L'utilisation des POMDP pour les résumés multi-documents orientés par une thématique

Yllias Chali[1], Sadid A. Hasan[1], Mustapha Mojahid[2]
(1) University of Lethbridge, Lethbridge, AB, Canada
(2) Université Paul Sabatier – IRIT, 118 Rte de Narbonne 31062 Toulouse Cedex
(1) yllias.chali@uleth.fr, (2) mustapha.mojahid@irit.fr

RESUME

L'objectif principal du résumé multi-documents orienté par une thématique est de générer un résumé à partir de documents sources en réponse à une requête formulée par l'utilisateur. Cette tâche est difficile car il n'existe pas de méthode efficace pour mesurer la satisfaction de l'utilisateur. Cela introduit ainsi une incertitude dans le processus de génération de résumé. Dans cet article, nous proposons une modélisation de l'incertitude en formulant notre système de résumé comme un processus de décision markovien partiellement observables (POMDP) car dans de nombreux domaines on a montré que les POMDP permettent de gérer efficacement les incertitudes. Des tests approfondis sur les jeux de données du banc d'essai DUC ont démontré l'efficacité de notre approche.

ABSTRACT

The main goal of topic-focused multidocument summarization is to generate a summary from the source documents in response to a given query or particular information requested by the user. This task is difficult in large part because there is no significant way of measuring whether the user is satisfied with the information provided. This introduces uncertainty in the current state of the summary generation procedure. In this paper, we model the uncertainty explicitly by formulating our summarization system as a Partially Observable Markov Decision Process (POMDP) since researchers in many areas have shown that POMDPs can deal with uncertainty successfully. Extensive experiments on the DUC benchmark datasets demonstrate the effectiveness of our approach.

MOTS-CLES : POMDP, Génération de résumé orienté par une thématique.
KEYWORDS : POMDP, Topic-focused multi-document summarization.

1 Introduction et travaux connexes

Un résumé multi-documents orienté par la thématique (i.e. par une requête) est utile dans la gestion des documents et les systèmes de recherche. Il peut fournir par exemple des services d'information personnalisés en fonction des besoins des utilisateurs. Dans cet article, nous considérons le problème de produire des résumés multi-documents orientés par la thématique reviendrait à extraire un sous-ensemble de phrases choisies à partir des documents originaux (Mani et Maybury, 1999). Juger de l'importance d'une phrase est l'aspect le plus essentiel de la génération de résumé. Cette tâche est difficile, en grande partie parce qu'il n'est pas certain que la phrase choisie soit suffisamment importante pour être considérée comme une phrase du résumé. Il est également difficile de garantir que les phrases choisies vont satisfaire totalement l'utilisateur. Ce problème peut être résolu en reformulant la tâche dans une problématique de prise de décisions séquentielles.

Les modèles de processus de décision markovien (MDP) se sont avérés utiles dans une variété de problèmes de décision (Puterman, 1994). La spécification d'un problème de décisions séquentielles en environnement totalement observable avec un modèle de transition markovien est appelé MDP (Russel et Norvig, 2003). Les MDP sont utiles pour modéliser la prise de décision dans des situations où les résultats sont en partie aléatoires et en partie dépendent du choix d'un décideur. À tout instant, un MDP est dans un certain état *s*, et une action disponible *a* est ensuite choisie déplaçant le MDP aléatoirement dans un nouvel état *s'* ; et une récompense correspondante $R_a(s, s')$ est attribuée. Ainsi, le nouvel état *s'* dépend de l'état actuel *s* et de l'action *a* en étant conditionnellement indépendant de tous les états et actions antérieurs. La tâche principale des MDP est de trouver une fonction de décision π qui spécifie une action particulière qui sera choisie dans un état *s* ; le but étant de choisir une politique qui maximise la récompense. Cependant, les MDP ne peuvent traiter l'incertitude où se trouvent les états de l'utilisateur, l'historique des transitions, et les actions. Le modèle MDP partiellement observables (POMDP) généralise le modèle MDP en permettant de prendre en compte également ces formes d'incertitude. Ainsi, dans différents domaines, plusieurs applications ont fait appel au POMDP (Young, 2006 ; Lison, 2010 ; Bui *et al*., 2007). Dans cet article, nous proposons d'utiliser le POMDP pour modéliser l'incertitude inhérente à la tâche de résumé multi-documents.

Dans notre tâche de résumé, l'environnement est partiellement observable, car il est incertain de savoir si une phrase choisie conduit le système à un état de résumé ou non. Ainsi, nous définissons un modèle d'observation $O(s, o)$ qui définit la probabilité de percevoir l'observation *o* dans l'état *s*. Ceci construit un ensemble d'états de croyance qui est une représentation de probabilités des états réels possibles. Puisque nous ne pouvons déterminer de manière sûre si nous avons atteint l'état résumé ou non, nous supposons le fait que l'agent a connaissance de tous les états de croyance et ainsi, la politique optimale peut être apprise par la transformation des états de croyance par les actions.

Pour formuler notre tâche en termes de POMDP, nous supposons qu'un ensemble de documents sources et leurs résumés de référence (RR) créés par un expert humain sont donnés garantissant la satisfaction des utilisateurs. Lorsqu'une phrase est sélectionnée comme candidate pour un résumé, sa probabilité d'observation est calculée en mesurant sa parenté avec le RR. Une valeur de récompense est attribuée lorsque le système atteint l'état de résumé. Nous représentons chaque phrase d'un document comme un vecteur de traits-valeurs (Voir section 4). Notre approche tente de produire des résumés automatiques qui soient le plus proche des RR. Dans la phase d'apprentissage, le système apprend les poids appropriés des traits en modélisant la relation entre le RR et le résumé candidat extrait. Une fois le modèle d'observation est appris, l'agent atteint l'état final de résumé et la phase d'apprentissage se termine. Les poids finaux appris pour chaque attribut sont utilisés pour produire des résumés à partir de nouvelles données dans la phase de test. Nous avons utilisé une version de l'algorithme $Q(\lambda)$ de la descente du gradient de Watkins (Sutton et Barto, 1998) pour résoudre notre modèle POMDP proposé. Des tests ont été menés sur les jeux de données de référence DUC[1] et les résultats des évaluations ont montré l'efficacité de notre approche. Dans la suite de l'article nous décrirons successivement la terminologie des POMDP, les bases formelles de notre travail définissant la tâche de résumé multi-documents orienté par une thématique comme un problème d'un POMDP, l'espace de traits pour

[1] http://duc.nist.gov/

représenter les phrases, les paramètres expérimentaux et les résultats de l'évaluation.

2 Terminologie des POMDP

Contrairement aux MDP qui offrent un bon cadre statistique pour permettre la planification dans un environnement totalement observable, un POMDP fournit un modèle mathématique pour les problèmes de décisions séquentielles dans des environnements partiellement observables. Le principal avantage d'un POMDP est de fournir une architecture complète pour modéliser l'incertitude inhérente au problème étudié (Young, 2006).

2.1 Définition formelle

Un POMDP est un tuple <*S, A, Z, T, O, R*> (Lison, 2010) où :

1. *S* est *l'espace d'états* défini comme un ensemble d'états mutuellement exclusifs.
2. *A* représente *l'espace d'actions* possibles qu'un agent peut effectuer dans un état.
3. *Z* est l'espace des observations que l'agent peut percevoir.
4. *T(s, a, s') = Pr(s'|s, a)* est *le modèle de transition* qui dénote la probabilité d'atteindre l'état s' si l'action *a* est effectuée dans l'état *s*.
5. *O(s, o)* est le *modèle d'observation* qui spécifie la probabilité de percevoir l'observation *o* dans l'état *s*.
6. *R(s, a)* est la *fonction de récompense* qui définit l'utilité de l'agent à effectuer une action *a* étant dans l'état *s*.

2.2 L'état de croyance

Un POMDP suppose que l'état du monde n'est pas directement observable. Par conséquent, un POMDP peut seulement déduire des informations utiles à partir d'observations disponibles. Cette incertitude peut être codée par l'état de croyance *b*, qui est une distribution de probabilité sur tous les états possibles (Lison, 2010). On note *b(s)* la probabilité attribuée à l'état réel par l'état de croyance *b*. Nous pouvons calculer un état de croyance courant comme une distribution de probabilité conditionnelle sur les états réels étant donnée la séquence des observations et des actions effectuées jusqu'ici. Dans l'état de croyance courant *b(s),* si l'action est effectuée et une observation *o* est perçue, le nouvel état de croyance sera donné par (Russel et Norvig, 2003) :

$$b'(s') = \alpha O(s',o)\sum_{s} T(s,a,s')b(s) \qquad (1)$$

où α est une constante de normalisation qui met la somme des états de croyance à 1. Dans un POMDP, l'action optimale d'un agent s'appuie uniquement sur l'état de croyance courant de l'agent puisque l'agent n'est pas conscient de son état actuel. Par conséquent, le cycle de décisions d'un agent POMDP consiste à exécuter l'action *a* en tenant compte de l'état actuel de croyance *b*, de l'observation *o*, et de la transition au nouvel état de croyance (équation 1). La tâche suivante consiste à effectuer une recherche de la politique optimale dans l'espace continu des états de croyances (Russel et Norvig, 2003). Un état de croyance initial *b0* (ayant une distribution uniforme) est spécifié à l'exécution lors de l'initialisation de notre système proposé.

2.3 Les politiques

La solution au problème d'un POMDP doit préciser l'action possible de l'agent dans un état qu'il pourrait atteindre dans un certain intervalle de temps. Ce type de solution est appelé une politique et noté π (Russel et Norvig, 2003). L'idée générale d'un POMDP est que l'action optimale ne dépend que de l'état de croyance courant de l'agent, car il n'a accès qu'à l'état réel actuel. Par conséquent, la politique optimale (qui donne la plus grande utilité espérée), notée $\pi^*(b)$, est une application des états de croyance à l'action. Nous définissons la valeur de l'action a dans l'état de croyance b selon la politique π notée $Q^\pi(b, a)$:

$$Q^\pi(b,a) = E_\pi\left\{R_t \mid b_t = b, a_t = a\right\}$$
$$= \left\{\sum_{k=0}^{\infty} \gamma k^{r}{}_{t+k+1} \mid b_t = b, a_t = a\right\} \qquad (2)$$

Ici, E_π représente la valeur espérée étant donné que l'agent suit la politique π, R_t est le *rendement espéré* qui est définie en fonction de la séquence de récompenses, r_{t+1}, r_{t+2}, ... où r_t est la récompense numérique que l'agent reçoit à l'instant t. Nous appelons Q^π la fonction action-valeur pour la politique π. γ représente le facteur d'actualisation qui détermine l'importance des futures récompenses. Le système essaie de trouver la politique optimale par l'itération de la politique. Une fois la politique optimale π^* est obtenue, l'agent choisit les actions en utilisant le principe d'utilité maximale espérée (Russel et Norvig, 2003).

3 Modèle POMDP pour le résumé

3.1 Environnement, état et action

La tâche de résumé multi-documents orientée par une thématique considère une requête q (une thématique) et une collection de documents voisins $D = \{d_1, d_2, ..., d_n\}$ pour générer un résumé. L'espace d'états qui décrit l'environnement, inclut les états de résumé et de non résumé. L'action optimale de l'agent, ne sachant pas dans quel état il est, dépend de son état de croyance courant. L'espace d'états des croyances est continu, représenté par une distribution de probabilités, comprend l'ensemble des états que l'agent pourrait prendre. Dans le problème d'extraction de résumé, l'objectif est de sélectionner un ensemble de phrases importantes de la collection de documents donnée pour constituer le résumé candidat. Dans chaque état de croyance, on dispose d'un ensemble d'actions possibles qui pourraient être opérées sur l'environnement où chaque action indique la sélection d'une phrase particulière (en utilisant la fonction politique de l'équation 2) à partir des phrases des documents restantes non encore incluses dans le résumé candidat.

3.2 Observation et récompense

Lorsque nous choisissons une phrase considérée importante, nous calculons sa probabilité d'observation en mesurant sa parenté avec les résumés de références fournis. Nous mesurons cette similarité en utilisant l'outil automatique ROUGE (Recall Oriented Understudy For Gisting Evaluation) (Lin, 2004). Les mesures concernent le nombre d'unités qui se chevauchent tels que les n-grammes, les mot-séquences, et les paires de mots entre les deux résumés (de référence et extrait). Les mesures de ROUGE considérées sont ROUGE-

N (N = 1, 2, 3, 4), ROUGE-L, ROUGE-W et ROUGES. Dès que la probabilité[2] la plus élevée de la croyance est atteinte, on arrive à l'état de résumé, et une récompense de 1 est attribuée, sinon 0 est retourné. Le POMDP pour un résumé fonctionne de la manière suivante : à chaque instant, l'environnement est dans un certain état non observé *s*. Cela signifie qu'une phrase est sélectionnée et déplacée dans le résumé candidat sans savoir si le système a atteint l'état de résumé ou non. Puisque *s* n'est pas connu de manière précise, une distribution de probabilité (espace des états de croyance) sur tous les états possibles (résumé ou non résumé) est maintenue, où *b(s)* indique la probabilité d'être dans un état *s* particulier. Sur la base de l'état actuel de croyance, une action optimale *a* est choisi et reçoit une récompense *r* basée sur la probabilité d'observation, et l'environnement se déplace à un nouvel état non observé *s'*. L'environnement génère alors une observation *o'* pour actualiser l'état de croyance. L'ensemble du processus tente de produire un résumé similaire au résumé de référence en assurant également que le résumé candidat ne contient pas d'informations redondantes.

3.3 Résolution du POMDP

Dans notre formulation, le nombre d'états réels est égal à deux (état résumé, état non résumé). Puisque l'environnement est partiellement accessible, nous avons modélisé le problème comme un POMDP en introduisant un nombre infini d'états de croyance, qui sont observables par l'agent. Pour résoudre ce problème, nous considérons une approche d'approximation fonctionnelle au problème. Nous représentons la fonction approximée action-valeur comme une fonction linéaire paramétrée avec le vecteur de paramètres $\vec{\theta}_t$. A chaque paire état de croyance-action *(b, a)*, il existe un vecteur colonne de traits, $\vec{\varphi}_b = (\varphi_b(1), \varphi_b(2), \ldots, \varphi_b(n))^T$ avec le même nombre de composants que $\vec{\theta}_t$. La fonction approximée action-valeur est donnée par :

$$Q_t(b,a) = \vec{\theta}_t^T \vec{\varphi}_b = \sum_{i=1}^{n} \theta_t(i)\varphi_b(i) \qquad (3)$$

Nous considérons notre problème comme un problème de prise décision séquentielle à horizon infini qui trouve un vecteur de paramètres $\vec{\theta}$ pour maximiser *Q(b, a)* à partir de l'équation 3. Nous utilisons un algorithme du gradient de la politique pour résoudre notre problème de POMDP. Les Algorithmes du gradient de la politique ont tendance à estimer les paramètres θ en effectuant une montée du gradient stochastique. Le gradient approximé par l'interaction avec l'environnement, et la récompense qui en résulte est utilisé pour mettre à jour l'estimation de θ. Les algorithmes du gradient de la politique optimise un objectif non convexe et ne sont garantis que pour trouver un optimum local (Branavan *et al.*, 2009). Nous utilisons une version de l'algorithme modifiée *Q (λ)* de Watkins avec descente de gradient linéaire (Sutton et Barto, 1998) en appliquant la politique ε-gloutonne afin de déterminer la meilleure action possible pour sélectionner les phrases les plus importantes. Nous nous servons de la politique ε-gloutonne (ce qui signifie que la plupart du temps cette politique choisit une action avec la valeur maximale estimée, mais une action est sélectionnée au hasard avec une probabilité ε) pour trouver l'équilibre entre l'exploration et l'exploitation pendant la phase d'apprentissage. Nous avons posé ε = 0.1.

[2] Dans chaque état, la probabilité d'observation est mise à jour lorsque que le système essaie de générer un résumé qui est un pas de plus vers le résumé de référence. La probabilité la plus élevée de la croyance est observée lorsque le résumé du système a la correspondance la plus proche avec le résumé de référence.

Ainsi, notre algorithme choisit une action avec la meilleure action-valeur à 90% et il choisit une action au hasard à 10%.

Les étapes de notre solution POMDP sont présentées dans l'algorithme 1 où φ est un vecteur de traits-valeurs (Voir section 4) utilisé pour représenter chaque phrase de document ; $\vec{\theta}_t$ est le vecteur de poids pour le vecteur de traits que le système va apprendre et γ est le coefficient d'actualisation utilisé pour calculer la récompense d'une paire état-action. Le facteur d'actualisation détermine l'importance des futures récompenses. Nous avons gardé la valeur initiale de γ à 0.1 qu'on fait diminuer d'un facteur correspondant au nombre d'itération. Tant que la politique initiale sélectionne les actions gloutonnes, l'algorithme continue l'apprentissage de la fonction action-valeur avec une la politique gloutonne. Cependant, quand une action exploratoire est sélectionnée par la politique de comportement, les traces d'éligibilité[3], $\vec{e}$, sont initialisées pour tous les couples état-action et elles sont mises à jour en deux étapes. Dans la première étape, si une action exploratoire est prise, elles sont mises à 0 pour toutes les paires état-action. Dans le cas contraire, les traces d'éligibilité pour toutes les paires état-action sont décomposées par γ^{λ}. Dans la deuxième étape, la valeur de la trace d'éligibilité de la paire actuelle état-action est incrémentée de 1 tout en accumulant les traces. La version originale de l'algorithme *Q(λ)* de Watkins utilise une approximation linéaire de la fonction de la descente de gradient avec des traits binaires. Toutefois, étant donné que nous traitons un mélange de traits à valeur réelle et booléenne (voir section 4), nous avons modifié l'algorithme pour déclencher une mise à jour différente pour les traces d'éligibilité. Dans la deuxième étape de la mise à jour des traces d'éligibilité, on incrémente la valeur avec le score du trait correspondant. L'ajout d'une étape de saut aléatoire évite les maximums locaux dans notre algorithme. Le paramètre λ définit quel crédit nous pouvons accorder aux états antérieurs. α est le paramètre de la taille de pas pour le procédé de la descente de gradient qui est réduit par un facteur de 0,99 de sorte que l'apprentissage converge vers le but. δ est le taux d'erreur de prédiction état-valeur. Un « épisode » est lancé à la phase d'apprentissage pour chaque thème et se termine quand l'état de résumé final est atteint. La condition « *b* n'est pas terminale » est satisfaite lorsque *b* se réfère à l'état final de résumé. L'état de croyance initial a une distribution de probabilité uniforme sur les états réels. L'action initiale est choisie en fonction de la probabilité d'observation des phrases. A notre connaissance, la formulation proposée par la version modifiée de l'algorithme *Q(λ)* de Watkins est originale dans la façon dont il représente la tâche de résumé orientée sur une thématique.

4 Espace des traits

Nous représentons chaque phrase d'un document comme un vecteur de trait-valeur (φ dans l'algorithme 1). Notre ensemble de traits comprend deux types, le premier caractérise l'importance d'une phrase dans un document et le second mesure la similarité entre chaque phrase et la requête de l'utilisateur. Ces traits ont été adoptés par plusieurs travaux connexes à cette problématique (Edmundson, 1969 ; Litvak *et al.*, 2010 ; Schilder et Kondadadi, 2008).

[3] Une trace d'éligibilité est un enregistrement temporaire de l'occurrence d'un événement, comme la visite d'un état ou le choix d'une action (Sutton et Barto, 1998).

4.1 Mesure de l'importance d'une phrase

Position des phrases : les phrases qui se trouvent au début et à la fin d'un document ont souvent tendance à inclure les informations les plus précieuses. Nous avons analysé manuellement la collection de documents donnés et nous avons constaté que la première et les 3 dernières phrases d'un document répondent bien à ce trait. Nous leurs attribuons ainsi le score de 1 et 0 aux autres phrases.

Longueur des phrases : les plus longues phrases contiennent plus de mots et ont une plus grande probabilité de contenir des informations importantes. Par conséquent, une longue phrase a de meilleure chance de figurer dans un résumé. Nous donnons le score de 1 à une longue phrase et 0 aux autres. L'analyse manuelle de la collection de documents nous a amené à fixer le seuil de 11 mots pour considérer qu'une phrase est longue.

Correspondance avec le titre : Si nous trouvons des chevauchements de mots exacts, de synonymes, ou d'hyponymes entre le titre et une phrase, nous lui attribuons le score de 1, sinon 0. Nous utilisons la base de données WordNet[4] (Fellbaum, 1998) pour l'accès aux synonymes et aux hyponymes.

Algorithme Q(λ) de Watkins modifié :

Entrés : $\alpha, \vec{\theta}, \lambda, \gamma, \varphi, \varepsilon,$ nombre d'itérations T
Sorties : vecteur $\vec{\theta}$ des poids appris
Initialisation : $\vec{\theta}$ à $\vec{0}$, $\vec{e}$ à $\vec{0}, \alpha$ à 0.01, λ à 0.9, γ à 0.1
b, a ← état de croyance et action de l'épisode initiaux
φ ← ensemble des traits présents dans *s*, *a*
Pour *i = 1 .. T* **Faire**
  Si b est non terminal **Alors**
    Pour $i \in \varphi$ **Faire**
      $e(i) \leftarrow e(i) + \varphi(i)$
    faire l'action a, observer la récompense et l'état suivant b
    $\delta \leftarrow r - \sum_i \varphi(i)\theta(i)$
    Pour a $\in A(b)$ **Faire**
      φ ← ensemble de traits présents dans *b*, *a*
      $Q_a \leftarrow \sum_i \varphi(i)\theta(i)$
    $\delta \leftarrow \delta + \gamma \max_a Q_a$
    $\theta \leftarrow \theta + \alpha\delta\vec{e}$
    $\delta \leftarrow 0.99 \times \alpha$
    Si probabilité ≤ 1- ε **Alors**
      Pour $a \in A(b)$ **Faire**
        $Q_a \leftarrow \sum_i \varphi(i)\theta(i)$
      $a \leftarrow \arg\max_a Q_a$
      $\vec{e} \leftarrow \gamma^{\lambda}\vec{e}$
    sinon $a \leftarrow$ *action aléatoire* $\in A(b)$
      $\vec{e} \leftarrow \vec{0}$
retourner $\vec{\theta}$

[4] Nous utilisons dans cette recherche la version 3.0 de WordNet.

Entités nommées (EN) : Le score de 1 est attribué à une phrase contenant une classe d'EN parmi : Personne, Localisation, Organisation, Entité géopolitique, Installation, Date, Monnaie, Nombre, Horaire. Nous pensons qu'une EN accroît l'importance d'une phrase et nous utilisons le système OAK (Sekine, 2002) pour la reconnaissance d'entités nommées.

Lexique spécifique : La pertinence probable d'une phrase est affectée par la présence de marqueurs lexicaux (« important », « impossible », « en conclusion », « enfin », etc.). Nous utilisons une liste[5] de 228 expressions et nous donnons le score de 1 à une phrase contenant une expression dans cette liste et 0 sinon.

4.2 Mesure de similarité avec la requête de l'utilisateur

Chevauchement de n-gramme : Il s'agit du rappel entre la requête et la phrase candidate dans laquelle n représente la longueur du n-gramme (n = 1, 2, 3, 4). Cette valeur est obtenue en divisant le total des co-occurrences de n-grammes dans la requête et dans la phrase candidate, par le nombre de n-grammes dans la phrase (Lin, 2004).

LSC : Etant donné deux séquences S1 et S2, la plus longue séquence commune (LSC) de S1 et S2 est une sous-séquence commune avec la longueur maximale.

LSCP : La plus Longue Séquence Commune Pondérée (LSCP) améliore la méthode de base LSC en prenant en compte la longueur des correspondances consécutifs rencontrés (Lin, 2004). La LSCP permet de conserver la durée des correspondances consécutives dans une table à deux dimensions de programmation dynamique. Le calcul de WLCS est basé sur la F-mesure entre une requête et une phrase.

Saut-bigramme : ce trait mesure le chevauchement de bigrammes entre une phrase candidate et une phrase requête. Le Saut-bigramme compte scrupuleusement toutes les paires de mots en correspondance, alors que LSC s'intéresse seulement à la plus longue séquence commune.

Chevauchement de mots exacts : Il s'agit de calculer le nombre de mots équivalents dans la phrase candidate et la pharse de la requête.

Chevauchement de Synonymes : C'est le chevauchement entre la liste des synonymes des mots lexicaux[6] extraite de la phrase candidate et les mots de la requête associée. Les mots liés à la requête sont obtenus en les remplaçant par leur sens premier synonyme utilisant WordNet (Fellbaum, 1998). Nous considérons que les *synsets* premiers de chaque mot.

Chevauchement des hyperonymes et des hyponymes : C'est le chevauchement entre la liste des hyperonymes et hyponymes (jusqu'au niveau 2 dans WordNet) des noms extraits de la phrase et les mots de la requête associée.

Chevauchement du glossaire : Notre système extrait les glossaires pour les noms propres de WordNet. Le chevauchement des glossaires est le chevauchement entre la liste des mots lexicaux qui sont extraits de la définition du glossaire des noms dans la phrase candidate, et les mots de la requête associée.

---

[5] Nous avons construit notre lexique en se référant aux expressions de transition disponibles dans http://www.smart-words.org/transitionwords.html

[6] Les mots lexicaux sont les noms, les verbes, les adverbes et les adjectifs.

Trait syntaxique : La première étape pour calculer la similarité syntaxique entre la requête et la phrase consiste à analyser leurs arbres syntaxiques en utilisant un analyseur syntaxique ; nous utilisons celui de (Charniak, 1999)[7]. La similarité entre deux arbres syntaxiques est calculée en utilisant la *fonction noyau sur les arbres* (Collins et Duffy, 2001).

Chevauchement des Éléments de Base (EB) : Nous extrayons les EB des phrases dans la collection de documents en utilisant le package des EB d'ISI[8]. Nous filtrons ensuite les BE en vérifiant s'ils contiennent *un mot de la requête ou un mot connexe*. Nous obtenons ainsi le meilleur score de chevauchement des EB (Hovy *et al*., 2005).

5 Expérimentation et évaluation

5.1 Définition de la tâche et corpus

Cet article s'intéresse à la tâche de résumé multi-documents orientée par une thématique (i.e. orientée par une requête) comme cela est définit dans la Document Understanding Conference, DUC-2007 : *« Etant donnée une question complexe (description du sujet) et une collection de documents pertinents, la tâche consiste à produire un résumé bien construit n'excédant pas 250 mots ».* L'ensemble des documents[9] DUC-2006 et 2007 provenaient du corpus AQUAINT, qui est composé d'articles provenant de fils de presse de l'Associated Press et New York Times (1998 - 2000) et de la Xinhua News Agency (1996-2000). Nous utilisons les groupes de documents (contenant chacun 25 articles de presse) de DUC-2006 pour apprendre les poids respectifs de chaque trait considéré qui seront exploités pour produire des résumés pour les groupes de documents de DUC-2007.

5.2 Paramètres du système

Le jeu de données produit un nombre de phrases volumineux. Pour simplifier notre tâche nous filtrons les 100 premières phrases d'un groupe de documents donnés en employant une approche supervisée basée sur la méthode d'entropie maximale (MaxEnt). Nous choisissons MaxEnt car il a montré de bonnes performances en résumé automatique de texte (Ferrier, 2001). Les systèmes supervisés nécessitent une grande quantité de données pendant l'apprentissage. Nous appliquons les noyaux de sous-séquences de chaîne étendue (Extended String Subsequence Kernel - ESSK) pour étiqueter automatiquement toutes les phrases de DUC-2006 et produire suffisamment de données pour l'apprentissage. L'ESSK a été appliqué avec succès pour l'annotation automatique (Hirao *et al*., 2004) ; nous l'exploitons dans le calcul de similarité entre chaque phrase candidate et le résumé de référence. Les N premières phrases sont ensuite choisies en fonction du score donné par ESSK et étiquetées « +1 » (phrase résumé), et les autres « -1 » (phrase non résumé). Nous construisons le système MaxEnt utilisant le package[10] MaxEnt de Lin. Pour définir l'exponentiel avant les valeurs λ dans les modèles MaxEnt, un paramètre supplémentaire est utilisé dans le package lors de l'apprentissage. Nous conservons la valeur α comme valeur par défaut. Le modèle MaxEnt appris est utilisé pour prédire les étiquettes des phrases non

[7] Disponible sur ftp://ftp.cs.brown.edu/pub/nlparser/
[8] http://www.isi.edu/~cyl/BE
[9] DUC-2006 et DUC-2007 ont fourni respectivement 50 et 45 groupes de documents.
[10] http://www.cs.ualberta.ca/~lindek/downloads.htm

observées des données DUC-2007. Les valeurs de probabilité correspondant aux étiquettes prédites sont utilisées pour classer les phrases.

Pour résoudre notre modèle POMDP, nous appliquons l'algorithme *Q(λ)* modifié de Watkins sur les données de DUC-2006 et nous utilisons les résultats des poids finaux pour obtenir les 250 mots du résumé à partir des 100 premières phrases (déjà sélectionnées par le système MaxEnt de base) des goupes de documents de DUC-2007. Nous avons également généré des résumés de 250 mots en utilisant le système MaxEnt seul. Pour comparer les performances du système POMDP, nous avons construit trois autres systèmes supervisés en utilisant les techniques bien connues : Machines à vecteurs de support, Modèles de Markov cachés et Champs conditionnels aléatoires (respectivement en anglais SVM, HMM, CRF).

Nous utilisons les mêmes (ESSK étiquetés) données dans la phase d'apprentissage de ces systèmes. Nous utilisons également un modèle de résumé non supervisé pour évaluer le modèle POMDP ; les approches supervisées sont souvent plus difficiles à appliquer à des ensembles de données arbitraires où des modèles de résumés générés par l'humain ne sont pas disponibles. L'algorithme de regroupement K-means est utilisé pour construire le système non supervisé. Dans tous ces systèmes, le même ensemble de traits (Section 4) est utilisé pour représenter les phrases du document comme vecteurs de traits.

5.3 Evaluation automatique : ROUGE

En DUC-2007, chaque thème et son groupe de documents ont été remis aux 4 différents évaluateurs du NIST. L'évaluateur a créé ensuite un résumé de 250 mots qui répond au besoin d'informations exprimées dans le sujet. Ces multiples « résumés de référence » sont utilisés dans l'évaluation de notre contenu de résumé. Nous avons considéré les mesures d'évaluation largement pratiquées, la précision (P), le rappel (R) et la F-mesure pour notre tâche d'évaluation des résumés générés par le système en utilisant la boîte à outils d'évaluation automatique ROUGE (Lin, 2004), qui a été adoptée par DUC.

5.3.1 Comparaison avec différents systèmes

Les tableaux 1 et 2 donnent les scores ROUGE-2 et ROUGE-SU de tous les systèmes, et le tableau 3 indique les intervalles de confiance à 95% de tous les systèmes afin de pouvoir faire des comparaisons significatives. Ces résultats montrent que le système POMDP est le plus performant, la plupart du temps, prouvant ainsi l'efficacité de notre modèle POMDP, à l'exception d'un petit nombre cas[11]. La raison de cette performance est due au fait que dans le cas des approches supervisées et non supervisées, le comportement appris est basé sur un corpus d'apprentissage fixe alors que le modèle POMDP met à jour en permanence son état de croyance en se basant sur la probabilité d'observation et utilise la récompense obtenue pour réajuster les poids des traits dans chaque itération afin de fournir un environnement d'apprentissage plus efficace. En particulier, l'avantage de l'approche POMDP est obtenu en prenant des décisions séquentielles, là où les autres systèmes vont traiter indépendamment chaque phrase (SVM), ou considérer seulement les dépendances entre les phrases directement adjacentes (CRF, HMM).

[11] Les chevauchements des intervalles de confiance ROUGE-SU pour SVM, MaxEnt et POMDP indiquent qu'ils ne sont pas significativement différents les uns des autres.

Systèmes	Rappel	Précision	F-mesure
SVM	0.0801	0.0878	0.0838
HMM	0.0865	0.0951	0.0906
CRF	0.0767	0.0844	0.0804
MaxEnt	0.0815	0.0897	0.0854
K-mens	0.0779	0.1072	0.0902
POMDP	0.1286	0.1065	0.1164

Tableau 1 : Les mesures ROUGE-2

Systèmes	Rappel	Précision	F-mesure
SVM	0.1324	0.1592	0.1445
HMM	0.1349	0.1636	0.1478
CRF	0.1238	0.1487	0.1461
MaxEnt	0.1339	0.1611	0.1461
K-mens	0.1348	0.1742	0.1520
POMDP	0.2089	0.1432	0.1694

Tableau 2 : Les mesures ROUGE-SU

Systèmes	ROUGE-2	ROUGE-SU
SVM	0.0678 – 0.1027	0.1243 – 0.1703
HMM	0.0838 – 0.0967	0.1306 – 0.1593
CRF	0.0629 – 0.0964	0.1161 – 0.1546
MaxEnt	0.0681 – 0.1036	0.1267 – 0.1636
K-mens	0.0662 – 0.0953	0.1241 – 0.1594
POMDP	0.1078 – 0.1263	0.1635 – 0.1762

Tableau 3: Les intervalles de confiance à 95% pour les différents systèmes

5.3.2 Comparaison avec les systèmes de base

Dans le tableau 4, notre modèle POMDP proposé est comparé aux systèmes de base (SB) officiels de DUC-2007. SB-1 retourne toutes les meilleures phrases (à hauteur de 250 mots) dans le champ <TEXT> du plus récent document. L'idée principale de SB-2 est d'ignorer le thème narratif tout en générant automatiquement des résumés basés sur une formulation HMM[12]. Selon ces résultats, le modèle POMDP proposé dépasse très largement tous les systèmes de base. Le tableau 5 indique les intervalles de confiance à 95% des F-mesures ROUGE permettant de vérifier la significativité des résultats.

Systèmes	ROUGE-2	ROUGE-SU
SB-1	0.060039	0.10507
SB-2	0.09382	0.14641
POMDP	0.1164	0.1694

Tableau 4 : Comparaison avec les Systèmes de Base (F-scores)

Systèmes	ROUGE-2	ROUGE-SU
SB-1	0.0563 – 0.0643	0.1007 – 0.1091
SB-2	0.0892 – 0.0980	0.1422 – 0.1506
POMDP	0.1078 – 0.1263	0.1635 – 0.1762

Tableau 5 : Intervalles de confiance à 95%

5.3.3 Comparison avec l'état de l'art

Dans le tableau 6, notre système POMDP est comparé aux systèmes ayant participé au DUC-2007. La moyenne-DUC représente la moyenne des scores ROUGE de tous les systèmes participant à DUC-2007. Les scores du meilleur système au DUC-2007 (Pingali *et al.*, 2007) sont également indiqués dans le tableau. Nous constatons que notre système a atteint des scores plus élevés que les scores moyens ROUGE de tous les systèmes participants à DUC-2007 tout en rivalisant de très près avec le meilleur système. Le tableau 7 donne les intervalles de confiance à 95% des F-mesures ROUGE.

[12] http://duc.nist.gov/pubs/2004papers/ida.conroy.ps

Systèmes	ROUGE-1	ROUGE-2
Moyenne-DUC	0.4006	0.0955
Meilleur système	0.4388	0.1228
POMDP	0.4370	0.1164

Tableau 6 : Comparaison avec les systèmes de l'état de l'art (F-scores)

Systèmes	ROUGE-1	ROUGE-2
Meilleur système	0.4316 – 0.4459	0.1180 – 0.1276
POMDP	0.4273 – 0.4486	0.1078 – 0.1263

Tableau 7 : Intervalles de confiance à 95%

Systèmes	Qualité ling.	Qualité Réponses
SR-1	4.24	1.86
SR-2	4.48	2.71
Meilleur	4.11	3.40
SVM	3.30	3.50
HMM	3.20	3.10
CRF	3.00	2.70
MaxEnt	3.20	2.90
K-mens	3.60	3.40
POMDP	4.00	3.80

Tableau 8 : Scores moyens de la qualité linguistique et des réponses

5.4 Evaluation Manuelle

Nous avons mené une évaluation manuelle intensive afin d'analyser l'efficacité de notre approche, en demandant à deux universitaires diplômés de langue maternelle anglaise de juger[13] la qualité linguistique des résumés et des réponses globales conformément aux directives d'évaluation[14] DUC-2007. Le tableau 8 présente la qualité moyenne linguistique et les scores des réponses globales de tous les systèmes. Nous pouvons constater que le système POMDP dépasse de manière significative[15] tous les systèmes dans la plupart des cas, tout en se rapprochant de très près des systèmes de référence et du meilleur système en termes de qualité linguistique. Ce résultat s'explique par le fait que notre modèle POMDP ne considère aucun algorithme de post-traitement ou d'ordonnancement de phrases pour améliorer les résumés générés par le système. Cependant, en termes de réponse générale du contenu, notre système POMDP a emporté le meilleur score en apportant une meilleure précision pour répondre au besoin d'information demandé par l'utilisateur.

6 Conclusion

La contribution principale de cet article est une formulation du problème POMDP du résumé multi-documents orienté par une thématique. Comme il n'est pas certain de savoir si un résumé centré sur un thème répond au besoin d'information de l'utilisateur ou non, nous avons proposé que cette incertitude soit modélisée en considérant la tâche de résumé comme étant un problème POMDP. Nous avons comparé le système POMDP proposé avec quatre méthodes d'apprentissage supervisées bien connues : MaxEnt, CRF, SVM et HMM. Nous avons également évalué notre approche sur un modèle de clustering k-means non supervisé. Notre système POMDP a dépassé les deux systèmes de références officiels et les scores moyens de DUC tout en s'approchant des performances du meilleur système de DUC-

[13] L'accord inter-annotateur de Cohen est calculé pour les deux juges ($\kappa = 0.61$), ce qui traduit un degré important d'accord entre eux (Landis et Koch, 1977).
[14] http://www-nlpir.nist.gov/projects/duc/duc2007/qualityquestions.txt
[15] Les différences sont statistiquement significatives à $p < .05$ en utilisant le test du t de Student.

2007. Les intervalles de confiance à 95% ont montré qu'il n'existe pas de différence significative entre notre système et ce lauréat. Nous avons aussi procédé à une évaluation approfondie manuelle des résumés générés par le système. Les résultats montrent l'efficacité de notre système POMDP proposé dans la modélisation de l'incertitude de la tâche de résumé multi-documents orientée par une thématique.

Parmi les perspectives à ce travail, nous envisageons étudier l'apport des indices fournis par la mise en forme matérielle et plus particulièrement dans les citations et les structures énumératives.

Remerciements

Les recherches présentées dans cet article ont été prises en charge par le Natural Sciences and Engineering Research Council (NSERC) du Canada – Discovery Grant et l'Université de Lethbridge.

Références

S.R.K. Branavan, H. Chen, L. S. Zettlemoyer, and R. Barzilay. 2009. Reinforcement Learning for Mapping Instructions to Actions. In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP 2009)*, pages 329–332, Suntec, Singapore.

T. H. Bui, B. van Schooten, and D. Hofs. 2007. Practical Dialogue Manager Development using POMDPs. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium*, pages 215– 218.

Y. Chali, S. A. Hasan, K. Imam: A reinforcement learning framework for answering complex questions. *IUI 2011*, pages 307-310

Y. Chali, S. A. Hasan, K. Imam: Improving the performance of the reinforcement learning model for answering complex questions. *CIKM 2012,* pages 2499-2502

E. Charniak. 1999. A Maximum-Entropy-Inspired Parser. In *Technical Report CS-99-12*, Brown University, Computer Science Department.

M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver, Canada.

H. P. Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery (ACM)*, 16(2):264–285. C. Fellbaum. 1998. WordNet - An Electronic Lexical Database. Cambridge, MA. MIT Press.

L. Ferrier, 2001. *A Maximum Entropy Approach to Text Summarization*. M.Sc. thesis, School of Artificial Intelligence, Division of Informatics, University of Edinburgh.

T. Hirao, J. Suzuki, H. Isozaki, and E. Maeda. 2004. Dependency-based Sentence Alignment for Multiple Document Summarization. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 446–452, Geneva, Switzerland.

E. Hovy, C. Y. Lin, and L. Zhou. 2005. A BE-based Multi-document Summarizer with Query Interpretation. In *Proceedings of the Document Understanding Conference*, Canada.

T. Joachims. 1999. Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods*

- *Support Vector Learning.*

J. R. Landis and G. G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.

C. Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, Barcelona, Spain.

P. Lison. 2010. Towards relational POMDPs for adaptive dialogue management. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 7–12.

M. Litvak, M. Last, and M. Friedman. 2010. A New Approach to Improving Multilingual Summarization using a Genetic Algorithm. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* pages 927–936. ACL.

I. Mani and M. T. Maybury, 1999. *Advances in Automatic Text Summarization*. MIT Press. A. K. McCallum. 2002. MALLET: A Machine Learning for Language Toolkit.

D. Pelleg and A. Moore. 1999. Accelerating exact k-means algorithms with geometric reasoning. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 277–281. ACM.

P. Pingali, R. K, and V. Varma. 2007. IIIT Hyderabad at DUC 2007. In *Proceedings of the Document Understanding Conference*, Rochester, USA. NIST.

M. L. Puterman. 1994. Markov Decision Processes– Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc, New York.

S. Russel and P. Norvig, 2003. *Artificial Intelligence A Modern Approach, 2nd Edition*. Prentice Hall.

S. Ryang, T. Abekawa: Framework of Automatic Text Summarization Using Reinforcement Learning. *EMNLP-CoNLL 2012,* pages 256-265

F. Schilder and R. Kondadadi. 2008. FastSum: Fast and Accurate Query-based Multi-document Summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 205–208. ACL.

S. Sekine. 2002. Proteus Project OAK System, http://nlp.nyu.edu/oak.

R. S. Sutton and A. G. Barto. 1998. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts, London, England.

S. Young. 2006. Using POMDPs for dialog management. In *Proceedings of the 1st IEEE/ACL Workshop on Spoken Language Technologies (SLT06).*