ وَ- -غَداً
         ARGM-MNR   * -> 11:1, سابِقاً * تَعَهَّدَ -ما -كَ
    4   *-1
    5   إلَى
    6   إقْفالِ
    7   الطُّرُقاتِ
    8   اليَوْمَ
        sense: yawom-n.4
    9   وَ-
    10  -غَداً
    11  كَ-
    12  -ما
    13  تَعَهَّدَ
        prop:  taEah~ad.01
         v          * -> 13:0, تَعَهَّدَ
         ARG0       * -> 1:1,  -ه
                    * -> 14:0, *
         ARGM-TMP   * -> 15:1, سابِقاً
    14  *
        coref: IDENT        70817 14-14  *
    15  سابِقاً
    16  .
```

Figure 5.7. Exemple d'un fichier d'OntoNotes 5.0

Répertoire : ...\ontonotes-release-5.0\data\files\data\arabic\annotations\nw\ann\02

Fichier : ann_0290.onf

1	2		3	4	5	6	7	8	9	10	11	12 : N	N+1	
nw/ann/02/ann_0290	0		0	لِ-#l`kin~a#lkn#l`kin~a-كِنَّ	PSEUDO_VERB	(TOP(S(VP*	l`kin~a	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		1	-ُ#clitics#h#-hu	PRON_3MS	(NP*)	clitics	-	-	-	*	(ARG0*)	(ARG0*)	(6)
nw/ann/02/ann_0290	0		2	أَم#lam#lm#lam	NEG_PART	(S(VP(PRT*)	lam	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		3	يُشِرْ#>a$Ar#y$r#yu+$ir+o	IV3MS+IV+IVSUFF_MOOD:J	*	>a$Ar	01	-	-	*	(V*)	*	-
nw/ann/02/ann_0290	0		4	إِلَى#<ilaY#AlY#<ilaY	PREP	(PP*	<ilaY	-	-	-	*	(ARG1*	*	-
nw/ann/02/ann_0290	0		5	إِقْفالِ#<iqofAl#AqfAl#<iqofAl+i	NOUN+CASE_DEF_GEN	(NP*	<iqofAl	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		6	الطُّرُقاتِ#Tariyq#AlTrqAt#Al+Turuq+At+i	DET+NOUN+NSUFF_FEM_PL+CASE_DEF_GEN	(NP*)))	Tariyq	-	-	-	*	*)	*	-
nw/ann/02/ann_0290	0		7	الْيَوْمَ#yawom#Alywm#Al+yawom+a	DET+NOUN+CASE_DEF_ACC	(NP*	yawom	-	4	-	*	(ARGM-TMP*	*	-
nw/ann/02/ann_0290	0		8	وَ-#clitics#w#wa-	CONJ	*	clitics	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		9	غَداً-#gad#gdA#-gad+AF	NOUN+CASE_INDEF_ACC	*)	gad	-	-	-	*	*)	*	-
nw/ann/02/ann_0290	0		10	كَ-#clitics#k#ka-	PREP	(PP*	clitics	-	-	-	*	(ARGM-MNR*	*	-
nw/ann/02/ann_0290	0		11	-ما#kamA#mA#-mA	SUB_CONJ	(SBAR*	kamA	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		12	تَعَهَّدَ#taEah~ad#tEhd#taEah~ad+a	PV+PVSUFF_SUBJ:3MS	(S(VP*	taEah~ad	01	-	-	*	*	(V*)	-
nw/ann/02/ann_0290	0		13	سابِقاً#sAbiq#sAbqA#sAbiq+AF	NOUN+CASE_INDEF_ACC	(NP*))))))))	sAbiq	-	-	-	*	*)	(ARGM-TMP*)	-
nw/ann/02/ann_0290	0		14	.#DEFAULT#.#.	PUNC	*))	DEFAULT	-	-	-	*	*	*	-

Figure 5.8. Exemple d'un fichier dans le format conll

Répertoire : ...\conll-2012\v4\data\development\data\arabic\annotations\nw\ann\02

Fichier : ann_0290.v4_gold_conll

Package : conll-2012-development.v4.tar

4- Données utilisées

Nous utilisons une importante ressource qui possède de nombreux avantages :

- Crédibilité dans la communauté internationale du TALN ;
- Vérification et mise à jour ultérieures des niveaux d'annotation comme TreeBank ;
- La plus grande ressource annotée en rôles sémantiques pour la langue arabe ;
- Disponible gratuitement ;
- Grandes subvention et coopération : ministère de la défense américain, BBN Technologies, université du Colorado, université de Pennsylvanie, etc. qui rendent cette ressource très importante et difficile à reproduire.
- La non-existence d'une ressource analogue à celle-ci pour la langue arabe.

Cette ressource répond aux exigences évoquées précédemment qui mettent en valeur notre contribution. Ces avantages et d'autres encouragent les recherches dans la langue arabe à utiliser cette ressource. Pour cela, nous détaillons la préparation des données pour une éventuelle utilisation dans les recherches en traitement automatique du langage naturel (TALN) en général, et pour l'annotation des rôles sémantiques en particulier.

4.1- Préparation des données CoNLL-2012

Cette première étape de préparation des données est nécessaire pour n'importe quelle recherche qui utilisera les données CoNLL-2012.

Les données sont disponibles gratuitement sur le site de CoNLL-2012 (`http://conll.cemantix.org/2012/data.html`). Nous disposons de huit (08) fichiers et chaque fichier compressé contient un ensemble de dossiers et de fichiers.

1. conll-2012-trial-data.tar
2. conll-2012-train.v4.tar
3. conll-2012-development.v4.tar
4. conll-2012-test-official.v9.tar
5. conll-2012-test-supplementary.v9.tar
6. conll-2012-test-key.tar
7. conll-2012-scripts.v3.tar
8. conll-2012-submissions.tar

Cependant, certaines informations d'OntoNotes 5.0 manquent dans les données CoNLL-2012 [165].

Nous reprenons l'exemple de la figure 5.8. Dans la nouvelle figure 5.9, originale depuis le fichier téléchargé de CoNLL-2012, nous remarquons que les colonnes **Constituant** (colonne 4) et **Lemma** (colonne 7) ne sont pas disponibles, le format de ces fichiers est (`.skel`).

Pour compléter ces données, il est nécessaire d'utiliser un script Python disponible avec ces données et la ressource OntoNotes 5.0.

4.2- Fusionnement entre CoNLL-2012 et OntoNotes 5.0

Vu les difficultés pour former les fichiers d'extension (`.conll`) et le manque d'instructions explicites, nous avons vu la nécessité de noter, en ordre chronologique, les étapes importantes que nous avons suivies dans cette construction des données et qui sont nécessaires pour toutes constructions de ces données par d'autres utilisateurs (Annexe C).

À la fin de l'exécution du script, nous aurons des fichiers au format (`.conll`). La figure 5.9 montre une phrase dans un fichier d'extension (`.skel`) avant l'exécution du script, il manque les informations de deux colonnes, puis la figure (Figure 5.8) montre la même phrase dans un fichier d'extension (`.conll`) où les deux colonnes sont complétées.

Pour d'autres nécessités de traitement du langage naturel, d'autres scripts sont disponibles avec les mêmes données [W23].

5- Élaboration des données

Nous avons décomposé le processus d'élaboration en trois (03) étapes, selon le type des données et la méthode de préparation.

5.1- Première étape : Pré-élaboration

Une étape de préparation et de structuration des données est nécessaire afin d'utiliser des données pertinentes par rapport à notre objectif de recherche.

Après l'exécution du script Python, nous obtenons pour la langue arabe un ensemble de dossiers et chaque dossier contient plusieurs extensions. Comme nous l'avons cité précédemment, l'extension a un but ou un niveau d'annotation. Cependant, il faut préciser que l'élimination d'une extension ne réduit pas le nombre de phrases dans ces données, c'est juste l'élimination d'un fichier qui représente une information sur la phrase non nécessaire pour notre travail.

1	2		3	4	5	6	7	8	9	10	11	12 : N		N+1
nw/ann/02/ann_0290	0		0	[WORD]	PSEUDO_VERB	(TOP(S(VP*	[LEMMA]	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		1	[WORD]	PRON_3MS	(NP*)	[LEMMA]	-	-	-	*	(ARG0*)	(ARG0*)	(6)
nw/ann/02/ann_0290	0		2	[WORD]	NEG_PART	(S(VP(PRT*)	[LEMMA]	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		3	[WORD]	IV3MS+IV+IVSUFF_MOOD:J	*	[LEMMA]	01	-	-	*	(V*)	*	-
nw/ann/02/ann_0290	0		4	[WORD]	PREP	(PP*	[LEMMA]	-	-	-	*	(ARG1*	*	-
nw/ann/02/ann_0290	0		5	[WORD]	NOUN+CASE_DEF_GEN	(NP*	[LEMMA]	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		6	[WORD]	DET+NOUN+NSUFF_FEM_PL+CASE_DEF_GEN	(NP*)))	[LEMMA]	-	-	-	*	*)	*	-
nw/ann/02/ann_0290	0		7	[WORD]	DET+NOUN+CASE_DEF_ACC	(NP*	[LEMMA]	-	4	-	*	(ARGM-TMP*	*	-
nw/ann/02/ann_0290	0		8	[WORD]	CONJ	*	[LEMMA]	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		9	[WORD]	NOUN+CASE_INDEF_ACC	*)	[LEMMA]	-	-	-	*	*)	*	-
nw/ann/02/ann_0290	0		10	[WORD]	PREP	(PP*	[LEMMA]	-	-	-	*	(ARGM-MNR*	*	-
nw/ann/02/ann_0290	0		11	[WORD]	SUB_CONJ	(SBAR*	[LEMMA]	-	-	-	*	*	*	-
nw/ann/02/ann_0290	0		12	[WORD]	PV+PVSUFF_SUBJ:3MS	(S(VP*	[LEMMA]	01	-	-	*	*	(V*)	-
nw/ann/02/ann_0290	0		13	[WORD]	NOUN+CASE_INDEF_ACC	(NP*))))))))	[LEMMA]	-	-	-	*	*)	(ARGM-TMP*)	-
nw/ann/02/ann_0290	0		14	[WORD]	PUNC	*))	[LEMMA]	-	-	-	*	*	*	-

Figure 5.9. Exemple d'un fichier dans le format skel

Répertoire : ...\conll-2012\v4\data\development\data\arabic\annotations\nw\ann\02

Fichier : ann_0290.v4_gold_skel

Package : conll-2012-development.v4.tar

Nous ne gardons que les fichiers avec l'extension (`.conll`), car ils contiennent tous les niveaux d'annotation (déjà abordé ultérieurement). Les autres extensions supplémentaires sont éliminées. Nous éliminons ensuite les fichiers conll dans " `test-supplementary.v9.tar` " et " `test-official.v9.tar` ", car ils ne contiennent pas des données PropBank (annotation des rôles sémantiques). Le tableau 5.3 donne des détails sur le dossier arabe dans chaque package.

Tableau 5.3. Informations des dossiers arabes

Package	Nombre de fichiers .conll	Nombre de fichiers avec annotation PropBank
trial-data.tar	2	2
train.v4.tar	359	359
development.v4.tar	44	44
test-key.tar	44	44
test-supplementary.v9.tar	-	0
test-official.v9.tar	44	0 (contient les mêmes fichiers de "test-key.tar" la différence dans les niveaux d'annotation)

Les fichiers des packages `trial-data.tar`, `train.v4.tar` et `development.v4.tar` sont utilisés pour l'élaboration de la base de cas (cas sources) et les fichiers du package `test-key.tar` sont utilisés pour l'élaboration de la base de test (cas cibles).

5.2- Deuxième étape : Extraction des features et élaboration de la base utilisée pour le système des K plus proches voisins (K-PPV)

Dans cette étape, nous avons développé des scripts Python pour la construction automatique de la base de cas. Les bases de cas sources et cibles sont utilisées dans notre système K-PPV. Les scripts Python dédiés à la création de ces cas sont disponibles dans GitHub et dans l'annexe B (`https://github.com/Hamzameguehout/Semantic-Annotation`).

- **create_case.py :** premièrement, il faut noter que ce script est responsable de l'élaboration des cas sources et cibles. C'est le script principal et il possède quatre (04) fonctions :
  - ✓ La création des dossiers ;
  - ✓ La création des fichiers de la base de cas ;
  - ✓ Appeler le script de création des cas sources ;
  - ✓ Appeler le script de création des cas cibles sans et avec les rôles sémantiques.

Nous remarquons une certaine similarité entre l'organisation des dossiers et les noms de fichiers entre OntoNotes 5.0 et CoNLL-2012. Pour des raisons de conformité et de bonne structuration, nous choisissons de respecter l'organisation des données de CoNLL-2012. Pour cela, le code de ce script crée des dossiers et des noms de fichiers similaires à la ressource originale issue de la pré-élaboration.

5.2.1- Base des cas sources (base de cas)

L'objectif est de construire pour chaque argument (un ou plusieurs constituants) un cas source. Cette élaboration de cas respecte la structuration globale du cas source présentée dans le chapitre précédent.

- **srce_case.py :** la fonction de ce script est d'élaborer un cas source à partir d'un argument PropBank dans son format original `conll`. Il lit les données des phrases depuis la ressource originale et génère un cas source avec les informations (feature) nécessaires pour les autres étapes du cycle RàPC.

Un cas d'utilisation est nécessaire pour montrer l'input et l'output de cette étape d'élaboration de cas sources. La figure 5.10 montre une phrase arabe dans son format d'origine `conll`, puis la figure 5.11 montre quatre arguments de cette partie dans notre fichier (output).

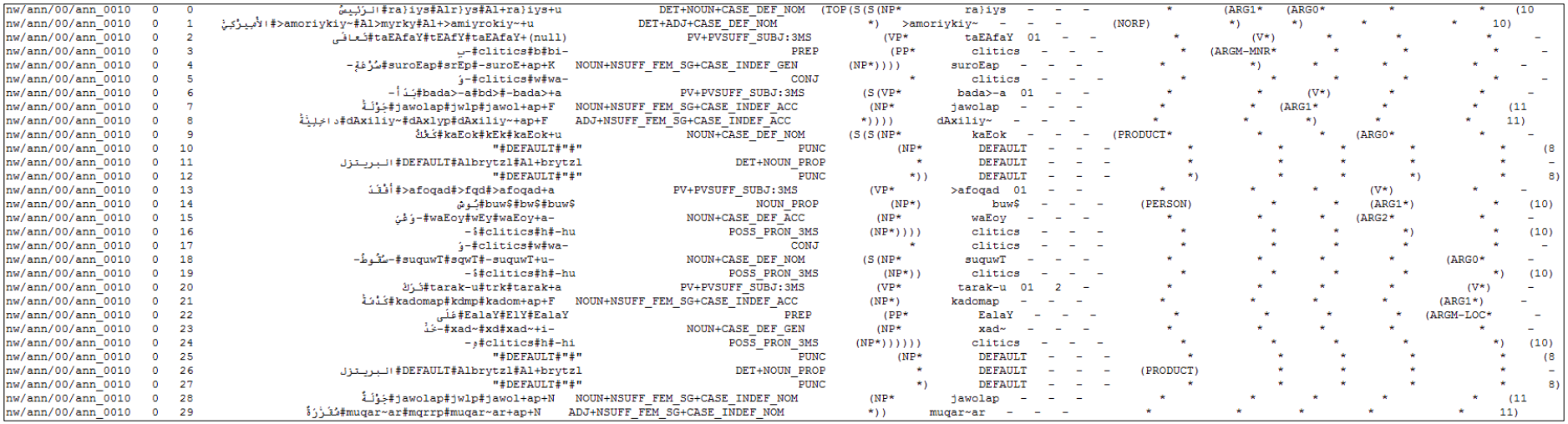
Dans la figure 5.11, il y a quatre (04) cas, chacun contient un ensemble d'attributs et des blancs (caractères) séparent les features. Le dernier représente le rôle sémantique de l'argument (solution cas source). Dans l'exemple, les rôles sémantiques sont ARG1, ARGM-MNR, ARG0 et ARG1.

5.2.2- Base des cas cibles (base de test)

Le but de cette étape est d'avoir une base de test pour notre approche. La base de cas cibles ou base de test contient un ensemble de cas cibles (problème cible). Le script `create_case.py` appelle le script `tgt_case.py`.

- ***tgt_case.py :*** les cas sources et les cas cibles ont la même structure, ainsi, le code du script `tgt_case.py` est presque similaire au code de `srce_case.py`.

Le script `srce_case.py` élabore les deux parties d'un cas source (problème source, solution du problème source), mais le script `tgt_case.py` élabore seulement la partie problème cible du cas cible, puis la partie solution du problème cible sera complétée par un raisonnement à partir de cas. Ainsi, l'output est similaire à l'output de la figure 5.11, sans l'attribut rôle sémantique. Un autre scripte `tgt_case_semantic_roles.py` élabore les cas cibles, mais avec l'attribution des rôles sémantiques afin de les utiliser pour tester le résultat.



```
nw/ann/00/ann_0010   0    0                                  الـرَّئِيسُ#ra}iys#Alr}ys#Al+ra}iys+u                DET+NOUN+CASE_DEF_NOM  (TOP(S(S(NP*          ra}iys   -   -   -          *       (ARG1*   (ARG0*          *          *    (10
nw/ann/00/ann_0010   0    1    الأَمِيرْكِيِّ#>amoriykiy~#Al>myrky#Al+>amiyrokiy~+u                DET+ADJ+CASE_DEF_NOM             *)    >amoriykiy~   -   -   -         (NORP)          *)       *)         *          *     10)
nw/ann/00/ann_0010   0    2                                  تَعافَى#taEAfaY#tEAfY#taEAfaY+(null)                    PV+PVSUFF_SUBJ:3MS          (VP*          taEAfaY  01   -   -          *          (V*)       *          *          *      -
nw/ann/00/ann_0010   0    3                                  بـ#clitics#b#bi-                                        PREP          (PP*          clitics   -   -   -          *   (ARGM-MNR*         *          *          *      -
nw/ann/00/ann_0010   0    4                              سُرْعَةِ#suroEap#srEp#-suroE+ap+K   NOUN+NSUFF_FEM_SG+CASE_INDEF_GEN       (NP*))))       suroEap   -   -   -          *          *)         *          *          *      -
nw/ann/00/ann_0010   0    5                                  وَ#clitics#w#wa-                                        CONJ                *          clitics   -   -   -          *          *          *          *          *      -
nw/ann/00/ann_0010   0    6                                  بَدَأَ#bada>-a#bd>#-bada>+a                          PV+PVSUFF_SUBJ:3MS        (S(VP*        bada>-a  01   -   -          *          *       (V*)       *          *      -
nw/ann/00/ann_0010   0    7                              جَوْلَةُ#jawolap#jwlp#jawol+ap+F   NOUN+NSUFF_FEM_SG+CASE_INDEF_ACC          (NP*        jawolap   -   -   -          *          *    (ARG1*        *          *     (11
nw/ann/00/ann_0010   0    8                          داخِلِيَّةُ#dAxiliy~#dAxlyp#dAxiliy~+ap+F    ADJ+NSUFF_FEM_SG+CASE_INDEF_ACC        *))))      dAxiliy~   -   -   -          *          *          *)         *          *     11)
nw/ann/00/ann_0010   0    9                                  كَعْكُ#kaEok#kEk#kaEok+u                             NOUN+CASE_DEF_NOM       (S(S(NP*          kaEok   -   -   -   (PRODUCT*          *          *    (ARG0*         *          *      -
nw/ann/00/ann_0010   0   10                                  "#DEFAULT#"#"                                           PUNC          (NP*        DEFAULT   -   -   -          *          *          *          *          *      (8
nw/ann/00/ann_0010   0   11                  البريـتزل#DEFAULT#Albrytzl#Al+brytzl                                DET+NOUN_PROP             *        DEFAULT   -   -   -          *          *          *          *          *      -
nw/ann/00/ann_0010   0   12                                  "#DEFAULT#"#"                                           PUNC             *))       DEFAULT   -   -   -          *)         *          *          *)         *      8)
nw/ann/00/ann_0010   0   13                              أَفْقَدَ#>afoqad#>fqd#>afoqad+a                      PV+PVSUFF_SUBJ:3MS        (VP*        >afoqad  01   -   -          *          *          *       (V*)       *          *      -
nw/ann/00/ann_0010   0   14                                  بُوشْ#buw$#bw$#buw$                                     NOUN_PROP             (NP*)          buw$   -   -   -     (PERSON)          *          *    (ARG1*)       *          *     (10)
nw/ann/00/ann_0010   0   15                              وَعْيِ#waEoy#wEy#waEoy+a-                            NOUN+CASE_DEF_ACC          (NP*          waEoy   -   -   -          *          *          *    (ARG2*         *          *      -
nw/ann/00/ann_0010   0   16                                  -ـهُ#clitics#h#-hu                                   POSS_PRON_3MS        (NP*))))        clitics   -   -   -          *          *          *          *)         *     (10)
nw/ann/00/ann_0010   0   17                                  وَ#clitics#w#wa-                                        CONJ                *          clitics   -   -   -          *          *          *          *          *      -
nw/ann/00/ann_0010   0   18                              سُقُوطِ#suquwT#sqwT#-suquwT+u-                       NOUN+CASE_DEF_NOM        (S(NP*          suquwT   -   -   -          *          *          *          *       (ARG0*        -
nw/ann/00/ann_0010   0   19                                  -ـهُ#clitics#h#-hu                                   POSS_PRON_3MS          (NP*))        clitics   -   -   -          *          *          *          *          *)    (10)
nw/ann/00/ann_0010   0   20                                  تَرَكَ#tarak-u#trk#tarak+a                           PV+PVSUFF_SUBJ:3MS          (VP*        tarak-u  01   2   -          *          *          *          *        (V*)        -
nw/ann/00/ann_0010   0   21                          كَدْمَةً#kadomap#kdmp#kadom+ap+F   NOUN+NSUFF_FEM_SG+CASE_INDEF_ACC          (NP*)       kadomap   -   -   -          *          *          *          *     (ARG1*)       -
nw/ann/00/ann_0010   0   22                                  عَلَى#EalaY#ElY#EalaY                                   PREP          (PP*          EalaY   -   -   -          *          *          *          *   (ARGM-LOC*        -
nw/ann/00/ann_0010   0   23                                  حَدِّ-#xad~#xd#xad~+i-                               NOUN+CASE_DEF_GEN          (NP*          xad~   -   -   -          *          *          *          *          *      -
nw/ann/00/ann_0010   0   24                                  -ِهِ#clitics#h#-hi                                   POSS_PRON_3MS       (NP*))))))       clitics   -   -   -          *          *          *          *          *)    (10)
nw/ann/00/ann_0010   0   25                                  "#DEFAULT#"#"                                           PUNC          (NP*        DEFAULT   -   -   -          *          *          *          *          *      (8
nw/ann/00/ann_0010   0   26                  البريـتزل#DEFAULT#Albrytzl#Al+brytzl                                DET+NOUN_PROP             *        DEFAULT   -   -   -     (PRODUCT)          *          *          *          *      -
nw/ann/00/ann_0010   0   27                                  "#DEFAULT#"#"                                           PUNC             *)        DEFAULT   -   -   -          *          *          *          *          *      8)
nw/ann/00/ann_0010   0   28                              جَوْلَةً#jawolap#jwlp#jawol+ap+N   NOUN+NSUFF_FEM_SG+CASE_INDEF_NOM          (NP*        jawolap   -   -   -          *          *          *          *          *     (11
nw/ann/00/ann_0010   0   29                      مُقَرِّرَةً#muqar~ar#mqrrp#muqar~ar+ap+N    ADJ+NSUFF_FEM_SG+CASE_INDEF_NOM           *))      muqar~ar   -   -   -          *          *          *          *          *     11)
```

Figure 5.10. Phrase dans le format conll

```
تَعافَى#taEAfaY#tEAfY#taEAfaY+(null)      taEAfaY      01      -2      الـرَّئِيسُ#ra}iys#Alr}ys#Al+ra}iys+u  DET+NOUN+CASE_DEF_NOM  (TOP(S(S(NP*  /  الأَمِيرْكِيِّ#>amoriykiy~#Al>myrky#Al+>amiyrokiy~+u  DET+ADJ+CASE_DEF_NOM  *)      ARG1
تَعافَى#taEAfaY#tEAfY#taEAfaY+(null)      taEAfaY      01      1       بـ#clitics#b#bi-  PREP  (PP*  /  سُرْعَةِ#suroEap#srEp#-suroE+ap+K  NOUN+NSUFF_FEM_SG+CASE_INDEF_GEN  (NP*))))      ARGM-MNR

بَدَأَ#bada>-a#bd>#-bada>+a      bada>-a      01      -6      الـرَّئِيسُ#ra}iys#Alr}ys#Al+ra}iys+u  DET+NOUN+CASE_DEF_NOM  (TOP(S(S(NP*  /  الأَمِيرْكِيِّ#>amoriykiy~#Al>myrky#Al+>amiyrokiy~+u  DET+ADJ+CASE_DEF_NOM  *)      ARG0
بَدَأَ#bada>-a#bd>#-bada>+a      bada>-a      01      1       جَوْلَةُ#jawolap#jwlp#jawol+ap+F  NOUN+NSUFF_FEM_SG+CASE_INDEF_ACC  (NP*  /  داخِلِيَّةُ#dAxiliy~#dAxlyp#dAxiliy~+ap+F  ADJ+NSUFF_FEM_SG+CASE_INDEF_ACC  *))))      ARG1
```

Figure 5.11. Quatre cas sources après l'exécution du script

6- Description du système basé sur les K plus proches voisins (K-PPV)

Généralement, les travaux sur l'intelligence artificielle utilisent des bibliothèques dédiées aux buts de l'étude. C'est une méthode de travail courante, acceptable et efficace par rapport aux objectifs de chaque recherche. Dans notre système de K-PPV, nous avons développé l'intégralité des scripts utilisés pour permettre une réelle personnalisation et adaptation de notre système à son fonctionnement. Notre système d'annotation arabe est basé sur le RàPC, les K plus proches voisins et testé sur un nombre important (5.291) de rôles sémantiques**.**

6.1 Architecture fonctionnelle

6.1.1- Scripts

Nous avons développé quatre scripts : `srl_system.py`, `similarity.py`, `k_nn.py` , `write.py` et `testing.py`, disponibles dans GitHub et en annexe B.

- **srl_system.py :** c'est le script central ou pilote qui fait appel aux trois (03) autres scripts. Il a pour fonction de lire le problème cible et d'envoyer les informations du problème cible en cours aux scripts responsables de son traitement. Avec un possible avertissement dans le cas où il y a une erreur dans les lignes des fichiers qui contiennent les cas cibles.

- **similarity.py :** c'est le premier script évoqué par le script central. Sa fonction est de calculer la similarité entre le problème cible et les problèmes sources de la base de cas. Il parcourt la base, dossier par dossier, fichier par fichier, et cas par cas, selon l'ordre de son organisation. Il retourne un nombre **K** de solutions de problèmes sources (classes) les plus similaires au problème cible.

- **k_nn.py :** il a pour objectifs de faire trois (03) opérations : la standardisation des valeurs des classes, l'accumulation des valeurs de chaque classe et la sélection de la classe maximale.

- **write.py :** c'est le dernier script appelé par le script principal. Il a pour fonction la création des dossiers, des fichiers et l'ajout des cas cibles. La création des dossiers et fichiers respecte l'organisation et les noms des fichiers dans la base de test. Ainsi, l'ordre d'organisation des cas cibles et leurs informations dans chaque fichier sont respectés. Le but de cette organisation est de permettre une vérification lors de la phase de test.

- **testing.py :** c'est un script utilisé principalement dans la partie test des résultats afin d'évaluer les résultats donnés par le système.

6.1.2- Description du fonctionnement

Notre système est composé d'une base de cas et de trois (03) modules fonctionnels qui utilisent les scripts précédents : `srl_system.py`, `similarity.py`, `k_nn.py` et `write.py` (Figure 5.12) :

- **1er module " Élaboration des cas sources et des cas cibles " :** regroupe les trois (03) scripts responsables de l'élaboration de la base de cas (cas sources) et la base de tests (cas cibles).
- **2ème module " Recherche et adaptation "** : regroupe quatre scripts responsables de la concrétisation des phases du cycle de raisonnement à partir de cas (RàPC), sauf pour la phase de révision (optionnelle) et la phase de test.
- **3ème module " Tests "** : regroupe les scripts des tests. Ce module peut contenir plusieurs scripts, selon les tests à effectuer.

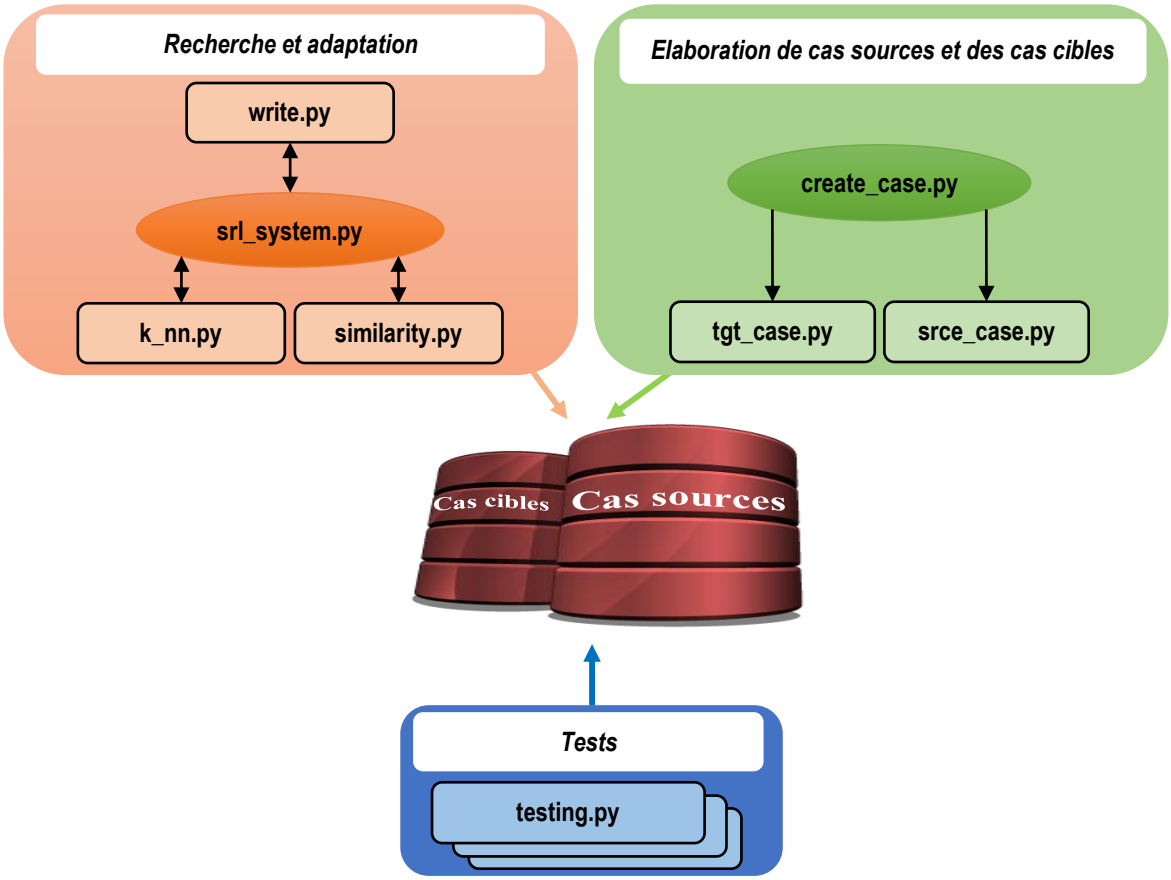


Figure 5.12. Architecture du système

6.2- Cas d'utilisation

Pour expliquer le fonctionnement de notre système, nous allons présenter un exemple d'attribution d'un rôle sémantique à un argument dans la phrase arabe :

وسيقام مشروع مماثل لابنية سكنية ومركز للتسوق على موقع مستشفى قديم قرب المسجد يحمل اسم قلعة اجياد نفسها، وسيبنى مستشفى جديد داخل المجمع.

Un projet similaire sera construit pour des bâtiments résidentiels et un centre commercial sur le site d'un ancien hôpital près de la mosquée, nommé la forteresse d'Agyad elle-même et un nouvel hôpital sera construit à l'intérieur du complexe.

- **Préparation des données :** la phrase arabe est dans son format original `conll` (Figure 5.13), avec deux (02) colonnes manquantes. L'utilisation d'un script permet de compléter ces données à partir d'OntoNotes 5.0, pour obtenir la phrase de la figure 5.14. La partie encadrée représente le cas en question (argument). Cet argument est (مشروع مماثل لابنية سكنية ومركز للتسوق), (`Projet similaire pour des bâtiments résidentiels et un centre commercial`).

- **Élaboration Cas Cible :** un cas cible de l'argument (Figure 5.14, rectangle rouge) est élaboré (Figure 5.15). Ce cas cible ne contient que la partie problème cible et manque la partie solution du problème cible. En d'autres termes, le rôle sémantique de cet argument n'est pas ajouté lors de l'élaboration du cas cible, afin de tester le système. Dans la figure 5.15, le problème cible est présenté sur plusieurs lignes (en réalité dans son fichier, il est dans une seule ligne).

- **Solution Problème Cible :** après les phases de remémoration et d'adaptation, une solution est proposée au problème cible de la figure 5.15. La solution est un rôle sémantique PropBank donné à cet argument. La figure 5.16 montre le problème cible et la solution du problème cible. Ainsi, le cas cible est maintenant complet.

- **Autres étapes :** la mémorisation de ce problème cible et les tests changent selon les besoins de test.

nw/ann/00/ann_0009	0	0	[WORD]	CONJ	(TOP(S(S*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	1	[WORD]	FUT_PART	(VP(PRT*)	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	2	[WORD]	IV3MS+IV_PASS+IVSUFF_MOOD:I	*	[LEMMA]	02	1	-	*	(V*)	*	*	-
nw/ann/00/ann_0009	0	3	[WORD]	NOUN+CASE_INDEF_NOM	(NP(NP(NP*	[LEMMA]	-	-	-	*	(ARG1*	*	*	-
nw/ann/00/ann_0009	0	4	[WORD]	ADJ+CASE_INDEF_NOM	*)	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	5	[WORD]	PREP	(PP*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	6	[WORD]	NOUN+NSUFF_FEM_SG+CASE_INDEF_GEN	(NP*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	7	[WORD]	ADJ+NSUFF_FEM_SG+CASE_INDEF_GEN	*)))	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	8	[WORD]	CONJ	*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	9	[WORD]	NOUN+CASE_INDEF_GEN	(NP(NP*)	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	10	[WORD]	PREP	(PP*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	11	[WORD]	DET+NOUN+CASE_DEF_GEN	(NP*))))	[LEMMA]	-	-	-	*	*)	*	*	-
nw/ann/00/ann_0009	0	12	[WORD]	PREP	(PP*	[LEMMA]	-	-	-	*	(ARGM-LOC*	*	*	-
nw/ann/00/ann_0009	0	13	[WORD]	NOUN+CASE_DEF_GEN	(NP*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	14	[WORD]	NOUN+CASE_INDEF_GEN	(NP(NP(NP*	[LEMMA]	-	-	-	*	*	(ARG0*	*	-
nw/ann/00/ann_0009	0	15	[WORD]	ADJ+CASE_INDEF_GEN	*)	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	16	[WORD]	NOUN+CASE_DEF_ACC	(NP*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	17	[WORD]	DET+NOUN+CASE_DEF_GEN	(NP*)))	[LEMMA]	-	-	-	*	*	*)	*	(5)
nw/ann/00/ann_0009	0	18	[WORD]	IV3MS+IV+IVSUFF_MOOD:I	(SBAR(S(VP*	[LEMMA]	05	-	-	*	*	(V*)	*	-
nw/ann/00/ann_0009	0	19	[WORD]	NOUN+CASE_DEF_ACC	(NP*	[LEMMA]	-	-	-	*	*	(ARG1*	*	-
nw/ann/00/ann_0009	0	20	[WORD]	NOUN+NSUFF_FEM_SG+CASE_DEF_GEN	(NP(NP*	[LEMMA]	-	-	-	(FAC*	*	*	*	(21
nw/ann/00/ann_0009	0	21	[WORD]	NOUN_PROP	(NP*)	[LEMMA]	-	-	-	*)	*	*	*	-
nw/ann/00/ann_0009	0	22	[WORD]	NOUN+CASE_DEF_GEN	(NP*	[LEMMA]	-	4	-	*	*	*	*	-
nw/ann/00/ann_0009	0	23	[WORD]	POSS_PRON_3FS	(NP*))))))))))))	[LEMMA]	-	-	-	*	*)	*)	*	(21)|21)
nw/ann/00/ann_0009	0	24	[WORD]	PUNC	*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	25	[WORD]	CONJ	*	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	26	[WORD]	FUT_PART	(S(VP(PRT*)	[LEMMA]	-	-	-	*	*	*	*	-
nw/ann/00/ann_0009	0	27	[WORD]	IV3MS+IV_PASS+IVSUFF_MOOD:I	*	[LEMMA]	01	-	-	*	*	*	(V*)	-
nw/ann/00/ann_0009	0	28	[WORD]	NOUN+CASE_INDEF_NOM	(NP*	[LEMMA]	-	-	-	*	*	*	(ARG1*	-
nw/ann/00/ann_0009	0	29	[WORD]	ADJ+CASE_INDEF_NOM	*)	[LEMMA]	-	-	-	*	*	*	*)	-
nw/ann/00/ann_0009	0	30	[WORD]	NOUN+CASE_DEF_ACC	(NP*	[LEMMA]	-	-	-	*	*	*	(ARGM-LOC*	-
nw/ann/00/ann_0009	0	31	[WORD]	DET+NOUN+CASE_DEF_GEN	(NP*))))	[LEMMA]	-	-	-	*	*	*	*)	-
nw/ann/00/ann_0009	0	32	[WORD]	PUNC	*))	[LEMMA]	-	-	-	*	*	*	*	-

Figure 5.13. Phrase dans son format original conll

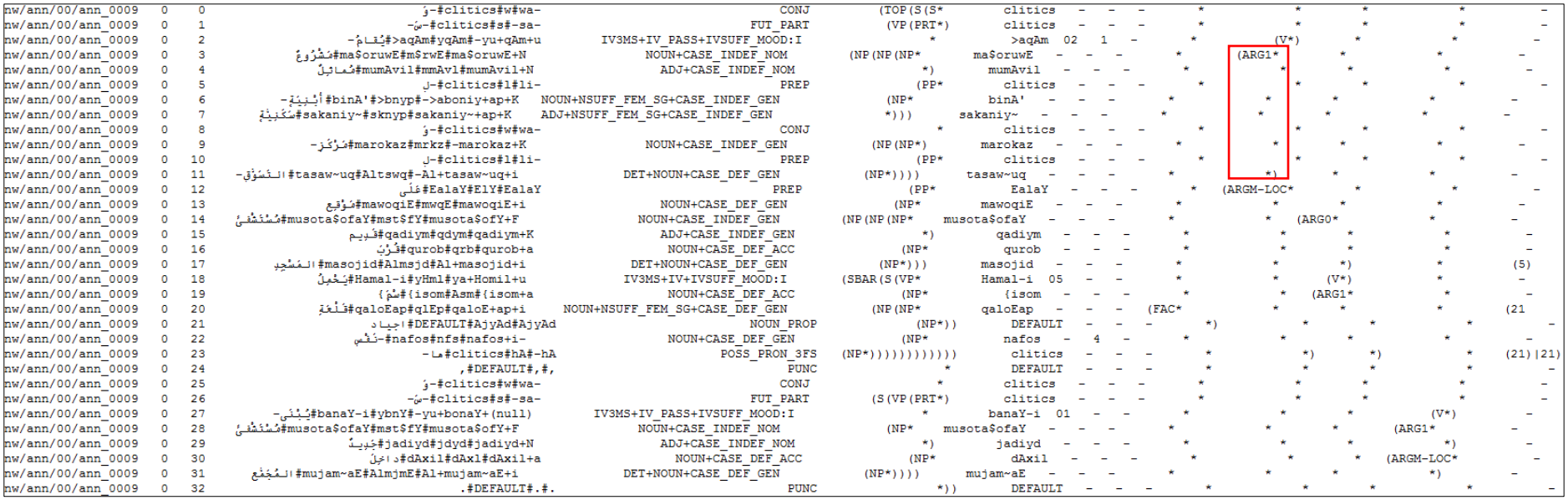


Figure 5.14. Phrase après l'exécution d'un script qui complète les colonnes manquantes

```
-يُقامُ#>aqAm#yqAm#-yu+qAm+u        >aqAm       02       1
مَشْرُوعٌ#ma$oruwE#m$rwE#ma$oruwE+N  NOUN+CASE_INDEF_NOM  (NP(NP(NP*  /
مُماثِلٌ#mumAvil#mmAvl#mumAvil+N  ADJ+CASE_INDEF_NOM  *)  /
لِ-#clitics#l#li-  PREP  (PP*  /
-أبْنِيَةٍ#binA'#>bnyp#->aboniy+ap+K  NOUN+NSUFF_FEM_SG+CASE_INDEF_GEN  (NP*  /
سَكَنِيَّةٍ#sakaniy~#sknyp#sakaniy~+ap+K  ADJ+NSUFF_FEM_SG+CASE_INDEF_GEN  *)))  /
وَ-#clitics#w#wa-  CONJ  *  /  -مَرْكَزِ#marokaz#mrkz#-marokaz+K  NOUN+CASE_INDEF_GEN  (NP(NP*)  /
لِ-#clitics#l#li-  PREP  (PP*  /
-التَسَوُّقِ#tasaw~uq#Altswq#-Al+tasaw~uq+i  DET+NOUN+CASE_DEF_GEN  (NP*))))
```

Figure 5.15. Un problème cible après l'exécution des scripts d'élaboration.

```
-يُقامُ#>aqAm#yqAm#-yu+qAm+u        >aqAm       02       1
مَشْرُوعٌ#ma$oruwE#m$rwE#ma$oruwE+N  NOUN+CASE_INDEF_NOM  (NP(NP(NP*  /
مُماثِلٌ#mumAvil#mmAvl#mumAvil+N  ADJ+CASE_INDEF_NOM  *)  /
لِ-#clitics#l#li-  PREP  (PP*  /
-أبْنِيَةٍ#binA'#>bnyp#->aboniy+ap+K  NOUN+NSUFF_FEM_SG+CASE_INDEF_GEN  (NP*  /
سَكَنِيَّةٍ#sakaniy~#sknyp#sakaniy~+ap+K  ADJ+NSUFF_FEM_SG+CASE_INDEF_GEN  *)))  /
وَ-#clitics#w#wa-  CONJ  *  /  -مَرْكَزِ#marokaz#mrkz#-marokaz+K  NOUN+CASE_INDEF_GEN  (NP(NP*)  /
لِ-#clitics#l#li-  PREP  (PP*  /
-التَسَوُّق#tasaw~uq#Altswq#-Al+tasaw~uq+i  DET+NOUN+CASE_DEF_GEN  (NP*))))
ARG1
```

Figure 5.16. Le problème cible et sa solution proposée par notre système (argument dans rectangle rouge)

7- Élaboration de la base utilisée pour le système basé sur un modèle d'apprentissage approfondi (Deep Learning)

Pour le modèle d'apprentissage approfondi, nous élaborons quatre (04) fichiers de type `CSV` (Extension .CSV) pour quatre genres d'expérimentations. Chaque fichier contient une base d'apprentissage (cas sources) et une base de test (cas cibles)

7.1- Scripts d'élaboration

7.1.1- Première catégorie

L'objectif de cette catégorie de scripts est d'élaborer un fichier qui contient les cas sources (base d'apprentissage) et les cas cibles (base de test) dans un seul fichier de type `CSV`.

À partir des scripts d'élaboration de la base utilisée pour le système de raisonnement à partir de cas (RàPC) et des K plus proches voisins (K-PPV), nous construisons trois (03) scripts :

- **create_CSV.py :** Script principal.
- **srce_CSV :** script dédié à l'élaboration de la base d'apprentissage.
- **tgt_roles_CSV.:** élaboration de la base de test.

À la fin, nous obtenons le fichier **`DeepLearning_Dataset_1.0.csv`**.

7.1.2- Deuxième catégorie

Au moment d'exécution du système basé sur le raisonnement à partir de cas (RàPC) et les K plus proches voisins (K-PPV) des fichiers de type `CSV` sont élaborés. Pour cela, nous avons adapté les trois (03) scripts du système basé sur le RàPC et les K-PPV : `srl_system.py, similarity.py` et `write.py`, et ajouter un nouveau script `Liste_knn_csv.py.`

À la fin de l'exécution du système basé sur les K plus proches voisins, nous obtenons trois (03) fichiers de type `CSV` :

- **tgt_error.csv (corpus de test):** contient les cas cibles qu'ont une mauvaise annotation par le système basé sur le RàPC et les K-PPV.
- **knn_tgt_errors.csv (corpus d'apprentissage):** il regroupe les K voisins des cas avec une mauvaise annotation.

- **liste_knn_all.csv (corpus d'apprentissage):** il regroupe l'intégralité des K voisins des cas cibles de la base de cas.

Le nombre **K** des cas voisins mémorisés est différent du **K** de la proche proposée.

7.2- Fichiers de type CSV (Dataset)

Pour les expérimentations du modèle d'apprentissage approfondi (Deep Learning), nous avons quatre (04) fichiers, chaque fichier contient une base d'apprentissage et une base de test :

- **DeepLearning_Dataset_1.0.csv :** issu de la première catégorie d'élaboration. Il contient les cas sources comme base d'apprentissage et les cas cibles comme base de test.
- **DeepLearning_Dataset_2.0.csv :** construit à partir de deux fichiers CSV : `liste_knn_all.csv` et la base de test du fichier `DeepLearning_Dataset_1.0.csv`.
- **DeepLearning_Dataset_3.0.csv :** il est la concaténation des deux fichiers `knn_tgt_errors.csv` et `tgt_error.csv`.
- **DeepLearning_Dataset_4.0.csv :** il est la concaténation des deux fichiers `liste_knn_all.csv` et `tgt_error.csv`.

Tous les fichiers CSV ont une structure similaire, la différence est dans les cas qui constitués la base d'apprentissage et la base de test. La figure 5.17 montre une partie de dix (10) cas de la base d'apprentissage d'un fichier de type CSV.

DeepLearning_Dataset 1.0.csv

```
1  تَعافَى#taEAfaY#tEAfY#taEAfaY+(null) taEAfaY 01 الرَئِيسُ#ra}iys#Alr}ys#Al+ra}iys+u ra}iys -2 BEFOR DET+NOUN+CASE_DEF_NOM (TOP(S(S(NP* ARG1
2  تَعافَى#taEAfaY#tEAfY#taEAfaY+(null) taEAfaY 01 بِ-#clitics#b#bi- clitics 1 AFTER PREP (PP* ARGM-MNR
3  -بَدَأَ#bada>-a#bd>#-bada>+a bada>-a 01 الرَئِيسُ#ra}iys#Alr}ys#Al+ra}iys+u ra}iys -6 BEFOR DET+NOUN+CASE_DEF_NOM (TOP(S(S(NP* ARG0
4  -بَدَأَ#bada>-a#bd>#-bada>+a bada>-a 01 جَوْلَةً#jawolap#jwlp#jawol+ap+F jawolap 1 AFTER NOUN+NSUFF_FEM_SG+CASE_INDEF_ACC (NP* ARG1
5  أَفْقَدَ#>afoqad#>fqd#>afoqad+a >afoqad 01 كَعْكٌ#kaEok#kEk#kaEok+u kaEok -4 BEFOR NOUN+CASE_DEF_NOM (S(S(NP* ARG0
6  أَفْقَدَ#>afoqad#>fqd#>afoqad+a >afoqad 01 بُوشِ#buw$#bw$#buw$ buw$ 1 AFTER NOUN_PROP (NP*) ARG1
7  أَفْقَدَ#>afoqad#>fqd#>afoqad+a >afoqad 01 وَعْيَ-#waEoy#wEy#waEoy+a- waEoy 2 AFTER NOUN+CASE_DEF_ACC (NP* ARG2
8  تَرَكَ#tarak-u#trk#tarak+a tarak-u 01 -سُقُوطِ-#suquwT#sqwT#-suquwT+u- suquwT -2 BEFOR NOUN+CASE_DEF_NOM (S(NP* ARG0
9  تَرَكَ#tarak-u#trk#tarak+a tarak-u 01 كَدْمَةً#kadomap#kdmp#kadom+ap+F kadomap 1 AFTER NOUN+NSUFF_FEM_SG+CASE_INDEF_ACC (NP*) ARG1
10 تَرَكَ#tarak-u#trk#tarak+a tarak-u 01 عَلَى#EalaY#ElY#EalaY EalaY 2 AFTER PREP (PP* ARGM-LOC
```

Figure 5.17. Partie du fichier DeepLearning_Dataset_1.0.csv

8- Conclusion

Dans ce chapitre, consacré au côté technique de notre approche, nous avons abordé les aspects de développement et de l'implémentation.

Nous avons détaillé la préparation des données de travail, ce qui permet leurs utilisations par d'autres chercheurs ou l'essai d'une autre méthode d'annotation sur ces données. Les détails sur les scripts Python développés donnent clairement les phases de développement de notre approche.

Dans le chapitre qui suit, nous allons présenter les diverses expérimentations réalisées, les résultats obtenus suivis d'une discussion et quelques interprétations.

Chapitre 6 :

Expérimentation et Discussion

Chapitre 6 : Expérimentation et Discussion

Chapitre 6 : Expérimentation et Discussion

1- Introduction

Dans ce chapitre, nous allons décrire les diverses expérimentations que nous avons menées pour la validation de notre approche et une discussion sur les résultats obtenus.

Nous commençons par présenter des statistiques et une comparaison des données que nous avons utilisées par rapport aux données utilisées par M. Diab.

Pour les expérimentations, nous les avons divisées sur deux grandes parties :

- Une série d'expérimentations sur le système basé sur le Raisonnement à Partir de Cas (RàPC) et l'approche des K plus proches voisins (K-PPV) ;
- Une série d'expérimentations avec l'utilisation du Deep Learning.

2- Données utilisées

À l'aide d'un script que nous avons développé, nous avons recueilli des statistiques (Tableau 6.1) sur nos données. Pour les données de M. Diab, les statistiques sont rassemblées à partir de notre lecture de ses travaux [11] [12] [13] [14]. Nous incluons une comparaison entre la source utilisée dans notre travail et celle des travaux de M. Diab.

Dans le présent travail, nous comptons 5.291 rôles sémantiques dans la base de tests, ce qui représente plus que le triple, comparé à 1.657 pour le corpus de M. Diab. Ainsi donc, le nombre de rôles sémantiques dans son corpus de test représente 31,31% par rapport à notre corpus. Pour les données d'entrainement, nous comptons 50.425 rôles sémantiques dans notre corpus contre 22.904 dans celui de M. Diab, ce qui ne représente même pas la moitié (45,42 %) de notre corpus.

Nous avons 29 arguments contre 24 dans le corpus de M. Diab. Ces arguments sont divisés en deux groupes :

- **Premier groupe :** regroupe les arguments numérotés, il y a neuf (09) arguments numérotés dans son corpus, contre cinq (05) dans notre corpus, ce qui ne facilite pas la tâche. Bien au contraire, généralement les arguments numérotés sont les plus faciles à détecter.

- **Deuxième groupe :** regroupe les arguments non-numérotés, dans ce groupe, certains types d'arguments (C-ARG et R-ARG) sont inclus dans nos données seulement. Il y a vingt-quatre (24)

arguments dans notre corpus et quinze (15) dans l'autre corpus. Ces types d'arguments sont les plus difficiles à annoter.

Un nombre important de rôles sémantiques dans le corpus de tests augmente considérablement la difficulté de l'expérimentation. Nous pensons que le nombre important de rôles sémantiques dans le corpus d'entrainement et la variété des types de rôles augmentent l'ambigüité de l'attribution des rôles sémantiques.

Notre travail concerne l'attribution des rôles sémantiques pour les arguments, nous n'incluons pas l'identification des limites d'arguments. C'est la métrique **Précision** qui évalue notre système d'annotation. Les tests consistent à donner un rôle sémantique parmi 29 rôles (Tableau 6.1) à chaque argument (cas cible).

Tableau 6.1. Informations sur les données de test

		Travaux de M. Diab		**Notre travail**	
Données	Entrainement	21 194		4 5011	
	Développement	1 710		5 414	
	Total	**22 904**		**50 425**	
	Test	**1 657**		**5 291**	
Rôles Sémantiques	Argument	ARG0 ARG0-STR ARG1 ARG1-PRD ARG1-STR	ARG2 ARG2-STR ARG3 ARG4	ARG0 ARG1 ARG2 ARG3 ARG4	
	ARG-M	ARGM ARGM-ADV ARGM-BNF ARGM-MNR ARGM-PRP ARGM-CAU ARGM-CND ARGM-REC	ARGM-DIR ARGM-NEG ARGM-TMP ARGM-DIS ARGM-EXT ARGM-LOC ARGM-PRD	ARGM-ADV ARGM-LOC ARGM-MNR ARGM-NEG ARGM-GOL ARGM-EXT ARGM-CAU	ARGM-PRP ARGM-TMP ARGM-COM
	C-ARG et R-ARG			C-ARG0 C-ARG1 C-ARG2 C-ARGM-ADV C-ARGM-LOC ARGMADV(C-ARG2 ARGM-ADV(C-ARG1	C-ARGM- PRP C-ARGM-TMP R-ARG0 R-ARG1 R-ARG2 R-ARGM-TMP R-ARGM-LOC
	Nombre de rôles sémantiques	**24**		**29**	

3- Expérimentations et interprétation du système de RàPC et K-PPV

3.1- Première expérimentation : Globale

Cette première expérimentation teste le fonctionnement général de l'approche proposée. Des tests plus méthodiques sont présentés ci-après dans les autres expérimentations. Les features utilisés dans ce test sont ceux cités dans le tableau 4.1 (Chapitre 4).

Le tableau 6.2 montre les résultats d'un premier test. La première ligne affiche la précision, selon le nombre de K voisins utilisés. Dans la deuxième ligne, on reprend les mêmes tests, mais avec certains features pondérés.

Tableau 6.2. Résultats de la précision avec variation des k voisins et pondération des features

	K = 3	K = 2	K = 1
Précision	52.39%	53.18%	53.18%
Précision avec pondération des features	59.55%	60.36%	60.36%

Dans la première et deuxième ligne du tableau 6.2, nous remarquons une légère augmentation de la précision lors de : la réduction des K voisins et la pondération de certains features. Dans la deuxième ligne, il y a une visible augmentation de la précision par rapport à la première ligne. Cela montre l'intérêt de la pondération des features importants dans l'annotation et que certains constituants de la phrase sont plus importants que d'autres lors de l'annotation.

Dans le tableau 6.3, nous avons refait les mêmes tests avec la phase de révision. Cette phase est automatique lors des tests. Elle consiste à mettre une annotation correcte avant sa mémorisation dans la base de cas, afin que ces cas corrigés soient utilisés comme cas sources pour de nouvelles annotations.

Tableau 6.3. Résultat de la précision avec l'étape de révision

	K = 3	K = 2	K = 1
Précision avec réversion	60.39%	62.42%	62.42%

Nous avons rajouté l'étape de **Révision** qui permet une mémorisation des cas cibles corrigés avec les cas sources de la base de cas. Ce test montre clairement l'intérêt de la mémorisation des cas cibles

corrects dans l'annotation de nouveaux arguments. Si un cas contient une erreur d'annotation, son impact sera surement négatif lors des prochaines annotations, d'où l'intérêt d'une étape de **Correction**.

3.2- Deuxième expérimentation : Sélection des features pertinents

Dans cette partie, nous utilisons un nombre plus important de features par rapport à la partie précédente. La colonne **Fonction** du tableau 6.4 donne un aperçu de ces features. L'objectif de cette expérimentation est de réduire le nombre des features utilisés, nous ne laissons que les features qui contribuent à une précision plus élevée.

Tableau 6.4. Première sélection des features pertinents

Fonction	Test 1	Test 2	Test 3	Test 4	Test 5
Attribut	O	X	O	X	X
Lemma attribut frameset	X	X	X	O	X
ID frameset	X	X	X	X	X
Colonne	X	X	X	X	X
Numéro de l'attribut dans la phrase	X	X	X	X	X
Position de l'attribut dans la phrase	X	X	X	X	X
Parties de discours de l'attribut	X	X	X	X	X
Analyse de l'attribut	X	X	X	X	X
1er constituant	X	O	O	X	X
Lemma 1er constituant	X	X	X	O	X
Position du 1er constituant par rapport à l'attribut	X	X	X	X	X
Endroit du 1er constituant par rapport à l'attribut	X	X	X	X	X
Parties de discours	X	X	X	X	X
Analyse	X	X	X	X	X
	53,86%	**49,29%**	**49,15%**	**48,06%**	**53,77%**

Dans le tableau 6.4, nous commençons par réduire les features (Attribut, Lemma attribut frameset, 1er constituant et Lemma 1er constituant). Par exemple, pour le **Test 4**, nous éliminons les features **Lemma attribut frameset** et **Lemma 1er constituant**. Nous obtenons le plus bas résultat (48,06%), ceci montre deux aspects :

- Il n'est pas nécessaire d'éliminer ces deux features ;
- Ces features sont importants pour une meilleure précision.

Le meilleur résultat (**53,86%**) est obtenu par l'élimination du feature **Attribut** (Test 1). Cependant, il n'y a pas une grande différence entre le résultat du **Test 1** (53,86%) et celui du **Test 5** (53,77%) qui regroupe l'intégralité des features. Ainsi, pour les prochains tests, nous gardons la totalité des features présents dans le tableau 6.4.

Lors des tests du tableau 6.5, nous gardons six (06) features que nous considérons comme importants et nécessaires pour n'importe quelle annotation. Ces features sont : ***Attribut***, ***Lemma attribut frameset***, ***1er constituant***, ***Lemma 1er constituant***, ***Parties de discours*** et ***Analyse***.

Le **Test 4** (Tableau 6.5) avec seulement les features importants donne un résultat de **56,54%**. Donc, si nous comparons ce test avec le **Test 5** (Tableau 6.4), nous constatons que parmi les features éliminés, il y a des features qui contribuent à la réduction de la précision du système.

Dans les prochains tests du tableau 6.5, nous intégrons les features éliminés un par un afin de voir leur contribution dans la précision du système. Le **Test 4** qui contient les six (06) features importants est utilisé comme référence pour les autres tests.

Pour exemple, dans le **Test 6** avec les six (06) features importants, nous ajoutons le feature **ID frameset**. Nous constatons une amélioration de la précision dans le **Test 6** par rapport au **Test 4**, cela signifie que le feature **ID frameset** contribue à l'amélioration de la précision.

Deux autres features contribuent à l'amélioration de la précision (Test 12 et 13) par rapport au **Test 4** qui sont : ***Position du 1er constituant par rapport à l'attribut*** et ***endroit du 1er constituant par rapport à l'attribut***.

D'autres features diminuent la précision, comme dans le **Test 9**, le feature **Position attribut dans la phrase** diminue la précision par rapport au **Test 4**. Même cas pour les features : ***colonne, numéro de l'attribut dans la phrase, parties de discours de l'attribut*** et ***analyse de l'attribut***.

Selon ces résultats, nous constatons que certains features améliorent la précision et d'autres la réduisent. Ainsi , pour les prochaines expérimentations, nous n'utilisons que les six (06) features et les trois (03) features : ***ID frameset, position du premier constituant par rapport à l'attribut*** et ***l'endroit du 1er constituant par rapport à l'attribut***.

Tableau 6.5. Deuxième sélection des features pertinents

Fonction	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
Attribut	X	X	X	X	X	X	X	X	X	X	
Lemma attribut frameset	X	X	X	X	X	X	X	X	X	X	X
ID frameset			X								
Colonne				X							
Numéro de l'attribut dans la phrase					X						
Position attribut dans la phrase						X					
Parties de discours de l'attribut							X				
Analyse de l'attribut								X			
1er constituant	X		X	X	X	X	X	X	X	X	X
Lemma 1er constituant	X	X	X	X	X	X	X	X	X	X	X
Position du 1er constituant par rapport à l'attribut									X		
Endroit du 1er constituant par rapport à l'attribut										X	
Parties de discours	X	X	X	X	X	X	X	X	X	X	X
Analyse	X	X	X	X	X	X	X	X	X	X	X
	56,54%	**53,24%**	**57,73%**	**55,43%**	**55,43%**	**54,65%**	**55,50%**	**56,34%**	**59,45%**	**58,02%**	**53,46%**

3.3- Troisième expérimentation : Réduction des features

Les neuf (09) features présents dans le tableau 6.6 sont ceux issus de la précédente étape de sélection. Nous testons l'apport des (03) trois features ***ID frameset***, ***Position du 1er constituant par rapport à l'attribut*** et ***Positionnement du 1er constituant par rapport à l'attribut*** (sélectionnés lors de la précédente étape), pour cela, nous éliminons ces trois features un par un. Pour exemple, dans le **Test 15**, nous éliminons le feature **Endroit du 1er constituant par rapport à l'attribut**.

Nous constatons que tous les features issus de la précédente étape sont tous importants, car le meilleur résultat **Test 18** est obtenu dans le cas où les trois (03) features sont présents.

Tableau 6.6. Réduction des features

Fonction	T15	T16	T17	T18
Attribut	X	X	X	X
Lemma attribut frameset	X	X	X	X
ID frameset	X	X	**O**	X
1er constituant	X	X	X	X
Lemma 1er constituant	X	X	X	X
Position du 1er constituant par rapport à l'attribut	X	**O**	X	X
Endroit du 1er constituant par rapport à l'attribut	**O**	X	X	X
Parties de discours	X	X	X	X
Analyse	X	X	X	X
	60,15%	**59,36%**	**60,27%**	**61,04%**

3.4- Quatrième expérimentation : Affectation des Poids

3.4.1- Première partie

Dans cette partie, nous affectons des poids aux features. Dans le tableau 6.7, nous affectons le poids (4) à un feature à la fois, afin de constater si l'affectation d'un poids à ce feature contribue à l'amélioration du résultat.

Tableau 6.7. Affectation initiale des poids

Fonction	T19	T20	T21	T22	T23	T24	T25	T26	T27
Attribut	**4**	1	1	1	1	1	1	1	1
Lemma attribut frameset	1	**4**	1	1	1	1	1	1	1
ID frameset	1	1	**4**	1	1	1	1	1	1
1er constituant	1	1	1	**4**	1	1	1	1	1
Lemma 1er constituant	1	1	1	1	**4**	1	1	1	1
Position du 1er constituant par rapport à l'attribut	1	1	1	1	1	**4**	1	1	1
Endroit du 1er constituant par rapport à l'attribut	1	1	1	1	1	1	**4**	1	1
Parties de discours	1	1	1	1	1	1	1	**4**	1
Analyse	1	1	1	1	1	1	1	1	**4**
	58,00%	**65,07%**	**60,85%**	**61,50%**	**62,57%**	**58,15%**	**61,08%**	**60,89%**	**59,40%**

Par exemple, l'affectation d'un poids au feature **Attribut** (Test 19) diminue la précision (58,00%) par rapport au **Test 18** (61,04%) (Tableau 6.6). Contrairement, le feature **Lemma attribut frameset** améliore la précision (65,07%). Les features qui améliorent la précision sont : ***Lemma attribut frameset*** *(65,07%)*, ***1er constituant*** *(61,50%)*, ***Lemma 1er constituant*** *(62,57%)* et ***Endroit du 1er constituant par rapport à l'attribut*** *(61,08%)*.

Après l'observation des features les plus importants depuis le tableau 6.7, nous procédons dans la deuxième partie à une affectation plus précise des poids dans les tableaux 6.8 et 6.9.

3.4.2- Deuxième partie

Nous commençons par l'affectation des poids (Test 28 et 29) (Tableau 6.8) aux features des deux meilleurs résultats du tableau 6.7 : ***Lemma attribut frameset*** *(65,07%)* et ***Lemma 1er constituant*** *(62,57%)*. La meilleure précision obtenue est de 67,47%.

Tableau 6.8. Première affectation des poids

Fonction	T28	T29	T30	T31	T32	T33	T34	T35	T36	T37
Attribut	1	1	1	1	1	1	1	1	1	1
Lemma attribut frameset	4	5	5	5	6	7	6	6	6	6
ID frameset	1	1	1	1	1	1	1	1	1	1
1er constituant	1	1	3	3	3	3	2	4	3	3
Lemma 1er constituant	3	4	4	4	4	4	4	4	3	5
Position du 1er constituant par rapport à l'attribut	1	1	1	1	1	1	1	1	1	1
Endroit du 1er constituant par rapport à l'attribut	1	1	1	2	2	2	2	2	2	2
Parties de discours	1	1	1	1	1	1	1	1	1	1
Analyse	1	1	1	1	1	1	1	1	1	1
	67,20%	**67,47%**	**68,11%**	**68,49%**	**68,72%**	**67,75%**	**67,79%**	**68,43%**	**67,71%**	**68,90%**

Tableau 6.9. Deuxième affectation des poids

Fonction	T38	T39	T40	T41	T42	T43	T44	T45	T46	T47	T48
Attribut	1	1	1	1	1	1	1	1	1	2	1
Lemma attribut frameset	6	6	6	6	6	6	6	6	6	6	6
ID frameset	1	1	1	2	1	1	1	1	1	1	1
1er constituant	3	3	3	3	3	3	3	3	3	3	3
Lemma 1er constituant	5	5	5	5	5	5	5	5	5	5	5
Position du 1er constituant par rapport à l'attribut	1	1	1	1	1	1	1	1	1	1	2
Endroit du 1er constituant par rapport à l'attribut	3	4	5	4	4	4	4	4	3	3	3
Parties de discours	1	1	1	1	2	3	2	2	2	2	2
Analyse	1	1	1	1	1	1	2	3	2	2	2
	68,94%	**69,07%**	**69,06%**	**68,94%**	**69,19%**	**68,77%**	**69,53%**	**69,28%**	**69,58%**	**69,34%**	**69,26%**

Dans les **Tests 30 à 37** (Tableau 6.8) et les **Tests 38 à 40** (Tableau 6.9), en plus des deux premiers features, nous pondérons le troisième et le quatrième meilleurs feature du tableau 6.7. La meilleure précision est obtenue dans le **Test 39** (69,07%).

Enfin, lors des **Tests 41 à 48**, nos pondérons les autres features, la meilleure précision **69,58%** est obtenue dans le **Test 46**. Les features pondérés et leurs poids dans ce test sont : ***Lemma attribut frameset*** (6), ***1er constituant*** (3), ***Lemma 1er constituant*** (5), ***Endroit du 1er constituant par rapport à l'attribut*** (3), ***Parties de discours*** (2) et ***Analyse*** (2).

3.5- Quatrième expérimentation : Différents paramètres de test

Nous nous basons sur les paramètres du meilleur résultat obtenu (69,58%, Test 46), puis nous procédons à une série de tests.

3.5.1- Paramétrer le K des K plus proches voisins (K-PPV)

Dans le tableau 6.10, nous faisons varier le paramètre *K* des K plus proches voisins . Nous constatons que lors de l'augmentation du paramètre *K*, la précision diminue. Ainsi, le meilleur résultat est obtenu pour K = 1.

Tableau 6.10. Paramétrer les K voisins

K	1	3	5	10	15	20	25	30	40	50
	69,58%	69,38%	69,17%	67,51%	67,41%	67,03%	66,6%	66,01%	64,97%	63,90%

3.5.2- Sans les étapes de Révision et Mémorisation

Dans ce test, nous nous interrogeons sur l'apport des étapes **Révision** et **Mémorisation** du cycle de raisonnement à partir de cas pour l'annotation. Pour cela, nous éliminons, les étapes Révision et Mémorisation. La précision obtenue est de **68,45%**. Certainement, cela ne constitue pas une grande différence dans la meilleure précision (69,58%). En nombre de cas, cela ne représente que 60 cas incorrects.

3.5.3- Corpus déjà mémorisé

Nous examinons notre approche dans le cas où l'intégralité des cas de test sont présents dans la base de cas. Ceci permet de tester le principe de l'approche, de ne pas faire deux fois la même annotation. Pour cela, nous avons intégré dans la base des cas, les cas cibles avec leurs rôles sémantiques.

Nous obtenons une précision de **99,30%** (37 cas incorrects), donc un taux d'erreur de **0,7%** seulement. Probablement, les cas incorrects sont dus à l'égalité de la similarité globale entre des cas qui ont des rôles sémantiques différents. Un nombre réduit de cas incorrects montre que si un cas cible est mémorisé dans la base de cas le système ne refait pas deux fois la même annotation. Pour un système d'aide à l'annotation, cela réduit considérablement l'effort des annotateurs.

4- Expérimentation d'Apprentissage Profond (Deep Learning)

Comme le Deep Learning est une approche nouvelle dans l'intelligence artificielle, nous expérimentons pour la première fois dans la littérature, les avantages de Deep Learning pour un système d'annotation des rôles sémantiques dans la langue Arabe.

4.1- Données utilisées

Pour un système basé sur l'apprentissage profond (Deep Learning), il est nécessaire d'adapter les cas sources (base d'apprentissage) et les cas cibles (base de test) à un format spécifique de type CSV, pour qu'ils puissent être utilisés par le modèle Deep Learning.

Les quatre (04) fichiers (Chapitre 5) en format CSV (DeepLearning_Dataset_1.0.csv, DeepLearning_Dataset_2.0.csv, DeepLearning_Dataset_3.0.csv et DeepLearning_Dataset_4.0.csv) contiennent une base d'apprentissage et une base de test.

Chaque ligne représente un cas source pour la base d'apprentissage et un cas cible pour la base de test. Un cas source ou un cas cible sont un ensemble de features similaires aux features[30] cités dans le tableau 6.6, plus le rôle sémantique du cas à la fin de la ligne.

4.2- Expérimentation du Deep Learning sur l'ensemble des cas cibles et sources

Le modèle Deep Learning que nous utilisons au début de cette expérimentation a quatre (04) couches : une couche d'entrée, deux couches cachées et une couche sortie. Les paramètres comme : fonction d'activation, nombre d'époques, etc., sont déterminés lors des expérimentations.

Dans cette première expérimentation (Tableau 6.11), nous utilisons le Deep Learning pour l'annotation des rôles sémantiques, sans aucune relation avec le système basé sur le raisonnement à partir de cas et les K plus proches voisins.

[30] Les features Attribut et 1er constituant sont disponibles dans les fichiers CSV, mais ils ne sont pas utilisés, pour des raisons de mémoire dans le modèle Deep Learning lors d'exécution

Nous utilisons le fichier `DeepLearning_Dataset_1.0.csv`, il contient un total de 55.716 cas : 50.425 cas dans la base d'apprentissage (cas sources) et 5.291 cas dans base de test (cas cibles).

Tableau 6.11. Première expérimentation Deep Learning sur la première base

Nbr époques / Fonction d'activation dernière couche	*10 époques*	*20 époques*	*40 époques*	*60 époques*	**Btch size**
Sigmoid	*78,41%*	-	-	-	**90**
	79,11%	*78,37%*	*76,45%*	*76,56%*	**45**
	79,07%	*79,03%*	*77,73%*	*77,05%*	**30**
	78,41%	*76, 67%*	*76,39%*	*74,57%*	**15**
	78,43%	*77,11%*	*75,24%*	-	**5**
Relu	-	*22,60%*	-	-	**30**
	-	-	-	-	**15**
	22,60%	-	-	-	**5**

Nous remarquons qu'avec l'augmentation du nombre d'époques et le Batch size, la précision est réduite. Les résultats obtenus sont meilleurs que ceux obtenus avec le raisonnement à partir de cas et les K plus proches voisins. Ceci est évident vu la nouveauté du Deep Learning dans l'intelligence artificielle et son apport considérable dans l'amélioration des systèmes d'apprentissages automatiques.

Dans une deuxième expérimentation (Tableau 6.12), nous prenons ces paramètres : Nombre d'époques = 10 et Batch size = 30. Puis nous testons notre modèle sur plusieurs couches cachées et un nombre de neurones variés.

Les résultats obtenus dans le tableau 6.12 montrent que le changement du nombre de neurones et le nombre de couches cachées n'ont pas un grand impact sur la précision.

Tableau 6.12. Deuxième expérimentation Deep Learning sur la première base

1er couche cachée	2ème couche cachée	3ème couche cachée	4ème couche cachée	Précision
120	60	-	-	**79,49%** **78,83%**
240	60	-	-	**79,28%**
300	60	-	-	**79,47%**
600	60	-	-	**79,72%**
1000	60	-	-	**79,83%**
1000	500	-	-	**80,13%**
1000	1000	-	-	**80,40%**
500	500	-	-	**80,15%** **79,47%**
300	150	-	-	**78,51%**
300	300	-	-	**79,79%**
400	400	-	-	**79,89%**
120	120	-	-	**79,77%**
120	120	120	-	**79,28%**
240	120	60	-	**78,94%** **79,83%**
240	180	120	60	**78,30%**

4.3- Hybridation entre le système basé sur le RàPC et le K-PPV et le modèle Deep Learning

Nous faisons une hybridation entre le système du raisonnement à partir de cas et le modèle Deep Learning. Nous exécutons notre système basé sur le RàPC et les K-PPV selon les paramètres du meilleur résultat. Lors de cette exécution, des bases d'apprentissages et de tests sont créées à partir des K plus proches voisins (K = 15) de chaque cas de la base de test (Chapitre 5 – Section 7).

Les paramètres du modèle Deep Learning sont :

- ✓ Couches cachées = 2 ;
- ✓ Nombre d'époques = 10 ;
- ✓ Batch size = 30 ;
- ✓ Nombre de neurones 1er couche cachée = 120 ;
- ✓ Nombre de neurones 2ème couche cachée = 60.

4.3.1- Première expérimentation

Nous utilisons le fichier `DeepLearning_Dataset_2.0.csv`, il contient un total de 84.656 cas : 79.365 (5.291 x 15) cas dans la base d'apprentissage et 5.291 cas dans base de test. La différence de fichier CSV par rapport au précédent fichier est le corpus d'apprentissage. Ce dernier contient les 15 cas sources les plus similaires à chaque cas cible.

La précision obtenue est de **85,37%**, nous avons 4517 cas corrects de 5.291 cas de test. Une augmentation de presque 6% par rapport aux expérimentations précédentes.

4.3.2- Deuxième expérimentation

Nous utilisons le fichier `DeepLearning_Dataset_3.0.csv`, il contient un total de 25.744 cas : 24.135 (1.609 x 15) cas dans la base d'apprentissage et 1.609 cas dans base de test. Donc, ce fichier contient seulement, les cas incorrects par le précédent système basé sur le RàPC et leurs 15 cas sources les plus similaires.

Le système basé sur le raisonnement à partir de cas (RàPC) et les K-PPV a obtenu **3.682** cas corrects (**69,58%**), les 1.609 cas incorrects restants ont été corrigés pas le modèle Deep Learning pour obtenir **881** cas corrects de 1.609 (**54,75%**). Ainsi, nous avons **4.563** cas correctement annotés sur **5.291** cas cibles, ce qui représente une précision de **86,24%**.

4.3.3- Troisième expérimentation

Nous utilisons le fichier `DeepLearning_Dataset_4.0.csv`, il contient un total de 80.974 cas : 79.365 (5.291 x 15) cas dans la base d'apprentissage et 1.609 cas dans base de test.

Le résultat obtenu est de **66,00%**, nous avons 1.062 cas corrects de 1.609 cas de test. Si nous ajoutons ces cas annotés par le Deep Learning aux cas annotés par le précédent système 3.682, nous obtenons un total de **4.744** cas correct de 5.291 cas cibles, ainsi nous avons une précision de **89,66%**.

5- Discussion et interprétation

Nous avons effectué plusieurs expérimentations sur le système basé sur l'approche du raisonnement à partir de cas et les K plus proches voisins, puis des expérimentations sur une hybridation entre le précédent système et le Deep Learning. Le but étant de : sélectionner les meilleurs features, le meilleur poids pour chaque feature, la meilleure précision, etc.

- La première expérimentation a pour but de tester le fonctionnement global du système sans contraintes du côté des features utilisés et les poids. Nous avons obtenu une précision de 62,42%, ce qui montre la possibilité d'annotation en se basant sur le cycle du RàPC et les K-PPV.

- Dans une deuxième expérimentation, nous avons pour objectif de réduire les features afin de n'avoir que les features qui aident à l'amélioration de la précision. Un nombre plus grand de features est utilisé par rapport à la première expérimentation.

  Après une série d'élimination, nous obtenons une meilleure précision de **61,04%** et une liste de neuf (09) features :

  - ✓ Trois (03) features en relation avec l'attribut de la phrase : Attribut, Lemma attribut frameset et ID frameset ;
  - ✓ Quatre (04) features en relation avec le constituant ou l'argument de la phrase : 1er constituant, Lemma 1er constituant, Position du 1er constituant par rapport à l'attribut et Endroit du 1er constituant par rapport à l'attribut ;
  - ✓ Deux (02) features qui représentent la structure globale de la phrase : Parties de discours et Analyse.

- La troisième expérimentation a pour objectif d'avoir le meilleur poids pour chaque feature. Nous constatons depuis le premier test dans le tableau 6.7 que quatre (04) features sont plus importants que d'autres, car ils ont une relation directe avec l'attribut et l'argument de la phrase. Trois (03) features en relation avec l'argument (1er constituant, Lemma 1er constituant et Endroit du 1er constituant par rapport à l'attribut) et un feature d'attribut (Lemma attribut frameset).

  Dans un deuxième test (affectation des poids), en plus des quatre (04) meilleurs features, deux (02) autres features sont importants : Parties de discours et Analyse. Ces deux features sont les seuls qui donnent des informations sur la structure de la phrase. Nous obtenons une précision de **69,58%** et qui représente la meilleure précision obtenue pour ce système.

Dans le tableau 6.13 et la figure 6.1, nous présentons des statistiques sur les rôles sémantiques pour le meilleur résultat obtenu. Certains rôles sémantiques contribuent à l'amélioration de la précision telle que : ARG4, ARGM-TMP, R-ARG1, ARG1 et ARG2. Le total des cas de l'ARG0 (1.196) et l'ARG1 (2.052) représente plus que la moitié **61%** (3.248) du total des rôles sémantiques, ainsi, une correcte annotation

de ces deux rôles sémantiques contribue à l'amélioration de la précision. L'ARG1 a une annotation de **77,63%**, ce qui constitue une meilleure précision par rapport à la précision globale **69,58%**. Quant à l'ARG0, il a une précision de **69,31%**, qui est légèrement en dessous de la précision globale.

Nous remarquons que les rôles numérotés sont les mieux détectés et les rôles M sont les plus difficiles à annoter.

Tableau 6.13. Statistiques sur la précision pour chaque rôle

Rôle sémantique	Correct	Incorrect	Total	Précision
ARG4	4	0	4	100 %
ARGM-TMP	256	66	322	79,50 %
R-ARG1	46	12	58	79,31 %
ARG1	1593	459	2052	77,63 %
ARG2	502	207	709	70,80 %
ARG0	829	367	1196	69,31 %
ARG3	23	13	35	65,71 %
R-ARG0	47	34	81	58,02 %
ARGM-ADV	150	149	299	50,16 %
C-ARGM-ADV	1	1	2	50 %
R-ARGM-LOC	1	1	2	50 %
ARGM-MNR	73	75	148	49,32 %
ARGM-LOC	92	95	187	49,19 %
ARGM-PRP	34	47	81	41,97 %
ARGM-EXT	3	4	7	42,85 %
C- ARG1	11	17	28	39,28 %
ARGM-CAU	15	40	55	27,27 %
C-ARG2	1	3	4	25 %
ARGM-GOL	1	6	7	14,28 %
ARGM-COM	0	2	2	0 %
ARGM-NEG	0	2	2	0 %
R-ARG2	0	2	2	0 %
R-ARGM-TMP	0	1	1	0 %
C-ARG0	0	1	1	0 %
C-ARGM-PRP	0	1	1	0 %
C-ARM-TMP	0	1	1	0 %
C-ARGM-LOC	0	1	1	0 %
ARGM-ADV(C-ARG2	0	1	1	0 %
ARGM-ADV(C-ARG1	0	2	2	0 %
	3682	1609	5291	69,58 %

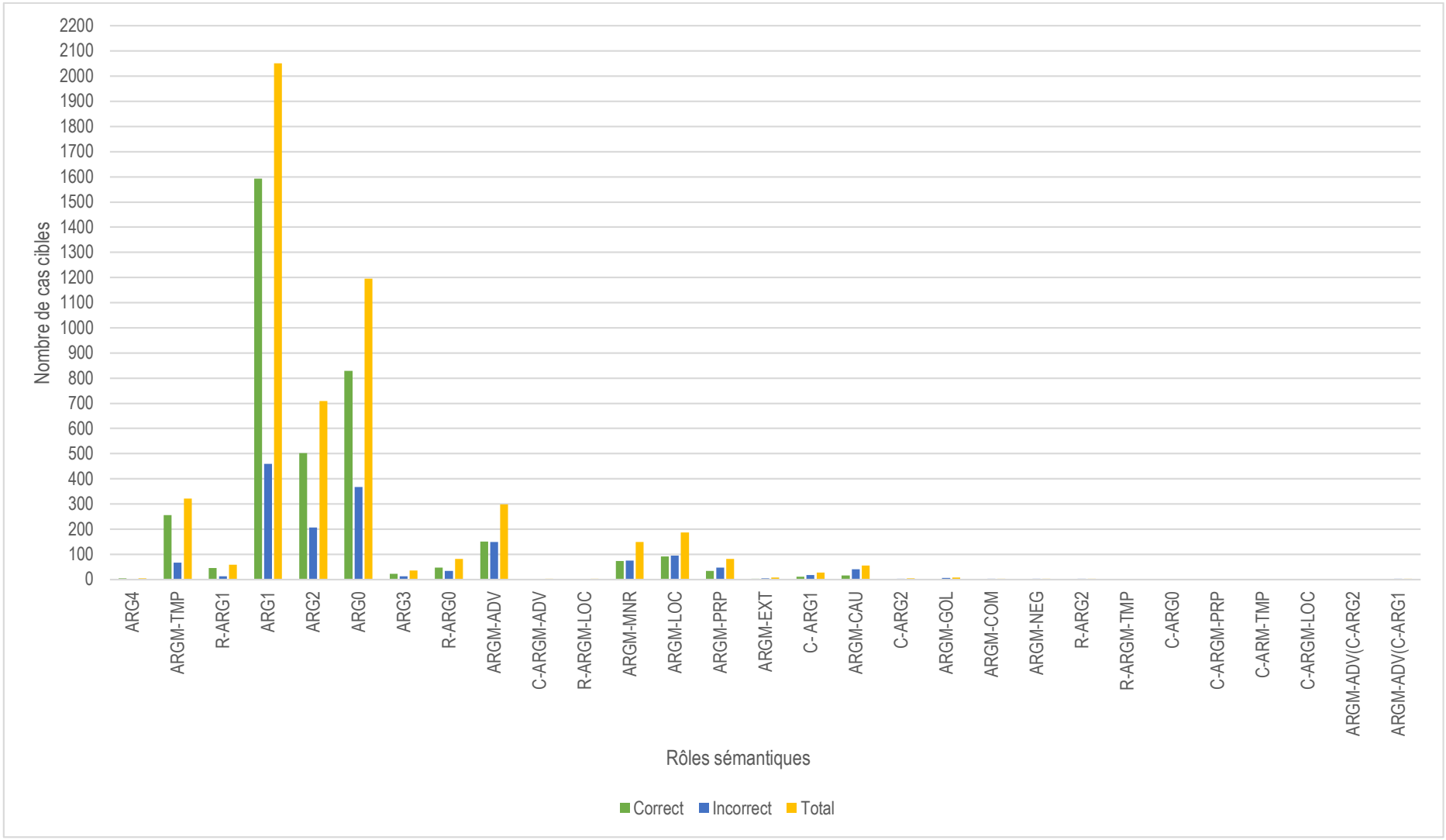


Figure 6.1. Histogramme groupé du meilleur résultat obtenu dans le système basé sur le RàPC et les K-PPV

- La quatrième expérimentation regroupe les tests liés au RàPC :
    - ✓ Le meilleur résultat est obtenu dans le cas où il y a un seul K voisin. Généralement lors de l'utilisation des K plus proches voisins dans le RàPC, K est égal à 1.
    - ✓ Les étapes Révision et Mémorisation participent à l'amélioration de la précision, mais ils n'ont pas un grand rôle dans la précision. Leur rôle est dans l'augmentation du corpus initial, la vérification de l'annotation avant la mémorisation et la construction de corpus annotés.
    - ✓ La mémorisation des cas de tests dans la base de cas aide l'approche ne fait pas deux fois la même annotation.

Dans d'autres expérimentations, nous avons intégré le Deep Learning afin d'exploiter la puissance de cette nouvelle approche d'intelligence artificielle dans un processus d'annotation des rôles sémantiques. Nous avons construit un modèle Deep Learning pour l'annotation des rôles sémantiques. Sa puissance par rapport aux approches classiques de l'IA, nous a permis d'obtenir des résultats au tour de **79,00%**. Mais dans notre cas d'étude, nous ne cherchons pas seulement une meilleure précision, mais aussi :

- Le problème de l'annotation et le manque de corpus annotés obligent le développement d'un système qui sert à la construction de corpus annotés ;
- Ne pas travailler dans une boite noire comme dans le Deep Learning ;
- L'augmentation du corpus initial ;
- La révision des annotations par les annotateurs.

Ces points et d'autres encore ne sont pas pris en charge par le modèle de Deep Learning.

Dans d'autres expérimentations, nous faisons une hybridation entre le système du RàPC et le modèle Deep Learning. Pour cela, nous exploitons les avantages du cycle RàPC et la sélection des cas par l'approche des K plus proches voisins. Nous obtenons une précision de **89,66%**, presque une augmentation de **20,08%** (1.062 cas) par rapport au meilleur résultat du système RàPC.

La figure 6.2 montre une récapitulation des principaux résultats obtenu par le système basé sur le raisonnement à partir cas, le modèle Deep Learning et leur l'hybridation.

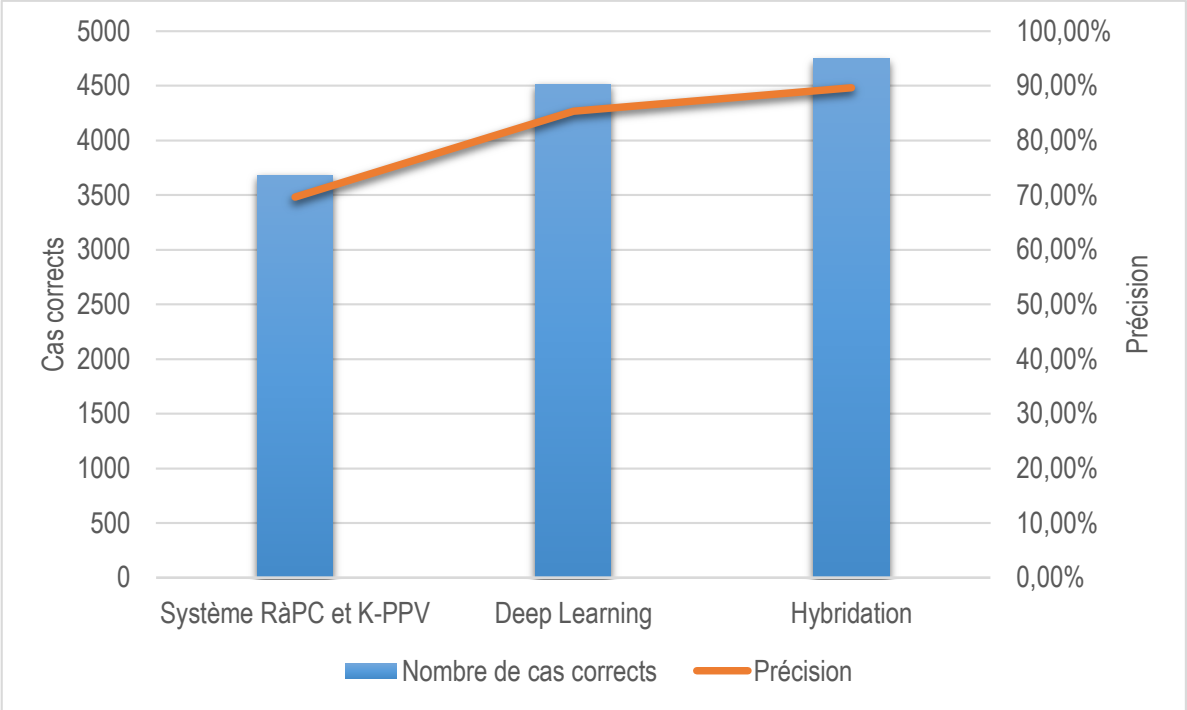


Figure 6.2. Principaux résultats

6- Conclusion

Les diverses expérimentations effectuées sur le système du cycle RàPC et les K-PPV permettent la validation de la démarche pour : la variation des k voisins, la pondération de certains attributs et l'intérêt d'une mémorisation après vérification des annotations du système. Aussi, l'exploitation du Deep Learning avec les avantages du premier système a permis une augmentation de **20%** dans la précision.

Conclusion et Perspectives

Conclusion et Perspectives

Nous avons abordé un aspect pertinent et délicat du traitement automatique du langage naturel (TALN), puisqu'il entre dans de nombreuses disciplines et il joue un rôle important dans la compréhension automatique du langage naturel.

La langue d'étude, en l'occurrence l'Arabe, ne facilite pas les tâches, à cause du manque considérable de travaux et de ressources dans cette langue, cela est justifiable par les difficultés liées au traitement automatique de cette langue et le manque de chercheurs intéressés par le développement du traitement automatique de la langue Arabe. C'est aussi une langue exprimée dans des pays où la contribution scientifique est très limitée par rapport au taux mondial et l'importance de la populations.

Notre travail a consisté en la proposition d'une nouvelle méthode d'annotation des rôles sémantiques pour divers objectifs et le développement d'un outil dédié à cette tâche.

Notre contribution à travers cette méthode basée sur le raisonnement à partir de cas (RàPC) réside dans :

- L'utilisation d'un cycle de raisonnement à partir de cas (RàPC) ;
- L'exploitation du Deep Learning pour l'annotation des rôles sémantiques ;
- L'exploitation des données CoNLL pour l'annotation automatique des rôles sémantiques ;
- Cela constitue la première fois qu'une méthode d'annotation des rôles sémantiques dans la langue arabe est testée sur de larges données ;
- La possibilité d'élaboration d'un outil d'annotation pour la construction de données annotées et de corpus nécessaire pour d'autres tâches comme la traduction.

Cette démarche prouve :

- La possibilité et l'intérêt d'une utilisation des expériences précédentes pour une tâche et une langue délicate dans le traitement automatique du langage ;
- Elle constitue un avantage pour les travaux sur l'Arabe, car il y a un manque de ressources annotées, donc chaque phrase annotée constitue une expérience nécessaire pour la proportion des corpus ;

- Un grand intérêt vient du fait que cette méthode peut être adaptée à un système d'aide à l'annotation ;
- Le développement d'un outil, la disponibilité et la préparation claire des données annotées, montrent l'intérêt de ces données et ouvre la voie pour les chercheurs pour proposer d'autres contributions sur cette question.

Nous envisageons, ultérieurement, de tester et de comparer la méthode avec d'autres algorithmes des K plus proches voisins et d'autres approches de l'intelligence artificielle, tels que machine à vecteur de support, arbres de décision, etc.

Ce travail ouvre plusieurs voies de recherche sur la langue Arabe et l'annotation des rôles sémantiques :

- Un outil d'aide à l'annotation PropBank ;
- Outils pour la construction de corpus nécessaire pour d'autres domaines du traitement automatique du langage naturel ;
- Adapter la méthode pour un système d'annotation adapté à VerbNet pour construire un corpus basé sur cette ressource ;
- Les données présentées ouvrent des perspectives de recherches sur l'annotation et sur d'autres tâches du traitement automatique du langage naturel arabe.

Références Bibliographiques

Références bibliographiques

[1] M. M. Richter et R. O. Weber, « General Aspects », in Case-based reasoning : a textbook, Berlin, Allemagne : Springer, 2013.

[2] S. Wess, K.-D. Althoff, et M. M. Richter, Topics in Case-Based Reasoning, First European Workshop, EWCBR-93, vol. 837. Allemagne : Springer Berlin Heidelberg, 1994.

[3] I. D. Watson, Progress in Case-Based Reasoning, First United Kingdom Workshop, vol. 1020. Allemagne : Springer Berlin Heidelberg, 1995.

[4] M. Veloso et A. Aamodt, Case-Based Reasoning Research and Development, First International Conference, vol. 1010. Allemagne : Springer Berlin Heidelberg, 1995.

[5] S. Zaidi, « Une plateforme pour la construction d'ontologie en arabe : Extraction des termes et des relations à partir de textes (Application sur le Saint Coran) », Thèse de Doctorat, Université Badji Mokhtar Annaba, Annaba, Algérie, 2013.

[6] D. Gildea et D. Jurafsky, « Automatic Labeling of Semantic Roles », Computational Linguistics., vol. 28, no 3, p. 245‑288, Septembre. 2002.

[7] X. Carreras et L. Màrquez, « Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling », CoNLL-2004 Shared Task on Semantic Role Labeling, Boston, États-Unis, 2004.

[8] X. Carreras et L. Màrquez, « Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling », in Proceedings of the Ninth Conference on Computational Natural Language Learning, Stroudsburg, Pennsylvanie, États-Unis, p. 152–164, 2005.

[9] W. Léchelle, « Utilisation de représentations de mots pour l'étiquetage de rôles sémantiques suivant FrameNet », Thèse de Master, Université de Montréal, Québec, Canada, 2014.

[10] D. Jurafsky et J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, vol. Third Edition draft. 2017.

[11] M. Diab, M. Alkhalifa, S. ElKateb, C. Fellbaum, A. Mansouri et M. Palmer, « SemEval-2007 Task 18: Arabic Semantic Labeling », in Proceedings of the Fourth International Workshop on Semantic Evaluations, Prague, République tchèque, p. 93–98, 2007.

[12] M. Diab et A. Moschitti, « Semantic parsing of modern standard Arabic », in International Conference Recent advances in natural language processing, Borovets, Bulgaria, p. 162‑166, 2007.

[13] M. Diab, A. Moschitti et D. Pighin, « CUNIT: A Semantic Role Labeling System for Modern Standard Arabic », in Proceedings of the Fourth International Workshop on Semantic Evaluations, Prague, République tchèque, p. 133–136, 2007.

[14] M. Diab, A. Moschitti et D. Pighin, « Semantic Role Labeling Systems for Arabic using Kernel Methods », in Proceedings of ACL-08: HLT, Columbus, Ohio, États-Unis, p. 798–806, 2008.

[15] R. C. Schank, Dynamic Memory: A Theory of Reminding and Learning in Computers and People. New York, États-Unis: Cambridge University Press, 1982.

[16] N. Armaghan, « Contribution à un système de retour d'expérience basé sur le raisonnement à partir de cas conversationnel : application à la gestion des pannes de machines industrielles », Thèse de Doctorat, Institut national polytechnique de Lorraine, Nancy, France, 2009.

[17] F. Gebhardt, A. Voß, W. Gräther et B. Schmidt-Belz, Reasoning with Complex Cases. Norwell, Massachusetts, États-Unis: Kluwer Academic Publishers, 1997.

[18] M. Minsky, « A Framework for Representing Knowledge », Massachusetts Institute of Technology, Cambridge, Massachusetts, États-Unis, Juin 1974.

[19] M. K. Haouchine, « Remémoration guidée par l'adaptation et maintenance des systèmes de diagnostic industriel par l'approche du raisonnement à partir de cas. », Thèse de Doctorat, Université de Franche-Comté, France, 2009.

[20] J. Kolodner, Case-Based Reasoning. San Mateo, Californie, États-Unis: Morgan Kaufmann Publishers, 1993.

[21] I. Watson et F. Marir, « Case-Based Reasoning: A Review », Knowledge Engineering Review, vol. 9, p. 327 – 354, 1994.

[22] B. Fuchs, « Representation des connaissances pour le raisonnement a partir de cas : le systeme rocade ROCADE », Thèse de doctorat, Université Jean Monnet de Saint-Etienne, France, 1997.

[23] S. Guessoum, K. Deghdegh, R. Benali et H. Djedi, « La modélisation informatique au service du raisonnement médical : le RàPC pour l'aide au diagnostic du cancer broncho-pulmonaire primitif (CBP) », Revue des Maladies Respiratoires., vol. 34, p. A88, Janvier 2017.

[24] A. Khelassi et M. Amin Chick, « Fuzzy knowledge-intensive case based classification for the detection of abnormal cardiac beats », Electronic Physician, vol. 4, p. 565, 2012.

[25] A. Khelassi, « Reasoning System for Computer Aided Diagnosis with explanations aware computing for medical applications », Thèse de Doctorat, Université Abou-Bekr Belkaid Tlemcen, Tlemcen, Algérie, 2014.

[26] M. S. Meflah, « Un serveur dédié à la recherche d'informations médicales basé sur le raisonnement à partir de cas », Thèse de Magister, Université Kasdi Merbah, Ouargla, Algérie, 2009.

[27] D. Mansouri, A. Mille et A. Hamdi-Cherif, « Adaptive Delivery of Trainings Using Ontologies and Case-Based Reasoning », Arabian Journal for Science and Engineering, vol. 39, no 3, p. 1849‑1861, Mars 2014.

[28] N. Dendani-Hadiby et M. T. Khadir, « A fault diagnosis application based on a combination case-based reasoning and ontology approach », International Journal of Knowledge-Based and Intelligent Engineering Systems, vol. 17, no 4, p. 305‑317, Novembre 2013.

[29] H. Abed et N. Rezoug, « Intégration de la logique floue dans le raisonnement à base de cas : application dans le domaine du bâtiment. », STIC'09, M'sila, Algérie, 2009.

[30] H. Meguehout, T. Bouhadada et M.-T. Laskri, « Un Raisonnement à Partir de Cas pour la Traduction Automatique Arabe-Français Basée sur la Sémantique », CEC-TAL'2013, Université du Québec à Montréal, Canada, 2013.

[31] J. Lieber, « Contributions to the design of case-based reasoning systems », Habilitation à diriger des recherches, Université Henri Poincaré - Nancy 1, France, 2008.

[32] M. M. Richter, « The Knowledge Contained in Similarity Measures », First International Conference on Case-Based Reasoning Research and Development, Sesimbra, Portugal, 23 Octobre 1995.

[33] L. Lamontagne et G. Lapalme, « Raisonnement à base de cas textuel - état de l'art et perspectives futures », Revue d'Intelligence Artificielle, vol. 16, no 3, p. 339–366, 2002.

[34] T. Roth-Berghofer, « Developing maintainable case-based reasoning systems: applying SIAM to empolis orenge », GWEM'03: German workshop on experience management, vol. 67, Lucerne, Suisse, 2003.

[35] A. Aamodt et E. Plaza, « Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches », AI Communication. IOS Press, vol. 7, no 1, p. 39–59, Mars 1994.

[36] A. Mille, « Traces Based Reasoning (TBR) Definition, illustration and echoes with story telling », Rapport Technique, Janvier 2006.

[37] D. Dhouib, « Aide multicritère au pilotage d'un processus basée sur le raisonnement à partir de cas », Thèse de Doctorat, Université Paris 8 Vincennes-Saint Denis, France, 2011.

[38] B. Fuchs, J. Lieber, A. Mille et A. Napoli, « Une première formalisation de la phase d'élaboration du raisonnement à partir de cas », 14ème atelier francophone de raisonnement à partir de cas, Besançon, France, 2006.

[39] R. Bergmann, J. Kolodner et E. Plaza, « Representation in case-based reasoning », The Knowledge Engineering Review, vol. 20, no 03, p. 209, Septembre 2005.

[40] D. W. Aha, D. Kibler et M. K. Albert, « Instance-based learning algorithms », Machine Learning, vol. 6, no 1, p. 37‑66, Janvier 1991.

[41] R. Bergmann, Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Lecture Notes in Artificial Intelligence, vol. 2432, Springer-Verlag Berlin Heidelberg, 2002.

[42] M. Lenz et H.-D. Burkhard, « Case retrieval nets: Basic ideas and extensions », KI-96: Advances in Artificial Intelligence, vol. 1137, G. Görz et S. Hölldobler, Allemagne: Springer Berlin Heidelberg, p. 227‑239, 1996.

[43] R. O. Weber, K. D. Ashley et S. BrüNinghaus, « Textual case-based reasoning », The Knowledge Engineering Review, vol. 20, no 03, p. 255, Septembre 2005.

[44] E. Plaza, « Cases as terms: A feature term approach to the structured representation of cases », Case-Based Reasoning Research and Development, vol. 1010, M. Veloso et A. Aamodt, Allemagne: Springer Berlin Heidelberg, p. 265‑276, 1995.

[45] J. R. Quinlan, « Induction of decision trees », Machine Learning, vol. 1, no 1, p. 81‑106, Mars 1986.

[46] R. Lopez De Mantaras, D. Mcsherry, D. Bridge et D. Leake, « Retrieval, reuse, revision and retention in case-based reasoning », The Knowledge Engineering Review, vol. 20, no 03, p. 215, Septembre. 2005.

[47] B. Fuchs, J. Lieber, A. Mille et A. Napoli, « Towards a Unified Theory of Adaptation in Case-Based Reasoning », Case-Based Reasoning Research and Development, vol. 1650, K.-D. Althoff, R. Bergmann et L. K. Branting, Allemagne: Springer Berlin Heidelberg, p. 104‑117, 1999.

[48] W. Wilke et R. Bergmann, « Techniques and knowledge used for adaptation during case-based problem solving », Tasks and Methods in Applied Artificial Intelligence, vol. 1416, A. Pasqual del Pobil, J. Mira et M. Ali, Allemagne: Springer Berlin Heidelberg, p. 497‑506, 1998.

[49] H. Karoui, R. Kanawati et L. Petrucci, « COBRAS: Cooperative CBR System for Bibliographical Reference Recommendation », Advances in Case-Based Reasoning : 8th European Conference, ECCBR 2006 Fethiye, Turkey, Proceedings, T. R. Roth-Berghofer, M. H. Göker et H. A. Güvenir, Allemagne: Springer Berlin Heidelberg, 2006, p. 76–90, September 2006.

[50] A. Cordier, B. Fuchs, J. Lieber et A. Mille, « Acquisition interactive des connaissances d'adaptation intégrée aux sessions de raisonnement à partir de cas — Principes, architecture IakA et prototype KayaK », 15ème atelier sur le raisonnement à partir de cas - RàPC-07, p. 71–84, Grenoble, France, 2007.

[51] S. K. Pal et S. C. K. Shiu, Foundations of Soft Case-Based Reasoning: Pal/Soft Case-Based Reasoning. Hoboken, NJ, États-Unis: John Wiley & Sons, Inc., 2004.

[52] R. Bénard et P. De Loor, « La révision et l'apprentissage de cas pour les simulations temps-réel en réalité virtuelle », 16ème atelier du raisonnement à partir de cas, p. 137, Grenoble, France, 2007.

[53] D. W. Aha, L. A. Breslow et H. Muñoz-Avila, « Conversational Case-Based Reasoning », Applied Intelligence, vol. 14, no 1, p. 9–32, Janvier 2001.

[54] R. D. Burke, K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro et S. Schoenberg, « Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System », AI Magazine, vol. 18, no 2, 1997.

[55] K.-D. Althoff, « Case-Based Reasoning », Handbook on Software Engineering and Knowledge Management, p. 549–587, 2001.

[56] Zhi-Wei Ni, Shan-Lin Yang, Long-Shu Li et Rui-Yu Jia, « Integrated case-based reasoning », Proceedings of the 2003 International Conference on Machine Learning and Cybernetics, vol. 3, p. 1845–1849, Xi'an, Chine, 2003.

[57] Q. Pradet, « Annotation en rôles sémantiques du français en domaine spécifique », Thèse de Doctorat, Université Paris Diderot (Paris 7), France, 2015.

[58] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross et K. J. Miller, « Introduction to WordNet: An On-line Lexical Database », International Journal of Lexicography, vol. 3, no 4, p. 235‑244, Décembre. 1990.

[59] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw et R. Weischedel, « OntoNotes: The 90% Solution », Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, p. 57–60, New York City, États-Unis, 2006,

[60] S. S. Pradhan, E. Loper, D. Dligach et M. Palmer, « Semeval-2007 task-17: English lexical sample, SRL and all words », Proceedings of the Fourth International Workshop on Semantic Evaluations, Prague, République tchèque, p. 87–92, 2007.

[61] R. Navigli, « A Quick Tour of Word Sense Disambiguation, Induction and Related Approaches », in SOFSEM 2012: Theory and Practice of Computer Science, vol. 7147, M. Bieliková, G. Friedrich, G. Gottlob, S. Katzenbeisser et G. Turán, Allemagne: Springer Berlin Heidelberg, p. 115‑129, 2012.

[62] E. Boros, R. Besançon, O. Ferret et B. Grau, « Étiquetage en rôles événementiels fondé sur l'utilisation d'un modèle neuronal » Actes de la 21ème conférence sur le Traitement Automatique des Langues Naturelles, p. 25–35, Marseille, France, 2014. Vintinième

[63] W. Léchelle et P. Langlais, « Utilisation de représentations de mots pour l'étiquetage de rôles sémantiques suivant FrameNet », Actes de la 21ème conférence sur le Traitement Automatique des Langues Naturelles, p. 36–45, Marseille, France, 2014.

[64] O. Michalon, « Modélisation probabiliste de l'interface syntaxe sémantique à l'aide de grammaires hors contexte probabilistes Expériences avec FrameNet », Actes des 16e Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues, p. 1–12, Marseille, France, 2014.

[65] F. Hadouche, « Annotation syntaxico-sémantique des actants en corpus spécialisé », Thèse de Doctorat, Université de Montréal, Québec, Canada, 2011.

[66] J. S. Gruber, « Studies in Lexical Relations », Thèse de Doctorat, Massachusetts Institute of Technlogy, États-Unis, 1965.

[67] C. J. Fillmore, « The case for case », Texas Symposium on Linguistic Universals, États-Unis, 1968.

[68] M. Djemaa, « Stratégie domaine par domaine pour la création d'un FrameNet du français : annotations en corpus de cadres et rôles sémantiques », Thèse de Doctorat, Université Sorbonne Paris Cité, France, 2017.

[69] F. Hadouche, G. Lapalme et M.-C. L'Homme, « Attribution de rôles sémantiques aux actants des lexies verbales », Actes de la 18e conférence sur le Traitement Automatique des Langues Naturelles, Montpellier, France, 2011.

[70] S. Wu, « Semantic Role Labeling Tutorial: Part 2 (Supervised Machine Learning methods) », NAACL-HLT 2013, Atlanta, États-Unis, Juin 2013.

[71] L. Barque, « Annotation sémantique de corpus », 9iémes Journées de formation du réseau LTT, France, 13 Septembre 2013.

[72] D. Gildea et D. Jurafsky, « Automatic labeling of semantic roles », Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, p. 512–520, Stroudsburg, Pennsylvanie, États-Unis, 2000.

[73] G. Miller, R. Beckwith, C. Fellbaum, D. Gross et K. Miller, « Five Papers on WordNet », Cognitive Science Laboratory. Princeton University, CSL Report 43, 1990.

[74] A. Abdulhay, « Constitution d'une ressource sémantique arabe à partir de corpus multilingues alignés », Thèse de Doctorat, Université de Grenoble, France, 2012.

[75] G. Lebboss, « Contribution à l'analyse sémantique des textes arabes », Thèse de Doctorat, Université Paris 8 Vincennes à Saint-Denis, France, 2016.

[76] C. F. Baker, C. J. Fillmore et J. B. Lowe, « The Berkeley FrameNet project », COLING-ACL '98: Proceedings of the Conference, p. 86‑90, Montreal, Canada, 1998.

[77] N. Ghazzawi, « Du terme prédicatif au cadre sémantique : méthodologie de compilation d'une ressource terminologique pour les termes arabes de l'informatique », Thèse de Doctorat, Université de Montréal, Québec, Canada, 2016.

[78] J. Ruppenhofer, M. Ellsworth, M. R. Petruck, C. R. Johnson et J. Scheffczyk, FrameNet II: Extended theory and practice. Berkeley, California, États-Unis : Institut für Deutsche Sprache, Bibliothek, 2016.

[79] B. Levin, English Verb Classes and Alternations: A Preliminary Investigation. Université de Chicago, États-Unis : University of Chicago Press, 1993.

[80] K. K. Schuler, « Verbnet: A Broad-coverage, Comprehensive Verb Lexicon », Thèse de Doctorat, Université de Pennsylvania, États-Unis, 2005.

[81] I. Falk, « Making Use of Existing Lexical Resources to Build a Verbnet like Classification of French Verbs », Thèse de Doctorat, Université de Lorraine, France, 2012.

[82] M. Palmer, D. Gildea et P. Kingsbury, « The Proposition Bank: An Annotated Corpus of Semantic Roles », Computational Linguistics, vol. 31, no 1, p. 71‑106, Mars 2005.

[83] C. Bonial, O. Babko-Malaya, J. D. Choi, J. Hwang et M. Palmer, « Propbank annotation guidelines », Center for Computational Language and Education Research, Institute of Cognitive Science, University of Colorado at Boulder, Décembre 2010.

[84] C. Bonial, J. Bonn, K. Conger, J. Hwang, M. Palmer, et N. Reese, « English propbank annotation guidelines », Center for Computational Language and Education Research, Institute of Cognitive Science, University of Colorado at Boulder, Juillet 2015.

[85] C. Mouton, « Ressources et méthodes semi-supervisées pour l'analyse sémantique de texte en français », Thèse de Doctorat, Université Paris Sud (Paris 11), France, 2010.

[86] D. Das, N. Schneider, D. Chen et N. A. Smith, « Probabilistic Frame-semantic Parsing », Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, p. 948–956, Stroudsburg, Pennsylvanie, États-Unis, 2010.

[87] C. A. Thompson, R. Levy et C. D. Manning, « A Generative Model for Semantic Role Labeling », Machine Learning: ECML 2003, vol. 2837, p. 397‑408, N. Lavrač, D. Gamberger, H. Blockeel, et L. Todorovski, Allemagne: Springer Berlin Heidelberg, 2003.

[88] M. Fleischman, N. Kwon et E. Hovy, « Maximum entropy models for FrameNet classification », Proceedings of the 2003 conference on Empirical methods in natural language processing, p. 49‑56, Sapporo, Japon, vol. 10, 2003.

[89] A.-M. Giuglea et A. Moschitti, « Shallow Semantic Parsing Based on FrameNet, VerbNet and PropBank », Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence, p. 563–567, Riva Del Garda, Italy, 2006.

[90] L. Shi et R. Mihalcea, « An Algorithm for Open Text Semantic Parsing », Proceedings of the 3rd Workshop on RObust Methods in Analysis of Natural Language Data, p. 59–67, Geneva, Suisse, 2004.

[91] K. Erk et S. Padó, « SHALMANESER - A Toolchain For Shallow Semantic Parsing », Proceedings of LREC 2006, Genoa, Italy, 2006.

[92] R. S. Swier et S. Stevenson, « Unsupervised semantic role labelling », Proceedings of EMNLP, p. 95–102, Barcelona, Espagne, 2004.

[93] H. Fürstenau et M. Lapata, « Semi-Supervised Semantic Role Labeling », in Proceedings of the 12th Conference of the European Chapter of the ACL, p. 220–228, Athens, Greece, 2009.

[94] L. Suanmali, N. Salim et M. S. Binwahlan, « SRL-GSM: A Hybrid Approach based on Semantic Role Labeling and General Statistic Method for Text Summarization », Journal of Applied Sciences, vol. 10, no 3, p. 166‑173, Mars 2010.

[95] G. Melli, Y. Wang, Y Liu, M. M. Kashani, Z. Shi, B. Gu, A. Sarkar et F. Popowich, « Description of squash, the sfu question answering summary handler for the duc-2005 summarization task », Proceedings of HLT/EMNLP Document Understanding Workshop, Vancouver, Canada, 2005.

[96] G. Melli, Z. Shi, Y. Wang, Y. Liu, A. Sarkar et F. Popowich, « Description of squash, the sfu question answering summary handler for the duc-2006 summarization task », Proceedings of Document Understanding Conference, Brooklyn, Etat-Unis, 2006.

[97] P. Moreda, H. Llorens, E. Saquete et M. Palomar, « The influence of Semantic Roles in QA: A comparative analysis », Procesamiento del lenguaje Natural, no 41, p. 55–62, 2008.

[98] S. Narayanan et S. Harabagiu, « Question answering based on semantic structures », Proceedings of the 20th International Conference on Computational Linguistics, Genève, Suisse, 2004.

[99] R. Sun, J. Jiang, Y. F. Tan, H. Cui, T.-S. Chua et M.-Y. Kan, « Using Syntactic and Semantic Relation Analysis in Question Answering », Proceedings of the Fourteenth Text REtrieval Conference, Gaithersburg, Maryland, États-Unis, 2005.

[100] D. Shen et M. Lapata, « Using Semantic Roles to Improve Question Answering », Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), p. 12–21, Prague, République tchèque, 2007.

[101] A. Lakhfif et M. T. Laskri, « A frame-based approach for capturing semantics from Arabic text for text-to-sign language MT », International Journal of Speech Technology, vol. 19, no 2, p. 203‑228, 2015.

[102] H. C. Boas, « Bilingual FrameNet Dictionaries for Machine Translation », Proceedings of the Third International Conference on Language Resources and Evaluation, p. 1364‑1371, Las Palmas, Espagne, 2002.

[103] M. Bazrafshan et D. Gildea, « Semantic Roles for String to Tree Machine Translation », Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), p. 419–423, Sofia, Bulgaria, 2013.

[104] M. Bazrafshan et D. Gildea, « Comparing Representations of Semantic Roles for String-To-Tree Decoding », Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 1786–1791, Doha, Qatar, 2014.

[105] C. Lo et D. Wu, « MEANT: An Inexpensive, High-accuracy, Semi-automatic Metric for Evaluating Translation Utility via Semantic Frames », Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, p. 220–229, Stroudsburg, Pennsylvanie, États-Unis, 2011.

[106] A. Chuchunkov, A. Tarelkin et I. Galinskaya, « Applying HMEANT to English-Russian Translations », Proceedings of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, p. 43–50, Doha, Qatar, 2014.

[107] A. H. Osman, N. Salim, M. S. Binwahlan, R. Alteeb et A. Abuobieda, « An improved plagiarism detection scheme based on semantic role labeling », Applied Soft Computing, vol. 12, no 5, p. 1493‑1502, Mai 2012.

[108] B. Xie, R. J. Passonneau, L. Wu et G. G. Creamer, « Semantic Frames to Predict Stock Price Movement », in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), p. 873–883, Sofia, Bulgaria, 2013.

[109] O. De Clercq, M. Schuhmacher, S. P. Ponzetto et V. Hoste, « Exploiting frameNet for content-based book recommendation », CBRecSys at ACM RecSys, Proceedings, p. 14–21, Foster City, États-Unis, 2014.

[110] J. Malmaud, E. Wagner, N. Chang et K. Murphy, « Cooking with Semantics », Proceedings of the ACL 2014 Workshop on Semantic Parsing, p. 33–38, Baltimore, Maryland, États-Unis, 2014.

[111] W. Zaghouani, « Critical Survey of the Freely Available Arabic Corpora », Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools, Reykjavik, Iceland, 2014.

[112] R. Steinberger, B. Pouliquen, M. Kabadjov, J. Belyaeva et E. Van der Goot, « JRC-NAMES: A Freely Available, Highly Multilingual Named Entity Resource », Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, p. 104–110, Hissar, Bulgaria, 2011.

[113] Y. Benajiba, P. Rosso et J. M. BenedíRuiz, « ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy », Computational Linguistics and Intelligent Text Processing, vol. 4394, p. 143‑153, A. Gelbukh, Allemagne: Springer Berlin Heidelberg, 2007.

[114] B. Mohit, N. Schneider, R. Bhowmick, K. Oflazer et N. A. Smith, « Recall-Oriented Learning of Named Entities in Arabic Wikipedia », Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, p. 162–173, Avignon, France, 2012.

[115] M. Azab, H. Bouamor, B. Mohit et K. Oflazer, « Dudley North visits North London: Learning When to Transliterate to Arabic », Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p. 439–444, Atlanta, Georgia, États-Unis, 2013.

[116] M. Attia, A. Toral, L. Tounsi, M. Monachini et J. van Genabith, « An Automatically Built Named Entity Lexicon for Arabic », Proceedings of the Seventh conference on International Language Resources and Evaluation. European Language Resources Association., Valletta, Malta, 2010.

[117] N. Habash, B. Mohit, O. Obeid, K. Oflazer, N. Tomeh et W. Zaghouani, « QALB: Qatar Arabic Language Bank », Proceedings of Qatar Annual Research Conference (ARC-2013), Doha, Qatar, 2013.

[118] A. Alfaifi, E. Atwell et G. Abuhakema, « Error Annotation of the Arabic Learner Corpus », in Language Processing and Knowledge in the Web, p. 14‑22, Berlin, Heidelberg, Allemagne, 2013.

[119] M. I. Alkanhal, M. A. Al-Badrashiny, M. M. Alghamdi et A. O. Al-Qabbany, « Automatic Stochastic Arabic Spelling Correction With Emphasis on Space Insertions and Deletions », IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no 7, p. 2111‑2122, Septembre 2012.

[120] R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, A. Mansouri, J. Choi, M. Foster, A. Hawwary, C. Bonial, J. D. Hwang, M. El-Bachouti, C. Greenberg, R. Belvin et A. Houston, « OntoNotes Release 5.0 with OntoNotes DB Tool v0.999 beta ». 28 Septembre 2012.

[121] K. Dukes et N. Habash, « Morphological Annotation of Quranic Arabic », Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 2010.

[122] N. Schneider, B. Mohit, K. Oflazer et N. A. Smith, « Coarse Lexical Semantic Annotation with Supersenses: An Arabic Case Study », Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2, p. 253–258, Stroudsburg, Pennsylvanie, États-Unis, 2012.

[123] K. Shereen, G. Roger et K. Gerry, « An Arabic Tagset for the Morphosyntactic Tagging of Arabic », Corpus Linguistics 2001, Lancaster University, Lancaster, Royaume-Uni, 2001.

[124] S. Hammami, L. H. Belguith et A. B. Hamadou, « Arabic anaphora resolution: corpora annotation with coreferential links », The International Arab Journal of Information Technology. IAJIT, vol. 6, no 5, p. 480‑489, Novembre 2009.

[125] S. ElKateb, W. Black, H. Rodríguez, M. Alkhalifa, P. Vossen, A. Pease et C. Fellbaum, « Building a WordNet for Arabic », LREC 2006 Conference, Genoa, Italy, 2006.

[126] C. Fellbaum, WordNet : An Electronic Lexical Database. Cambridge, MA : The MIT Press, 1998.

[127] P. Vossen, EuroWordNet : A multilingual database with lexical semantic networks. Dordrecht, Pays-Bas: Springer Netherlands, 1998.

[128] H. Rodríquez, D. Farwell, J. Ferreres, M. Bertran, M. Alkhalifa et M. A. Martí, « Arabic WordNet: Semi-automatic Extensions using Bayesian Inference », Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Maroc, 2008.

[129] A.-B. M. Sharaf et E. Atwell, « Knowledge representation of the Quran through frame semantics: A corpus-based approach », Proceedings of the Fifth Corpus Linguistics Conference, Liverpool, Royaume-Uni, 2009.

[130] N. Ghneim, E. Karhely et W. Sa, « First Step of Building an Arabic FrameNet (AFN) », 13th International Business Information Management Conference, Marrakech, Maroc, 2009.

[131] M. Maamouri, A. Bies, T. Buckwalter et W. Mekki, « The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus », NEMLAR Conference on Arabic Language Resources and Tools, Cairo, Egypt, 2004.

[132] M. Maamouri et A. Bies, « Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools », Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, p. 2–9, Stroudsburg, Pennsylvanie, États-Unis, 2004.

[133] A. I. El-taher, H. M. Abo Bakr, I. Zidan et K. Shaalan, « An Arabic CCG approach for determining constituent types from Arabic Treebank », Journal of King Saud University - Computer and Information Sciences, vol. 26, no 4, p. 441‑449, Décembre 2014.

[134] K. Dukes et T. Buckwalter, « A Dependency Treebank of the Quran using traditional Arabic grammar », The 7th International Conference on Informatics and Systems (INFOS), p. 1‑7, Cairo, Egypt, 2010.

[135] N. Habash et R. Roth, « CATiB: The Columbia Arabic Treebank », Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, p. 221–224, Suntec, Singapour, 2009.

[136] M. Maamouri, A. Bies, S. Kulick, M. Ciul, N. Habash et R. Eskander, « Developing an Egyptian Arabic Treebank: Impact of Dialectal Morphology on Annotation and Tool Development », Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 2014.

[137] D. Taji, N. Habash et D. Zeman, « Universal Dependencies for Arabic », Proceedings of the Third Arabic Natural Language Processing Workshop, p. 166–176, Valencia, Espagne, 2017.

[138] D. Halabi, A. Awajan et E. Fayyoumi, « Arabic LFG-inspired Dependency Treebank », International Conference on New Trends in Computing Sciences (ICTCS), p. 207‑215, Amman, Jordanie, 2017.

[139] R. B. Bahloul, M. Elkarwi, K. Haddar et P. Blache, « Building an Arabic Linguistic Resource from a Treebank: The Case of Property Grammar », Text, Speech and Dialogue, vol. 8655, p. 240‑246, P. Sojka, A. Horák, I. Kopeček, et K. Pala, Cham, Suisse: Springer International Publishing, 2014.

[140] M. Palmer, O Babko-Malaya, A. Bies, M. Diab, M. Maamouri, A. Mansouri et W. Zaghouaniet, « A Pilot Arabic Propbank », Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Maroc, 2008.

[141] T. Buckwalter, Buckwalter Arabic Morphological Analyzer Version 1.0. Web Download. Philadelphia : Linguistic Data Consortium, 2002.

[142] M. Diab et Y. Marton, « Semantic Processing of Semitic Languages », Natural Language Processing of Semitic Languages, p. 129‑159, I. Zitouni, Allemagne: Springer Berlin Heidelberg, 2014.

[143] M. Maamouri, A. Bies et S. Kulick, « Enhanced annotation and parsing of the arabic treebank », 6th International Conference on Computers and Informatics, INFOS2008, Cairo, Egypt, 2008.

[144] W. Zaghouani, M. Diab, A. Mansouri, S. Pradhan et M. Palmer, « The Revised Arabic PropBank », Proceedings of the Fourth Linguistic Annotation Workshop, p. 222–226, Uppsala, Suède, 2010.

[145] J. D. Choi, C. Bonial et M. Palmer, « Propbank Frameset Annotation Guidelines Using a Dedicated Editor, Cornerstone », Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 2010.

[146] J. D. Choi, C. Bonial et M. Palmer, « Propbank Instance Annotation Guidelines Using a Dedicated Editor, Jubilee », Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 2010.

[147] W. Zaghouani, A. Hawwari et M. Diab, « A Pilot PropBank Annotation for Quranic Arabic », Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature, p. 78–83, Montrèal, Canada, 2012.

[148] W. Zaghouani, « Le développement de corpus annotés pour la langue arabe », Thèse de Doctorat, Université Paris Ouest Nanterre La Défense, France, 2015.

[149] J. Mousser, « A Large Coverage Verb Taxonomy for Arabic », Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 2010.

[150] A. Korhonen et T. Briscoe, « Extended Lexical-semantic Classification of English Verbs », Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics, p. 38–45, Stroudsburg, Pennsylvanie, États-Unis, 2004.

[151] S. Elkateb, W. Black, P. Vossen, D. Farwell, C. Fellbaum et A. Pease, « Arabic WordNet and the Challenges of Arabic », Proceedings of Arabic NLP/MT Conference, p. 15‑24, London, Royaume-Uni, 2006.

[152] S. Pradhan, K. Hacioglu, W. Ward, J. H. Martin et D. Jurafsky, « Semantic Role Parsing: Adding Semantic Structure to Unstructured Text », Proceedings of the Third IEEE International Conference on Data Mining, p. 629–632, Melbourne, Florida, États-Unis, 2003.

[153] N. Xue et M. Palmer, « Calibrating Features for Semantic Role Labeling », Proceedings of the Conference on Empirical Methods in Natural Language Processing, p. 88–94, Barcelona, Espagne, 2004.

[154] V. N. Vapnik, Statistical learning theory. New York, États-Unis: John Wiley & Sons, Inc., 1998.

[155] A. Moschitti, « A Study on Convolution Kernels for Shallow Semantic Parsing », Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, p. 335–342, Barcelona, Espagne, 2004.

[156] C. Lo et D. Wu, « Evaluating Machine Translation Utility via Semantic Role Labels », Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, 2010.

[157] D. Wu et P. Fung, « Can Semantic Role Labeling Improve SMT? », Proceedings of the 13th Annual Conference of the EAMT, p. 218‑225, Barcelona, Espagne, 2009.

[158] H. Meguehout, T. Bouhadada et M. T. Laskri, « Semantic Role Labeling for Arabic Language Using Case-based Reasoning Approach », International Journal of Speech Technology, vol. 20, no 2, p. 363–372, Juin 2017.

[159] E. Mathieu-Dupas, « Algorithme des k plus proches voisins pondérés et application en diagnostic », 42èmes Journées de Statistique, Marseille, France, 2010.

[160] B. Campillo-Gimenez, W. Jouini, S. Bayat et M. Cuggia, « Improving Case-Based Reasoning Systems by Combining K-Nearest Neighbour Algorithm with Logistic Regression in the Prediction of Patients' Registration on the Renal Transplant Waiting List », PLOS ONE, vol. 8, no 9, p. 1‑10, Septembre 2013.

[161] R. Morante, W. Daelemans et V. Van Asch, « A Combined Memory-Based Semantic Role Labeler of English », CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning, p. 208–212, Manchester, Royaume-Uni, 2008.

[162] M. Surdeanu, R. Morante et L. Màrquez, « Analysis of Joint Inference Strategies for the Semantic Role Labeling of Spanish and Catalan », Computational Linguistics and Intelligent Text Processing, p. 206‑218, Haifa, Palestine occupée, 2008.

[163] K. Hechenbichler et K. Schliep, Weighted k-Nearest-Neighbor Techniques and Ordinal Classification, vol. 399. Ludwig-Maximilians Université, Munich, Allemagne, 2004.

[164] S. Pradhan et L. Ramshaw, « OntoNotes: Large Scale Multi-Layer, Multi-Lingual, Distributed Annotation », Handbook of Linguistic Annotation, p. 521‑554, N. Ide et J. Pustejovsky, Dordrecht, Pays-Bas: Springer Pays-Bas, 2017.

[165] P. Sameer, A. Moschitti, N. Xue, O. Uryupina et Y. Zhang, « CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes », Joint Conference on EMNLP and CoNLL - Shared Task, p. 1–40, Jeju Island, Corée, 2012.

Webographie

Webographie

[W1]: C'est quoi ?
https://c-est-quoi.fr/fr/definition/lexicographie, (Consulté le 18/06/2018)

[W2]: Université de Princeton, espace WordNet,
https://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html, (Consulté le 19/01/2018)

[W3]: Université de Princeton, Recherche WordNet,
http://wordnetweb.princeton.edu/perl/webwn, (Consulté le 14/04/2018)

[W4]: Université de Californie à Berkeley, Espace FrameNet,
https://framenet.icsi.berkeley.edu/fndrupal/current_status, (Consulté le 15/04/2018)

[W5]: Université de Californie à Berkeley, Espace FrameNet,
https://framenet.icsi.berkeley.edu/fndrupal/about, (Consulté le 20/01/2018)

[W6]: Université de Californie à Berkeley, Index FrameNet,
https://framenet2.icsi.berkeley.edu/fnReports/data/frameIndex.xml?frame=Text_creation,
(Consulté le 20/01/2018)

[W7]: Université de Californie à Berkeley, Recherche FrameNet,
https://framenet.icsi.berkeley.edu/fndrupal/framenet_search, (Consulté le 20/01/2018)

[W8]: Université du Colorado, Espace VerbNet,
https://verbs.colorado.edu/verbnet/, (Consulté le 22/01/2018)

[W9]: Projet PropBank dans GitHub,
http://propbank.github.io/, (Consulté le 24/01/2018)

[W10]: Livre sur Google book, ملاحظات على قواعد النحاة
https://books.google.dz/books?id=wyD2DQAAQBAJ&pg=PA35&lpg=PA35&dq=pro-drop+language+%D8%A7%D9%84%D8%B9%D8%B1%D8%A8%D9%8A%D8%A9&source=bl&ots=eAHF65svS4&sig=RdOP5mZtfrKsfMO42b01nh4_8_8&hl=fr&sa=X&ved=0ahUKEwjJ0ayUn8nZAhUGUhQKHavJD9c4ChDoAQgwMAE#v=onepage&q=pro-drop%20language%20%D8%A7%D9%84%D8%B9%D8%B1%D8%A8%D9%8A%D8%A9&f=false , (Consulté le 26/04/2018)

[W11]: OpenClassrooms, Apprenez à programmer en Python
https://openclassrooms.com/courses/apprenez-a-programmer-en-python/qu-est-ce-que-python, (Consulté le 20/05/2018)

[W12]: Apprendre le langage de programmation python,
http://apprendre-python.com/, (Consulté le 20/05/2018)

[W13]: Wikipédia l'encyclopédie libre, Python (langage),
https://fr.wikipedia.org/wiki/Python_(langage), (Consulté le 20/05/2018)

[W14]: IEEE spectrum, The 2017 Top Programming Languages,

https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages, (Consulté le 20/05/2018)

[W15]: IEEE spectrum, https://spectrum.ieee.org/ns/IEEE_TPL_2017/methods.html, (Consulté le 20/05/2018)

[W16]: Club des développeurs et IT pro, https://www.developpez.com/actu/150166/IEEE-Python-devient-le-meilleur-langage-en-2017-en-depassant-C-et-Java-decouvrez-le-classement-complet-selon-divers-criteres/, (Consulté le 21/05/2018)

[W17]: IEEE spectrum, Interactive: The Top Programming Languages 2017, https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2017, (Consulté le 20/05/2018)

[W18]: Makina Corpus, Présentation de l'écosystème Python scientifique, https://makina-corpus.com/blog/metier/2017/presentation-de-lecosysteme-python-scientifique#introduction, (Consulté le 21/05/2018)

[W19]: Makina Corpus, Initiation au Machine Learning avec Python - La théorie, https://makina-corpus.com/blog/metier/2017/initiation-au-machine-learning-avec-python-theorie, (Consulté le 21/05/2018)

[W20]: Makina Corpus, Initiation au Machine Learning avec Python - La pratique, https://makina-corpus.com/blog/metier/2017/initiation-au-machine-learning-avec-python-pratique, (Consulté le 21/05/2018)

[W21]: NLTK 3.3 documentation, Natural Language Toolkit, http://www.nltk.org/, (Consulté le 21/05/2018)

[W22]: Université de Pennsylvanie, Linguistic Data Consortium, OntoNotes Release 5.0, https://catalog.ldc.upenn.edu/ldc2013t19, (Consulté le 21/05/2018)

[W23]: CoNLL-2012 Shared Task, Data, http://conll.cemantix.org/2012/data.html, (Consulté le 01/06/2018)

Annexes

Annexes

Annexes

Annexe A : Rôles sémantiques dans CoNLL-2012

Tableau A.1. Description des rôles sémantiques dans CoNLL-2012

	Argument	Description
Arguments numérotés	ARG0	Agent
	ARG1	Patient
	ARG2	Instrument, Bénéficiaire, Attribut
	ARG3	Point de départ
	ARG4	Point final
Arguments modifier	ARGM-ADV	Adverbials
	ARGM-CAU	Cause
	ARGM-COM	Comitative
	ARGM-EXT	Ampleur
	ARGM-GOL	But
	ARGM-LOC	Locative
	ARGM-MNR	Manière
	ARGM-NEG	Negation
	ARGM-PRP	Purpose
	ARGM-TMP	Temporel
	ARGM-ADV(C-ARG1	/
	ARGM-ADV(C-ARG2	/
Autres arguments	C-ARG0, C-ARG1, C-ARG2, C-ARGM-ADV, C-ARGM-LOC, C-ARGM- PRP, C-ARGM-TMP	ARG argument dans la phrase et C-ARG vient après lui
	R-ARG0, R-ARG1, R-ARG2, R-ARGM-LOC, R-ARGM-TMP	R-ARG est une référence qui vient avant ou après ARG

Annexe B : Quelques Script Python

B.1- srl_system.py

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-
import glob
import os.path
import k_nn
import similarity
import write
import Liste_knn_csv

""" ----------------------------------------MAIN SYSTEM-------------------------------------------------
                    Call the other scripts
"""

list_path_tgt = \
    glob.glob('C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\tgt_base\\conll_2012_test\\*\\*'
              )# targets cases directory

# Extract a line from the base of target cases-----------------------------------------------------
number_case = 0

for path_tgt in list_path_tgt:
    open_tgt_file = open(path_tgt, 'r')
    tgt_file_lines = open_tgt_file.readlines()

  # Read a line of the target case-------------------------------------------------------------------

    for tgt_line in tgt_file_lines:
        if tgt_line != '\n':

            k = similarity.main(tgt_line) # Call similarity

            liste_knn = Liste_knn_csv.all(k) #creat liste knn CSV all
            k = k[:1]

            sem_role = k_nn.standard_weight_classe(k) # Call k_nn

            write.revise(path_tgt, tgt_line, sem_role[1], liste_knn, number_case) # Call write revise

            write.all_tgt(path_tgt, tgt_line, sem_role[1]) # Call write.tgt

        elif tgt_line == '\n':
            write.return_to_line(path_tgt, tgt_line) # Call write.return_to_line

        else:
            print ('Erreur dans cible_line')

        number_case += 1
```

B.2- similarity.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
""" ----------------------------------------------SCRIPT SIMILARITY-----------------------------------------------
                To calculate the similarity between the target case and other source cases
"""
import glob
import os.path

def main(tgt_line):
  tgt_line = tgt_line.strip("\n")
  tgt_line = tgt_line.split("      ")
  list_path_srce = \
    glob.glob('C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\srce_base\\*\\*\\*'
              )  # Sources cases directory

  k_nn = []

  # Extract a line from the base of sources cases--------------------
  for path_srce in list_path_srce:
    open_srce_file = open(path_srce, 'r')
    srce_file_lines = open_srce_file.readlines()

    # Read a source line-----------------------------------------------
    for srce_line in srce_file_lines:
      if srce_line != ('\n'):
        srce_line = srce_line.strip("\n")
        srce_line = srce_line.split("      ")

        sim_k = calculate_sim(tgt_line, srce_line)

        if sim_k[0] != 0:
          k_nn.append(sim_k)

  k_nn = sorted(k_nn, reverse=True)[:15] # ************** K for CSV file **************

  for k in k_nn:
    k[0] = 1./k[0]

  return k_nn

```

```python
# ##################### To calculate the similarity between the target case and the source case ####################
def calculate_sim(tgt_line, srce_line):

    i = 0
    sim_k = [0, "X", srce_line]

    while i in range(len(tgt_line)):
        if tgt_line[i] == srce_line[i]:

            if i == 1:
                sim_k[0] += 6

            elif i == 3:
                sim_k[0] += 3

            elif i == 4:
                sim_k[0] += 5

            elif i == 6:
                sim_k[0] += 3

            elif i == 7:
                sim_k[0] += 2

            elif i == 8:
                sim_k[0] += 2

            else:
                sim_k[0] += 1

        i += 1

    sim_k[1] = srce_line[9]
    return sim_k
```

B.3- K_nn.py

```
""" ----------------------------------------SCRIPT K_NN----------------------------------------
"""
# ################################### Standardization of values ###################################
def standard_weight_classe(k):

    sem_role = [99, "ARGX","Line"]

    k_nn_final = k

    sem_classe = accumulate_sim(k_nn_final) # Script accumulate_sim

    sem_role = classe_max(sem_classe) # Script classe_max

    return sem_role

# ############################## Cumulate the similarity of each class ##############################
def accumulate_sim(k_nn_final):
    sem_classe = []
    sem_classe.insert(0, k_nn_final[0][:]) # Insert element 0 of k_nn_final in sem_classe

    # 1 to last element in k_nn_final--------------------------------
    for i in range (1, len(k_nn_final)):
        j = 0
        boolean = False

        while boolean == False:
            if sem_classe[j][1] == k_nn_final[i][1]:
                sem_classe[j][0] += k_nn_final[i][0]
                boolean = True

            elif j == len(sem_classe)-1:
                sem_classe.append(k_nn_final[i][:])
                boolean = True

            j += 1

    return sem_classe

# ################################### Choose the best class ###################################
def classe_max(sem_classe):
    sem_role = sem_classe[0]

    for classe in sem_classe:
        if classe[0] > sem_role[0]:
            sem_role = classe

    return sem_role
```

B.4- Liste_knn_csv.py

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-
""" ----------------------------------------------SCRIPT CSV-----------------------------------------------
"""
import glob
import os.path

def all(K):

    #-------------Creat CSV files-----------------------------------------
    path_csv_knn = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\liste_knn_all.csv")
    #path_csv_test = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\dataset_01_test_3.0.csv")

    csv_file_knn = open(path_csv_knn, 'a')
    #csv_file_test = open(path_csv_test, 'a')

    #-------------Write K cases in CSV file-------------------------------------------
    for i in range (len(K)):
        case = K[i][2]
        j = 0
        for elem in case:
            if j != len(case)-1:
                csv_file_knn.write(elem+' ')
            else:
                csv_file_knn.write(elem+'\n')
            j += 1

    liste_knn = K
    return liste_knn

    return
```

B.5- write.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
""" ----------------------------------------------SCRIPT WRITE-----------------------------------------------
"""
import glob
import os.path

def revise (path_tgt, tgt_line, role, liste_knn, number_case):
    # Creat CSV files for liste tgt case with error
    path_csv_tgt_error = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\tgt_error.csv")
    csv_file_tgt_error = open(path_csv_tgt_error, 'a')

    # Creat CSV liste Knn of Tgt with error
    path_csv_knn_error = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\knn_tgt_errors.csv")
    csv_file_knn_error = open(path_csv_knn_error, 'a')

    # Information from path_tgt
    (path, name_file) = os.path.split(path_tgt)
    foldar = path.split('\\')[7]

    # Creation of revised path and files
    path_revised = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\srce_base\\revised\\", foldar)
    if not os.path.exists(path_revised):
        os.makedirs(path_revised)
    path_revised = os.path.join(path_revised, name_file)
    revised_file = open(path_revised, 'a')

    # Creation path of Role with error is special
    path_role_error_special = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\role_error_special\\", foldar)
    if not os.path.exists(path_role_error_special):
        os.makedirs(path_role_error_special)
    path_role_error_special = os.path.join(path_role_error_special, name_file)
    role_error_special_file = open(path_role_error_special, 'a')

    # Tgt line
    line = tgt_line.strip("\n")
    line = line.split("      ")

    # Verification of role : (correct) or (not correct)
    correct = glob.glob("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\tgt_base_semantic_roles\\conll_2012_test\\*\\*")
    i = 0
    boolean = False
    number = 0
```

```
    while i < len(correct) and boolean == False:

        file_correct = open(correct[i])
        lines_correct = file_correct.readlines()
        (path_correct, name_file_correct) = os.path.split(correct[i])

        j = 0
        while j < len(lines_correct) and boolean == False:
            if lines_correct[j] != ('\n'):
                lines_correct[j] = lines_correct[j].strip('\n')
                l_correct = lines_correct[j].split("      ")
                if line[:9] == l_correct[:9] and number == number_case:
                    # Correct role
                    if role == l_correct[9]:
                        correct_line = tgt_line.strip('\n') +"      "+ role + '\n'

                        # Save in case base, path revised
                        revised_file.write(correct_line)

                        # Save correct roles XXX
                        role_error_special_file.write(correct_line)
                        boolean = True

                    # Not correct
                    else:
                        # Save in case base, path revised
                        corrected_line = tgt_line.strip('\n') +"      "+ l_correct[9] + '\n'
                        revised_file.write(corrected_line)

                        # Save not correct in roles Error
                        error_line = tgt_line.strip('\n') +"      "+ "Error" + '\n'
                        role_error_special_file.write(error_line)

                        # Save Tgt CSV in liste case tgt with error
                        for i_line in range (len(line)):
                            if i_line != len(line)-1:
                                csv_file_tgt_error.write(line[i_line]+' ')
                            else:
                                csv_file_tgt_error.write(line[i_line]+' ')
                                csv_file_tgt_error.write(l_correct[9]+'\n')

                        # Save liste Knn of Tgt with error
                        for i_knn in range (len(liste_knn)):
                            case = liste_knn[i_knn][2]
                            j_knn = 0
                            for elem in case:
                                if j_knn != len(case)-1:
                                    csv_file_knn_error.write(elem+' ')
                                else:
                                    csv_file_knn_error.write(elem+'\n')
                                    j_knn += 1
                        boolean = True
                j += 1
                number += 1
            i += 1
    return
```

```
103  def all_tgt(path_tgt, tgt_line, role):
104      (path, name_file) = os.path.split(path_tgt)
105      foldar = path.split('\\')[7]
106      path_output = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\output\\", foldar)
107
108      if not os.path.exists(path_output):
109          os.makedirs(path_output)
110
111      path_output = os.path.join(path_output, name_file)
112      output_file = open(path_output, 'a')
113
114      tgt_line = tgt_line.strip('\n') +"      "+ role + '\n'
115      output_file.write(tgt_line)
116
117      return
118
119  def return_to_line(path_tgt, tgt_line):
120      (path, name_file) = os.path.split(path_tgt)
121      foldar = path.split('\\')[7]
122      path_output = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\output\\", foldar)
123
124      if not os.path.exists(path_output):
125          os.makedirs(path_output)
126
127      path_output = os.path.join(path_output, name_file)
128      output_file = open(path_output, 'a')
129      output_file.write(tgt_line)
130
131      # Creation of revised path and files
132      path_revised = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\srce_base\\revised\\", foldar)
133      if not os.path.exists(path_revised):
134          os.makedirs(path_revised)
135      path_revised = os.path.join(path_revised, name_file)
136      revised_file = open(path_revised, 'a')
137      revised_file.write("\n")
138
139      # Creation path of Role with error is special
140      path_role_error_special = os.path.join("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\role_error_special\\", foldar)
141      if not os.path.exists(path_role_error_special):
142          os.makedirs(path_role_error_special)
143      path_role_error_special = os.path.join(path_role_error_special, name_file)
144      role_error_special_file = open(path_role_error_special, 'a')
145      role_error_special_file.write("\n")
146
147      return
```

B.6- testing.py

```
"""-----------------------------------------SCRIPT TESTING-----------------------------------------
"""
from __future__ import division

import glob
import os.path

# ################################### List of roles ###################################
def classe (liste_correct):
    sem_classe = []
    sem_classe.insert(0, liste_correct[0][:]) # insert element 0 of liste_correct in sem_classe

    # 1 to last element in liste_correct
    for i in range (1, len(liste_correct)):
        j = 0
        boolean = False

        while boolean == False:
            if sem_classe[j][0] == liste_correct[i][0]:
                sem_classe[j][1] += liste_correct[i][1]
                boolean = True

            elif j == len(sem_classe)-1:
                sem_classe.append(liste_correct[i][:])
                boolean = True

            j += 1

    return sem_classe
# ###########################################################################################
```

```
i = 0
j = 0
role_correct = 0
role_in_correct = 0
liste_correct = []
liste_incorrect = []

test = glob.glob("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\tgt_base_semantic_roles\\conll_2012_test\\*\\*")
output = glob.glob ("C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\output\\*\\*")

while (i < len(output)) and (j < len(test)):
    i2 = 0
    j2 = 0

    file_output = open(output[i])
    lines_output = file_output.readlines()
    (path, name_file_output) = os.path.split(output[i])

    file_test = open(test[j])
    lines_test = file_test.readlines()
    (path, name_file_test) = os.path.split(test[j])

    while i2 < len(lines_output) and (j2 < len(lines_test)):
        if lines_output[i2] != ('\n') and lines_test[j2] != ('\n'):
            l_output = lines_output[i2].split("      ")
            l_test = lines_test[j2].split("      ")
            if l_output[:9] == l_test[:9]:
                l_output[9] = l_output[9].strip('\n')
                l_test[9] = l_test[9].strip('\n')

                if l_output[9] == l_test[9]:

                    role_correct += 1

                    role = [l_output[9], 1]
                    liste_correct.append(role)

                else:
                    role_in_correct += 1
                    role = [l_test[9], 1]
                    liste_incorrect.append(role)
            else:
                print ("sentences not equal----------------------------------")
                print ("name file output : ", name_file_output,"---",l_output)
                print ("name file test : ", name_file_test, "---",l_test,"\n")

        i2 += 1
        j2 += 1
    i += 1
    j += 1

print ("Role correct = ", role_correct)
print ("Liste = ", classe(liste_correct), ('\n'))

print ("Role incorrect = ", role_in_correct)
print ("Liste = ", classe(liste_incorrect), ('\n'))

precision = (role_correct/(role_correct+role_in_correct))*100
print ("Precision = ",precision, "\n")
```

B.7- Modèle Deep Learning

```
# Artificial Neural Network
# Sources :
    # https://www.udemy.com/machine-learning-arabic/
    # Help in Facebook goupes

# In[######## Part 1 - Data Preprocessing ########]

# In[Importing the libraries]
import numpy as np
import pandas as pd

# In[Importing the dataset]
dataset = pd.read_csv('C:\\Users\\Hamza\\Desktop\\Data PhD SRL System v5.0\\DeepLearning_Dataset 4.0.csv',delimiter=' ', header = None)
X = dataset.iloc[:, [1,2,4,5,6,7,8]].values
y = dataset.iloc[:, 9].values

# In[Encoding categorical data (the Independent Variable)]
from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder_X_0 = LabelEncoder()
X[:, 0] = labelencoder_X_0.fit_transform(X[:, 0])

labelencoder_X_1 = LabelEncoder()
X[:, 1] = labelencoder_X_1.fit_transform(X[:, 1])

labelencoder_X_2 = LabelEncoder()
X[:, 2] = labelencoder_X_2.fit_transform(X[:, 2])

labelencoder_X_3 = LabelEncoder()
X[:, 3] = labelencoder_X_3.fit_transform(X[:, 3])

labelencoder_X_4 = LabelEncoder()
X[:, 4] = labelencoder_X_4.fit_transform(X[:, 4])

labelencoder_X_5 = LabelEncoder()
X[:, 5] = labelencoder_X_5.fit_transform(X[:, 5])

labelencoder_X_6 = LabelEncoder()
X[:, 6] = labelencoder_X_6.fit_transform(X[:, 6])


onehotencoder = OneHotEncoder(categorical_features = [0,1,2,3,4,5,6])
X = onehotencoder.fit_transform(X).toarray()

# In[Encoding categorical data (Encoding the Dependent Variable) ]
labelencoder_y = LabelEncoder()
y = labelencoder_y.fit_transform(y)
```

```
# In[Splitting Small Data the dataset into the Training set and Test set]
X_train = X[:79365,:]
y_train = y[:79365]
X_test = X[79365:,:]
y_test = y[79365:]

# In[######## Part 2 - ANN ########]
# In[Importing the Keras libraries and packages]
from keras.models import Sequential
from keras.layers import Dense

# Initialising the ANN
classifier = Sequential()

# Adding the input layer and the first hidden layer
classifier.add(Dense(output_dim = 120, init = 'uniform', activation = 'relu', input_dim = 5429))
# Adding the second hidden layer
classifier.add(Dense(output_dim = 60, init = 'uniform', activation = 'relu'))
# Adding the output layer
classifier.add(Dense(output_dim = 29, init = 'uniform', activation = 'sigmoid'))

# Compiling the ANN
classifier.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])

# In[Normal learning]
# Fitting the ANN to the Training set
classifier.fit(X_train, y_train, batch_size = 30, nb_epoch = 10)
# Making the predictions and evaluating the model
y_pred = classifier.predict(X_test)
# real result into y_pred2
y_pred_all = np.argmax(y_pred, axis=1)

# In[Making the Confusion Matrix]
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred_all)

# In[Give the success percentage]
All_cases = 0
Just_cases = 0
# All cases
for i in range(len(cm)):
    for j in range(len(cm)):
        All_cases = All_cases + cm[i,j]
# Just cases
for i in range(len(cm)):
    Just_cases = Just_cases + cm[i,i]

print ('All cases = ',All_cases)
print ('Just cases = ',Just_cases)
print ('Precision = ',(Just_cases/All_cases)*100)
```

Annexe C : Construction de données CoNLL

1. Nous avons utilisé la version "14.04.2 / 64 bits" du système d'exploitation "Ubuntu" pour faire fonctionner le script "skeleton2conll.sh" contenu dans le fichier compressé numéro 7 "conll-2012-scripts.v3.tar".

2. Il est nécessaire de construire deux dossiers pour faire fonctionner le script. Le premier dossier sous le nom de "`ontonotes-release-5.0`" qui contient la ressource OntoNotes 5.0. Le deuxième sous le nom de "`conll-2012`" et contiens le contenu des sept fichiers compressés.

3. Vu la large utilisation du système Windows dans nos ordinateurs, il est possible de faire une mémoire externe (clé USB) avec un système d'exploitation Ubuntu. Dans notre cas, nous avons utilisé "`LinuxLive USB Creator 2.9.4`" pour construire une clé USB avec un système d'exploitation Ubuntu.

4. Booter l'ordinateur sur cette clé USB.

5. Dans le terminal d'Ubuntu, il faut exécuter les instructions suivantes (le symbole Underscore ' _ ' symbolise un espace et ne pas écrire les symboles crochet gauche ' [' et crochet droite '] ') :

   - Sortir du répertorie en cours → **`cd_/`**
   - Entrer dans le répertoire du script →**`CD_Répertoire du scripte`**
   - Faire exécuter le script → **`bash_skeleton2conll.sh_-D_`**[…**`/ontonotes-release-5.0/data/files/data`**]_[…**`/conll-2012`**]