

TAGHIT: Un étiqueteur Morphosyntaxique de l'Arabe Basé sur une Approche Multicritère

CHERAGUI Mohamed Amine

Ecole nationale Supérieure d'informatique (ESI)

HOCEINI Youssef
Université de Bechar

ABBAS Moncef
USTHB

Résumé:

Les applications traitant automatiquement les langues naturel ont atteint ces dernières décennies un certain seuil de fiabilité et de performance, cela est due à deux (02) facteurs : d'une part à l'effort déployé par les chercheurs pour faire progresser cette discipline et d'une autre part au développement technologique qu'a subi l'informatique. Mais malgré cette ascension le champ du traitement automatique du langage naturel reste toujours confronté à un certain nombre de difficultés. L'un de ces problèmes qui persiste encore et toujours est l'ambiguïté que soit : morphologique, syntaxique et sémantique. Cette ambiguïté est due généralement à la complexité de la langue traitée et cela constitue une grande partie des travaux réalisés en T.A.L.N.

Nous essayons à travers cet article, de décrire le phénomène de l'ambiguïté morphologique dans le traitement automatique du texte arabe et de présenter notre étiqueteur morphosyntaxique TAGHIT capable de lever cette forme d'ambiguïté. Ce système qui sera doté d'une nouvelle approche de désambiguïsation morphologique différente de celles qui existent actuellement, comme l'approche par contraintes ou l'approche stochastique (probabiliste/statistique). L'originalité de notre travail réside dans le fait de s'appuyer sur un formalisme mathématique solide qui est l'aide multicritère à la décision.

1. Introduction

Depuis l'apparition de l'intelligence artificielle au milieu des années 50 (lors de la conférence de Dartmouth), plusieurs chercheurs ont eue l'idée de rendre la machine capable de comprendre un énoncé en langage naturel; donnant ainsi naissance à une nouvelle branche de recherche baptisée T.A.L.N¹, qui veut dire traitement automatique du langage naturel.

Le champ du traitement automatique des langues dites naturelles est devenu avec le temps et le développement

¹ TALN : Traitement Automatique du Langage Naturel.

technologique qu'a subi l'informatique (en terme de vitesse de traitement et support de stockage); un domaine scientifique en pleine effervescence ; générant un nombre important d'applications, telle que: la traduction automatique, correction automatique des erreurs, analyse de texte, génération automatique de résumés, extraction des connaissances, interrogation de bases de données en langage naturel, ...etc. Réalisé de telles applications nécessite généralement une étape d'analyse du texte source, qui se fait à son tour par décomposition de cette analyse en sous tâches calquées sur les différents niveaux de traitement linguistique, que se soit : morphologique, syntaxique, sémantique et pragmatique. Ainsi, il est clair que d'un point de vue linguistique, ces tâches ne sont pas indépendantes les unes des autres et qu'à chaque niveau, les connaissances des autre niveaux sont nécessaires.

Avec l'internationalisation du web et la prolifération des applications visant à automatiser les langues naturelles. Ce développement ne cesse de s'accroître ; abordant des problématiques de plus en plus complexes comme celles de la structuration des connaissances linguistiques et l'ambiguïté.

C'est ce dernier problème qui nous intéresse à savoir l'ambiguïté dans le traitement automatique de la langue arabe. L'ambiguïté est comparée dans la littérature à un état de confusion, cet embrouillement se manifeste sous différents formes et selon les différents niveaux de traitement cité auparavant.

Ainsi nous essayons à travers cet article à la fois de présenter l'une des formes d'ambiguïtés la plus persistante en T.A.L.A.², à savoir l'ambiguïté morphologique tout en présentant une nouvelle génération d'étiqueteurs morphosyntaxiques capable de lever cette forme d'ambiguïté, où l'originalité de ce travail

² TALA : Traitement Automatique de la Langue Arabe.

réside dans l'adoption comme approche de désambiguïsation, les principes et méthodes issue de l'aide multicritère à la décision.

2. La langue arabe : origine et caractéristiques

La langue arabe, comme l'Akkadien, le Syriaque et l'Hébreu, fait partie de la famille des langues sémitiques⁶ dont les premières inscriptions datent de l'année 512 de notre ère sur une stèle trilingue près d'Alep en Syrie. L'arabe comme toutes les autres langues du monde, présente des caractéristiques particulières. Parmi celles-ci on peut évoquer (Benhamouda, 93)

- L'alphabet arabe est composé de 29 caractères, considérées tous comme des consonnes ;
- Une lettre peut changer de forme selon sa position ou selon son contexte phonétique ;
- Les mots de la langue arabe s'écrivent de droite à gauche ;
- La langue arabe est semi cursive³ ;
- Le verbe n'a que trois (03) aspects concernant le temps : l'accompli « الماضي », l'inaccompli « المضارع » et l'impératif « الأمر » ;
- La dérivation s'obtient soit par l'ajout de lettres additionnelles telles que : les affixes⁴ et les clitiques⁷ ;
- Une flexibilité syntaxique.

3. Ambiguïté Morphologique dans T.A.L.A

Le traitement morphologique porte sur l'unité élémentaire identifiable, en l'occurrence le morphème, à ce niveau on s'intéresse à deux (02) concepts essentiels qui sont :

³ Dans la langue arabe on peut trouver des mots qui contiennent des lettres séparées.
(Exemple : « دحرج »).

⁴ Le terme affixe regroupe les préfixes et les suffixes.

- *la synthèse* qui permet de générer des mots ou des phrases par le biais d'un ensemble de règles de dérivation, de flexions et d'adaptations.
- *l'analyse* qui a pour rôle d'associer à un mot graphique un ensemble d'informations décrivant les unités morphologiques et grammaticales entrant dans sa composition (proclitiques, préfixes, base, suffixes, enclitique) (Tuerlinckx, XX). C'est à ce stade, que l'ambiguïté morphologique se manifeste ; lorsque l'analyse assigne à une unité lexicale plusieurs informations (ou l'inverse) ce qui génère la notion de combinatoire (Vergne et Giguet, 98).

4. Etiquetage Morphosyntaxique comme Solution au Problème de L'ambiguïté Morphologique

Pour remédier au problème de l'ambiguïté morphologique, on utilise un Taggeur ou un étiquetage morphosyntaxique en français, le rôle d'un tel système est d'assigner une étiquette ou un tag à un mot. Le tag contient des informations morphosyntaxiques sur le mot, c'est-à-dire des informations concernant la forme et la fonction du mot. Elles comprennent notamment la catégorie grammaticale, le genre, le nombre, le temps et le mode (Lechel, 05).

Pour mettre en place un étiqueteur morphosyntaxique de la langue arabe, on doit construire trois (03) modules, respectivement:

4.1. Module de Segmentation: lorsqu'on évoque la segmentation dans le traitement automatique des langues dites naturelles, on parle le plus souvent de trois (03) niveaux de segmentations :

- Segmentation au niveau du texte ;
- Segmentation au niveau de la phrase ;
- Segmentation au niveau du mot.

4.2. *Module d'analyse morphologique*: Le but principal de ce type d'analyse est de vérifier l'appartenance d'un mot donnée au domaine linguistique choisi et de pouvoir disposer ainsi de tous les renseignements le concernant pouvant servir à l'analyse syntaxique.

4.3. *Module de désambiguïsation*: La désambiguïsation est une étape cruciale dans le processus d'étiquetage morphosyntaxique, à ce niveau du traitement si un mot est mal étiqueté, les règles de la grammaire s'appliqueront mal ou pas du tout. Cependant la phase de désambiguïsation, n'est pas toujours nécessaire ou obligatoire au bon déroulement du processus d'étiquetage. Il faut dire que le module de désambiguïsation rentre en jeu dans un seul cas de figure, celui où l'unité lexicale (mot) reçoit plus d'une étiquette (plus d'une information morphosyntaxique), ce qui va générer une situation de confusion ou ambiguïté.

La plupart des étiqueteurs qui existent aujourd'hui sont classés selon leur mode de désambiguïsation. Ainsi dans la littérature, les approches de désambiguïsation, se répartissent en deux (02) catégories et chaque catégorie englobe une ou plusieurs techniques pour lever l'ambiguïté morphologique. La figure 1 ci-après donne une indication sur les différentes approches et les techniques qui vont avec :

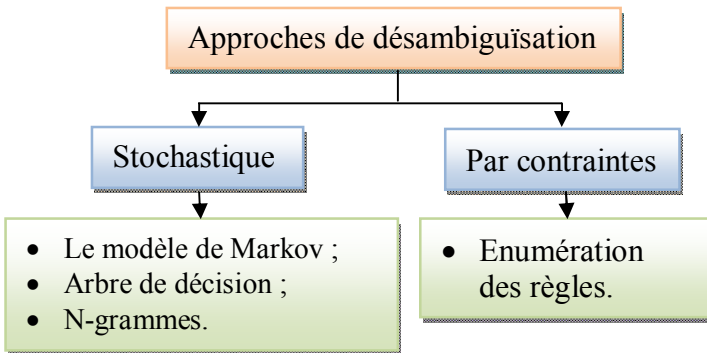


Figure 1: Approche et techniques de désambiguïsation

4.3.1. *Approche par contraintes:* L'approche de désambiguïsation par contraintes (ou par règles) est la méthode la plus ancienne qui a été mise en place pour remédier au problème de l'ambiguïté morphologique. Cette approche se base principalement sur l'intervention d'un linguiste (ou un grammairien) afin d'établir une liste de règles permettant de lever l'ambiguïté. Ces règles sont généralement classées en trois (03) catégories de règles (Alloti et Ponsard, 05), qui sont :

- Les règles contextuelles ou de principes ;
- Les heuristiques ;
- Les règles non contextuelles.

Parmi les étiqueteurs qui adoptent l'approche par contraintes on peut citer: Freeman's Tagger (Al-Sulaiti, 04).

4.3.2. *Approche stochastique:* Lever l'ambiguïté morphologique en adoptant une approche stochastique, cela consiste essentiellement à définir deux (02) sortes d'informations, la première est sur le mot à étiqueter (i.e. l'association entre le mot et l'étiquette) et une seconde information qui est contextuelle syntaxique (i.e. la possibilité de déterminer la probabilité d'avoir une étiquette « i » quant

elle est précédée de l'étiquette « j » dans le texte). En plus de ces deux (02) hypothèses qui sont considérées comme étant les formules de calcul, une phase d'apprentissage est obligatoire, et cela, en entraînant le module de désambiguïsation sur un corpus généralement annoté (i.e. étiqueter à la main) au préalable.

Parmi les étiqueteurs qui adoptent l'approche par contrainte on peut citer: « TnT : Trigrams 'n' Tags. » qui utilise le modèle caché de Markov (Constant, 07).

Remarque:

- Plusieurs travaux dans le domaine ont démontrés que l'étiqueteur par contraintes (ou par règles) est plus rapide et facile à mettre en place que l'étiqueteur stochastique. Il est aussi plus fiable et performant en termes d'étiquetage ;
- On peut trouver une troisième classe d'étiqueteur, qui combine entre les deux (02) approches pour augmenter la performance de ce dernier. Parmi les étiqueteurs qui associent l'approche par contrainte et l'approche stochastique, on peut citer : « MBT : Memory Based Tagger ».

5. Présentation du système TAGHIT

Nous présentons dans cette section les grands axes de conception de notre étiqueteur morphosyntaxique, que nous avons surnommé TAGHIT⁵. La figure 2 ci-dessous représente l'architecture générale de notre système.

⁵ TAGHIT : TAGger morphosyntaxique InformaTisé.

TAGHIT: Un étiqueteur Morphosyntaxique de l'Arabe
Basé sur une Approche Multicritère

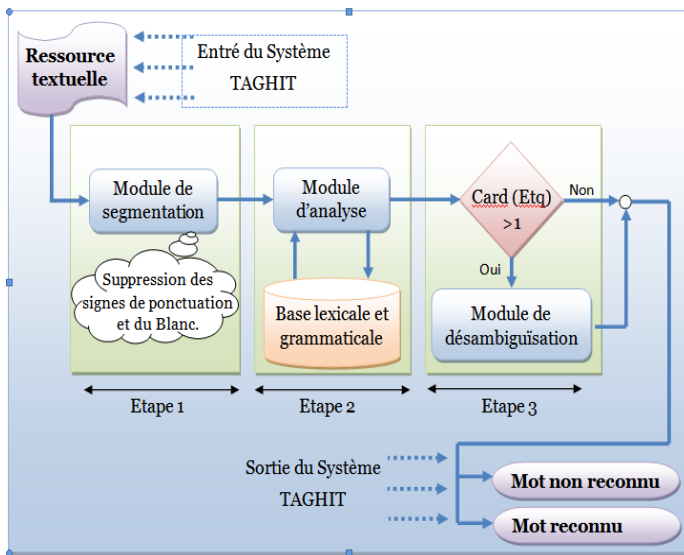


Figure 2: Architecture générale du système TAGHIT.

Comme en peut le constater sur le schéma, le système TAGHIT est subdivisé en trois (03) modules:

5.1. *Module de segmentation* : Ce module est responsable de du découpage du texte en phrases en se basant sur les signes de ponctuation et la segmentation de la phrase en mots en éliminant le blanc (voir figure 3).

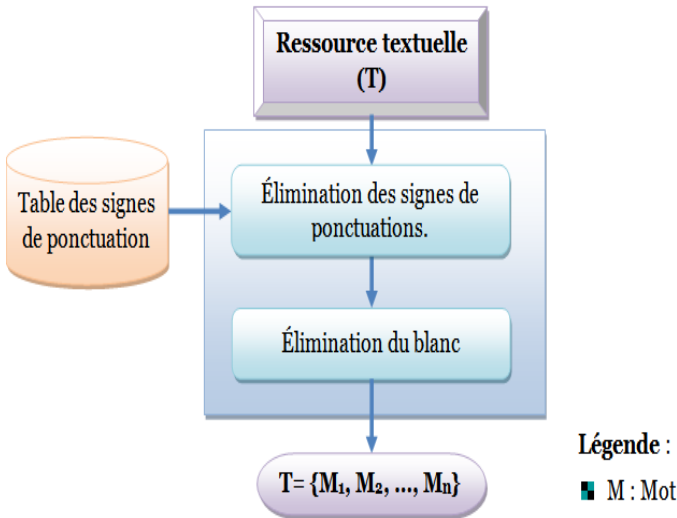


Figure 3. Architecture générale du module de segmentation du système TAGHIT.

Remarque

Concernant le troisième niveau de segmentation qui est la segmentation au sein du mot, nous l'avons introduite avec le module d'analyse morphologique. Pour la bonne raison d'avoir un étiquetage plus précis concernant les composants du mot (dans le cas où ce dernier est complexe).

5.2. *Le module d'analyse morphologique* : Pour mettre ce module d'analyse morphologique, nous avons eue l'idée de créer deux (02) parties complémentaires, à savoir

5.3. (Voir figure 4):

TAGHIT: Un étiqueteur Morphosyntaxique de l'Arabe
Basé sur une Approche Multicritère

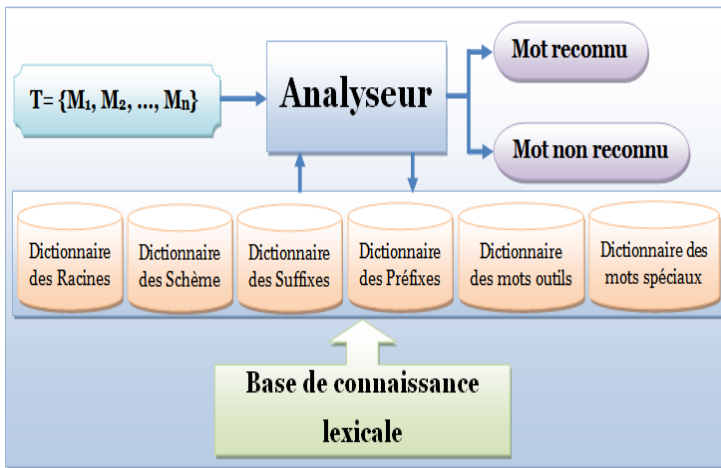


Figure 4: Architecture générale du module d'analyse morphologique du système TAGHIT.

5.3.1. *Base de connaissance lexicale* : cela se traduit par une base de données qui contient les unités lexicales de la langue arabe, ainsi que les informations morphologiques d'où l'appellation base de connaissance.

Remarque

Vue le lexique très volumineux que présente la langue arabe nous avons opté à ne représenté dans la base que les éléments lexicaux élémentaires complétés par des règles lexicales qui permettent d'analyser l'ensemble des mots de la langue. Ainsi le lexique sera composé de essentiellement de: Racines, schème générique (verbe et nom), des articles (préposition, conjonction, ...etc.), affixes et les mots spéciaux (noms propres et noms empruntés).

5.3.2. *Analyseur* : le rôle de cet analyseur est faire le lien entre l'unité lexicale et la base de connaissance, afin d'attribuer à cette dernière l'information convenable.

Remarque

Pour chaque lexique il existe un analyseur de reconnaissance. La figure 5 présente l'architecture générale de l'analyseur.

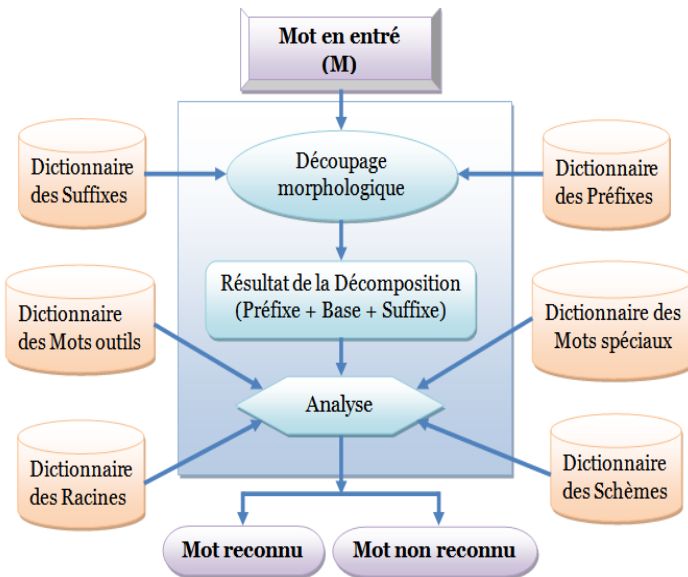


Figure 5: Architecture générale de l'analyseur.

5.4. *Le module de désambiguïsation* : C'est à ce stade que notre contribution va apparaître, par le fait de proposer une nouvelle approche de désambiguïsation différente de celles qui existent actuellement, comme l'approche stochastique et l'approche par contrainte. Cette nouvelle approche dédiée à la lever de l'ambiguïté morphologique s'appuie essentiellement sur les principes de la théorie décisionnelle et les méthodes issus de l'aide multicritère à la décision.

5.4.1. *Justification du choix d'une démarche multicritère :*
L'intérêt d'adopter une approche d'aide multicritère à la décision pour lever l'ambiguïté morphologique dans le T.A.L.A., peut être résumé en deux (02) points, qui sont:

- Réduction de l'ensemble des étiquettes candidates: elle permet de réduire d'emblée le nombre d'étiquettes de correction, en éliminant ceux dominées générant ainsi l'ensemble des étiquette efficaces ;
- Classification des étiquettes efficaces, selon un score global obtenu après traitement suivant un ordre décroissant.

5.4.2. *Formalisation d'un problème multicritère:*

- Soit $X = \{x_1, x_2, x_3, \dots, x_n\}$ l'ensemble des scénarios de correction. Ces scénarios sont différents, en nombre fini, et constituent l'intégralité des solutions possibles. Pour choisir le meilleur scénario de «X», on utilise un ensemble $F = \{f_1, f_2, f_3, \dots, f_q\}$ qui constitue une famille cohérente de critères.

Afin de juger chacun des scénarios de correction selon chacun des critères, on définit une fonction d'évaluation, cette fonction doit être maximisée ou minimisée selon le type de critère utilisé.

- Relation de dominance entre scénarios : on dit qu'un scénario « x_1 » domine un scénario « x_2 » si et seulement si : « $f_j(x_1) \geq f_j(x_2)$; $j = 1, \dots, q$ » sachant que l'une des inégalités, au moins, est strict « $f_j(x_1) \neq f_j(x_2)$; $j = 1, \dots, q$ ». En d'autres termes, « x_2 » ne possède aucune évaluation (selon tous les critères utilisés) qui soit meilleur de celle de « x_1 ».

Notons que pour cette relation de dominance, « $\geq$ » désigne « meilleur que ». Ainsi, si « f_j » est un critère à minimiser alors pour que « $f_j(x_1)$ » soit « meilleur que » « $f_j(x_2)$ », il faut qu'il vérifie : « $f_j(x_1) \leq f_j(x_2)$; $j = 1, \dots, q$ »

- Scénario efficace : un scénario « x » est dit efficace si et seulement si il n'est dominé par aucun scénario. L'ensemble

des scénarios efficaces est considéré comme l'ensemble des solutions les plus intéressantes (Belguith et Hamadou, 04) .

- Classement des scénarios : afin de déterminer le meilleur scénario de correction, nous procédons au classement des scénarios efficaces. L'objectif étant de déterminer le scénario qui jouit globalement des meilleures évaluations. Ainsi, nous calculons pour chaque scénario « x_i », un score d'évaluation global « $S(x_i)$ » qui représente la somme pondérée des différentes évaluations de « x_i » selon tous les critères de « F ».

5.4.3. *Différentes étapes d'une démarche d'aide multicritère à la décision* : La mise en place d'une démarche basée sur l'aide multicritères à la décision exige le plus souvent de suivre une succession d'étapes afin de générer un résultat. Ces étapes sont:

- Dresser la liste des actions potentielles, cette liste contient les actions (*i.e. solution, scénario, ...etc.*), qui sont provisoirement considérées comme étant candidates ou bien réaliste du point de vue du décideur.
- Dresser la liste des critères, une bonne application d'une démarche multicritères passe impérativement, par un bon choix concernant les critères, sur lesquelles le calcul sera posé. Ces critères seront définis en se basant sur les notions d'indifférence, de préférence stricte ou faible ou de non comparabilité.
- Définir une fonction d'évaluation (*i.e. fonction de performance*), Pour chaque critère sélectionné une fonction d'évaluation est définie. Cette dernière doit être maximisée ou bien minimisée selon le type de critère utilisé.
- Etablir un tableau de performance, ce tableau (*i.e. matrice d'évaluation*) contient tous les résultats des évaluations de chaque action potentielle suivant tous les critères ; de telle sorte que les lignes de ce tableau correspondent aux actions potentielles et les colonnes correspondent aux critères, et les intersections correspondent aux évaluations.

- **Pondération et agrégation des performances**, afin de s'assurer à la fois d'une bonne analyse des résultats et d'appliquer le concept de dominance (relation de dominance) une phase de normalisation est obligatoire. Cette phase est appelée la pondération des critères qui consiste à attribuer à chaque critère un poids justifiant ainsi son importance ; ainsi les critères les plus importants suivant les préférences du décideur auront les poids les plus élevés. Après la pondération une phase d'agrégation s'enchaîne afin de générer une évaluation en appliquant l'une des trois approches d'agrégation (globale, partielle ou locale).

5.4.4. *Description des critères de base pour l'évaluation des scénarios* : Mettre en place un critère capable d'influencer sur le choix de sélectionner un scénario « i » par rapport à d'autres scénarios de l'ensemble « X », n'est pas toujours une tâche facile.

Tout de même, le plus important lors de la définition d'un critère est le pouvoir de discrimination que ce critère peut exercer pour faire la différence entre les scénarios. Ainsi, plus la discrimination est forte plus facile sera la sélection du scénario adéquat. Cependant si un critère est discriminant dans une situation donnée il peut ne pas l'être dans une autre d'où l'obligation de construire un ensemble de critères qui doit satisfaire trois (03) conditions, à savoir: l'exhaustivité, la cohésion et la non redondance (Pomerol et Romero, 93).

Ainsi, nous proposons deux (02) critères de base pour discriminer entre les scénarios. Ces critères sont:

- **Critère 1: La concordance des voyelles**. Ce critère va vérifier la concordance entre les voyelles de l'unité lexicale et les voyelles de chaque scénario candidat, de telle sorte que chaque concordance vaut: un (1). Ainsi, il s'agit d'un critère à maximiser et la fonction d'évaluation qui va avec est l'addition (+).

• Critère 2: La Fréquence d'apparition. Ce critère s'appuie sur un calcul statistique fait sur la base d'un corpus annoté, de telle manière que le scénario qui se manifeste le plus souvent (une plus grande fréquence d'apparition) aura systématiquement le score le plus élevé. Chaque apparition vaut: un (1), il s'agit donc d'un critère à maximiser et la fonction d'évaluation qui va avec est l'addition (+).

5.4.5. Agrégation des critères « TOPSIS »:

Créée par Hwang et Yoon en 1981, cette méthode se base sur la relation de dominance qui résulte de la distance par rapport à la solution idéale. Son fondement consiste à choisir une solution qui se rapproche le plus de la solution idéal (*i.e. la meilleure sur tous les critères*) et de s'éloigner le plus possible de la pire solution qui dégrade tous les critères (Hwang et Yoon, 81). Cette méthode est composée de six (06) étapes:

• Etape 1: Calcul de la matrice de décision normalisée. Les valeurs normalisées « e_{ij} » sont calculées comme suit :

$$e'_{ij} = \frac{g_j}{\sqrt{\sum_i [g_j(a_i)]^2}} \quad \text{Avec :} \quad \begin{array}{l} i=1, \dots, m. \\ j=1, \dots, n. \end{array} \quad (1)$$

Les « $g_j(a_i)$ » correspondent aux valeurs déterministes des actions « i » pour le critère « j ».

• Etape 2: Calcul de la matrice de décision normalisée pondérée. Les valeurs normalisées pondérées sont calculées comme suit :

$$e''_{ij} = \pi_j \cdot e'_{ij} \quad \text{Avec :} \quad \begin{array}{l} i=1, \dots, m. \\ \sum \pi_j = 1 \end{array} \quad (2)$$

« π_j » est le poids du $j^{\text{ème}}$ critère et:

• Etape 3 : Détermination de la solution idéale « a^+ » et de la solution anti-idéal « a^- » :

$$\begin{aligned} a^+ &= \{\text{Max } e''_{ij}, i=1, \dots, m; \text{ et } j=1, \dots, n\}; \\ a^- &= \{\text{Min } e''_{ij}, i=1, \dots, m; \text{ et } j=1, \dots, n\}. \end{aligned} \quad (3)$$

- Etape 4 : Calcul des mesures d'éloignement ; L'éloignement entre les alternatives est mesuré par une distance euclidienne de dimension « n ». L'éloignement de l'alternative « i » par rapport à la solution idéale « a⁺ » qui peut être assimilé à la mesure d'exposition aux risques et donné par :

$$D_i^+ = \sqrt{\sum_{j=1}^n (e_{ij}^+ - e_j^*)^2} \quad \text{Avec: } \begin{matrix} i=1, \dots, m. \\ j=1, \dots, n. \end{matrix} \quad (4.1)$$

L'éloignement par rapport à la solution anti-idéal « a⁻ », quant à lui, peut être assimilé à la mesure de gestion des risques et il se calcule comme suit :

$$D_i^- = \sqrt{\sum_{j=1}^n (e_{ij}^- - e_j^*)^2} \quad \text{Avec: } \begin{matrix} i=1, \dots, m. \\ j=1, \dots, n. \end{matrix} \quad (4.2)$$

- Etape 5 : calculer un coefficient de mesure du rapprochement au profil idéal :

$$C_i^+ = \frac{D_i^-}{D_i^+ + D_i^-} \quad \text{Avec: } \begin{matrix} i=1, \dots, m. \\ 0 \leq C_i^+ \leq 1. \end{matrix} \quad (5)$$

- Etape 6 : Rangement des actions suivant leur ordre de préférences (« i » est meilleur que « j » si $C_i^+ > C_j^+$).

5.4.6. *Pondération des critères:* Comme la méthode TOPSIS ne permet pas de générer de manière automatique les poids pour les critères, on n'est obligé d'intégrer une autre méthode pour pondérer les critères à l'intérieur de TOPSIS. Cette méthode de pondération est l'Entropie, dont le fondement et l'algorithme sont données comme suit:

La méthode Entropie est une technique objective de pondération des critères, l'idée est qu'un critère « j » est d'autant plus important que la dispersion des évaluations des actions est importante. Ainsi les critères les plus importants sont ceux qui discriminent le plus entre les actions (dans notre

cas ce sont les étiquettes). Les étapes de cette méthode sont données comme suite :

- Etape1 : L'entropie d'un critère « j » est calculée selon les formules suivantes :

$$E_j = -K * \sum_{i=1}^n X_{ij} * \text{Log}(X_{ij}) \quad (6)$$

Où:

- « E_j » : Représente l'entropie du critère « j » ;
- « X_{ij} » : l'évaluation de l'action « i » suivant le critère « j ».
- « K » : est constante choisie de telle sorte que, pour tous « j », on a $0 \leq E_j \leq 1$, pour notre cas « K » est calculé comme suit :

$$K = \frac{1}{\text{Log}(n)}. \quad (7)$$

Où:

- « n » : étant le nombre de scénario de correction (étiquettes).
- Etape2 : L'entropie « E » est d'autant plus grande que les valeurs « e_j » sont proche. Ainsi, les poids seront calculés en fonction de la mesure de dispersion (opposée de l'entropie) :

$$D_j = 1 - E_j. \quad \text{Avec: } j=1, \dots, n. \quad (8)$$

- Etape 3 : les poids seront ensuite normalisés par :

$$W_j = \frac{D_j}{\sum_{j=1}^n D_j} \quad \text{Avec: } j=1, \dots, n. \quad (9)$$

6. Conclusion

Dans cet article, nous avons présenté notre système TAGHIT qui est un étiqueteur morphosyntaxique du texte arabe. Nous avons commencé par définir le problème de l'ambiguïté morphologique dans le traitement automatique de la langue arabe et les causes directe de ce problème ; ensuite nous avons proposé l'étiquetage morphosyntaxique comme solution à ce type de problème, toute en décrivant les approches de

désambiguïsation existantes. Avant de décortiquer l'architecture générale de notre système TAGHIT ainsi que ses différents constituants (module de segmentation, module d'analyse morphologique et module de désambiguïsation)

Mais notre principal objectif est de proposer une nouvelle approche de désambiguïsation, basée sur les techniques issues de l'aide multicritère à la décision.

Comme première conclusion, qu'on peut tirer de cette étude et le fait que l'approche multicritère peut être considérer comme une alternative de choix pour remplacer les deux (02) approches citez auparavant pour remédier au problème de l'ambiguïté morphologique qui constitue un point de passage qui condamne la réussite ou l'échec d'une application en T.A.L.A. concernant les résultats de nos travaux, ils seront publiés dans nos prochains articles.

Références

- ALLOTI, D. et PONSARD, C. (2005). Exposé sur les étiqueteurs statistiques et les étiqueteurs par contraintes.
- AL SULAITI, L. (2004). Deigning and Developing a Corpus of Contemporary Arabic, School of Computing, University of Leeds (UK).
- BELGUITH, L. et HAMADOU, A. (2004). Traitement des erreurs d'accord. RSTI-RIA, Volume 18, pp 679-707.
- BENHAMOUDA, B. (1993). Les clés de la langue arabe. O.P.U.
- CONSTANT, M. (2007). Etiquetage morphosyntaxique probabiliste", Master en informatique, Université Marne la Vallée, France .
- HWANG, C.R. et YOON, K. (1981) .Lectures Notes in Economics and Mathematical Systems. Springer-Verlag Berlin Heildelberg, New York, NY.
- LECHEL, M. (2005). Analyse et conception d'un correcteur grammatical libre pour le français. thèse de magister, université de Grenoble, France.
- POMEROL, J. et ROMERO, S. (1993). Choix multi-critère dans l'entreprise principe et pratique, Hermès.
- TUERLINCKX, L. La lemmatization de l'arabe non classique", U.C.L. Institut orientaliste, Belgique.
- VERGNE, J., GIGUET, E. 1998 Regards théorique sur le tagging. TALN, pp 22-31, Frence