Optimiser l'adaptation en ligne d'un module de compréhension de la parole avec un algorithme de bandit contre un adversaire

Emmanuel Ferreira, Alexandre Reiffers-Masson, Bassam Jabaian et Fabrice Lefèvre
CERI-LIA, Université d'Avignon et des Pays de Vaucluse, France
prenom.nom@univ-avignon.fr

RÉSUMÉ

De nombreux modules de compréhension de la parole ont en commun d'être probabilistes et basés sur des algorithmes d'apprentissage automatique. Deux difficultés majeures, rencontrées par toutes les méthodes existantes sont : le coût de la collecte des données et l'adaptation d'un module existant à un nouveau domaine. Dans cet article, nous proposons un processus d'adaptation en ligne avec une politique apprise en utilisant un algorithme de type bandit contre un adversaire. Nous montrons que cette proposition peut permettre d'optimiser un équilibre entre le coût de la collecte des retours demandés aux utilisateurs et la performance globale de la compréhension du langage parlé après sa mise à jour.

ABSTRACT

Adversarial bandit for optimising online active learning of spoken language understanding

Many speech understanding modules have in common to be probabilistic and to rely on machine learning algorithms to train their models from large amount of data. The difficulty remains in the cost of collecting such data and the time for updating an existing model to a new domain. In this paper, we propose to drive an online adaptive process with a policy learnt using the Adversarial Bandit algorithm. We showthat this proposition can optimally balance the cost of gathering valuable user feedbacks and the overall performance of the spoken language understanding module after its update.

MOTS-CLÉS : Compréhension de la parole, Apprentissage sans données de références, Bandit contre un adversaire, Adaptation en ligne.

KEYWORDS: Spoken language understanding, zero-shot learning, Adversarial bandit problem, online adaptation.

1 Introduction

Dans un système de dialogue, le rôle du module de compréhension de la parole est d'extraire, pour chaque énoncé d'utilisateur, des hypothèses qui représentent son contenu sémantique. Cela peut être représenté par une séquence d'actes de dialogue sous la forme `acttype(slot=value)`. Les `acttype` sont indépendants de la tâche et transmettent l'intention de l'utilisateur lors de la communication, alors que les `slot` et les `value` dépendent du domaine d'application et correspondent à l'information que le système peut manipuler (entrées dans une base de données, commandes à un robot, ...). Par exemple, l'énoncé « Bonjour je cherche un restaurant français dans la partie sud de la ville » correspond à l'acte de dialogue « `hello(), inform(food=french), inform(area=south)` ».

Dans la dernière décennie, les systèmes de compréhension ont évolué progressivement vers des approches probabilistes apprises sur de grandes quantités de données (Hahn *et al.*, 2010; Deoras & Sarikaya, 2013). Plusieurs travaux ont proposé des approches semi-supervisées (Celikyilmaz *et al.*, 2011; Heck & Hakkani-Tur, 2012) ou non-supervisées (Tur *et al.*, 2011; Lorenzo *et al.*, 2013) pour faire face au manque de données annotées. Dans ce même but, d'autres travaux ont proposé de porter les systèmes à travers les langues et les domaines (Lefèvre *et al.*, 2012; Jabaian *et al.*, 2013). Plusieurs recherches ont aussi étudié des procédures d'apprentissage actif (Active Learning, AL) pour réduire le coût d'annotation et de vérification de corpus (Gotab *et al.*, 2010; Bayer & Riccardi, 2013).

Par ailleurs, dans (Ferreira *et al.*, 2015), nous avons proposé une méthode sans données de référence pour la compréhension de la parole. Cette méthode est basée sur une représentation vectorielle compacte de mots (word embedding (Mikolov *et al.*, 2013a)) utilisée pour généraliser une base de connaissance d'un domaine cible (bases de données, description ontologique...). Cette approche ne nécessite pas de données annotées et peut atteindre instantanément des performances état-de-l'art. Une stratégie d'adaptation en ligne a également été proposée pour affiner le modèle de façon progressive avec une supervision minimale. En effet, avec cette stratégie, les utilisateurs doivent confirmer certaines hypothèses faites par le système par des retours binaires, mais sans corriger explicitement les erreurs qui nécessitent des retours utilisateurs plus complexes. Cela permet de corriger certaines erreurs de classification mais sans possibilité d'ajouter de nouveaux concepts ou valeurs dans le modèle. Aussi, dans cet article, nous proposons d'étendre cette stratégie d'adaptation en ligne afin de répondre également à ce problème et d'être ainsi en mesure d'étendre le modèle avec de nouvelles connaissances en permanence.

Pour définir cette nouvelle stratégie, nous proposons de considérer le problème d'adaptation comme un problème de Bandit contre un Adversaire (Adversarial Bandit). Cette proposition vise à minimiser le coût de supervision tout en demandant à l'utilisateur des retours qui peuvent avoir le maximum d'impact sur le modèle. Les algorithmes de bandit ont été largement étudiés dans la communauté de l'apprentissage automatique (Auer *et al.*, 2002; Bubeck & Cesa-Bianchi, 2012). Leur objectif est de déterminer le meilleur compromis entre l'exploration des options qui ont donné le meilleur rendement (gains) dans les itérations précédentes et l'exploration de nouvelles options qui pourraient donner une meilleure performance à l'avenir. Peu de travaux ont déjà employé ce genre de techniques pour optimiser un module de traitement de langage naturel. Parmi ceux-ci, on pourra donner comme exemple (Ralaivola *et al.*, 2011) où les auteurs appliquent un algorithme de bandit contextuel pour raffiner un classificateur multiclasses avec des retours utilisateurs de type oui/non sur une tâche visant à identifier le motif d'un appel téléphonique (*call routing*). Nous montrons que la technique proposée dans ce papier permet d'obtenir de bonnes performances avec un coût faible et une supervision minime sur une tâche de compréhension de la parole en utilisant les données de la seconde campagne d'évaluation Dialog State Tracking Challenge (DSTC2) (Henderson *et al.*, 2014).

2 Compréhension de la parole avec un apprentissage sans données de référence (Zero-shot)

Le modèle de compréhension employée dans notre étude est celui proposé dans (Ferreira *et al.*, 2015) et présenté dans la figure 1. Il est basé sur une analyse sémantique sans données de référence (Zero-shot Semantic Parser, ZSSP) et fait usage de trois composants principaux. Le premier est un espace sémantique F basé sur une représentation vectorielle compacte de mots apprise avec des

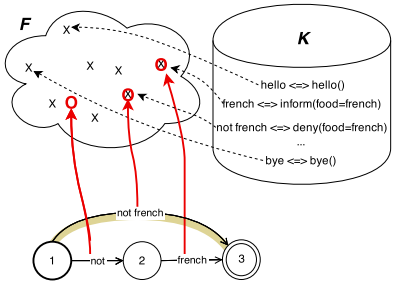

FIGURE 1: Modèle de d'analyseur sémantique sans données de référence (modèle ZSSP)

réseaux de neurones profonds (Mikolov *et al.*, 2013a) à partir de données générales.

Le deuxième composant est une base de connaissances K qui peut être vue comme un dictionnaire d'exemples dans F utilisé pour relier l'espace sémantique à l'espace de sortie du système. Dans K, des coefficients d'affectation mesurent la correspondance entre un exemple et un acte de dialogue connus. Ces valeurs, indiquent le degré de correspondance entre une phrase et une étiquette sémantique. Ces coefficients, initialisés à 1 pour l'ensemble d'exemples/actes extrait de l'ontologie, seront ensuite modifiés en fonction des retours des utilisateurs.

Enfin, un analyseur sémantique qui extrait une liste ordonnée des meilleures hypothèses de séquences d'étiquettes sémantiques à partir d'un transducteur à états finis représentant l'ensemble des hypothèses pour une phrase utilisateur (scorées par des informations issues de F et de K). Par exemple pour la phrase « not french », trois formes de surface différentes sont extraites : « not », « french » et « not french ». Ces formes de surface sont ensuite projetées dans l'espace sémantique F (cercles rouges dans la figure 1) pour être comparées aux vecteurs associés aux exemples de la base de connaissance K (croix noires dans la Fig. 1). Pour ce faire un critère de similarité (e.g. similarité cosinus) entre ces vecteurs est employé. On note que, malgré la présence de K, le ZSSP ne dispose pas de données de référence qui seraient des phrases entières annotées sémantiquement.

3 Optimisation en ligne de la stratégie d'adaptation du modèle

L'objectif principal des travaux présentés dans ce papier est d'enrichir la stratégie d'adaptation en ligne (telle que décrite dans (Ferreira *et al.*, 2015)) pour permettre également la création de concepts/valeurs (extension de domaine) tout en tentant d'optimiser un ratio coût/amélioration du modèle. Dans cette étude, nous adoptons une stratégie simple basée sur un algorithme de bandit contre un adversaire pour résoudre le problème d'optimisation de la stratégie d'adaptation du modèle. Avant d'aller plus loin dans la formulation, nous proposons en premier lieu d'expliciter la problématique sur un cas statique (i.e. sans adaptation).

3.1 Cas statique

Nous postulons que le système a le choix entre plusieurs actions vis à vis de l'utilisateur afin d'améliorer la détection automatique des étiquettes sémantiques associées aux énoncés de celui-ci.

Nous devons d'abord définir l'espace d'actions considéré (qui correspondent donc aux bras des bandits dans la littérature relative aux algorithmes de bandits). Toutefois, nous pouvons déjà prévoir que chaque action implique une collaboration plus ou moins importante de la part de l'utilisateur. Par conséquent, nous introduisons une mesure de l'effort utilisateur relatif à l'action effectivement choisie par le système sous la forme d'une fonction de coût. Nous définissons également une mesure de l'inefficacité du modèle que nous tenterons de réduire dans le temps. Cette mesure nous permet de quantifier l'amélioration de modèle résultant d'une action spécifique. Enfin le problème de l'adaptation du modèle est formulé comme un problème d'optimisation linéaire où le système a la pleine connaissance de la fonction objectif.

3.2 Espace d'actions et fonction de coût traduisant l'effort utilisateur

Lorsque l'utilisateur fournit une phrase, le système peut choisir une action (à partir d'une distribution de probabilité) parmi un ensemble $\mathcal{I}$ de M actions. Dans cette configuration préliminaire, nous considérons le cas où $M = 3$ et où $\mathcal{I}$ peut être défini comme :

$$\mathcal{I} := \{\text{Skip, YesNoQuestions, AskAnnotation}\}. \quad (1)$$

Soit $i \in \mathcal{I}$ l'indice de l'action. Nous supposons que l'effort de l'utilisateur (coût de l'action), $\phi(i) \in \mathbb{N}$, peut être mesuré par le nombre d'échanges réalisés entre le système et l'utilisateur pour mener à bien l'action i.

Soit $d \in [0, 1]$ le poids moyen des arcs constituant le meilleur chemin de la machine à états finis utilisée par l'analyseur sémantique . Selon l'action de raffinement choisie i, d est mis à jour en $d'(i) \in [0, 1]$ en raison de la modification du modèle qui en résulte. Une des composantes étant la fonction de coût nous cherchons à résoudre un problème de minimisation globale, l'inefficacité est prise en compte, au lieu de l'efficacité.

Une description des différentes actions et de leurs coûts et inefficacités associés est donnée ci-dessous :
•**Skip :** n'appliquer aucune mise à jour au modèle. Le coût de cette action est toujours considéré comme étant nul ($\phi(\text{Skip}) = 0$) puisque l'utilisateur n'est pas sollicité et la mesure de l'inefficacité reste donc constante, $d'(Skip) = d$.
•**YesNoQuestions :** mettre à jour K avec les réponses oui/non données par l'utilisateur aux questions de confirmation sur les étiquettes sémantiques détectées dans la meilleure hypothèse sémantique. Si cette action est prise, le système tentera en premier lieu une confirmation générale sur la meilleure hypothèse pour un coût de 1. En cas de négation, une confirmation ($+1$ sur le coût) sera demandée pour chaque étiquette sémantique détectée.
Ces évaluations utilisateurs sont converties en un ensemble U de m tuples $U := (c_l, T_l, f_l)_{1 \leq l \leq m}$, où ($c_l$, T_l) est un couple forme-de-surface / étiquette-sémantique proposé à l'utilisateur et f_l est son retour (1 positif, 0 négatif). Compte tenu de K et U après chaque interaction, l'algorithme 1 est utilisé pour mettre à jour K en K^* Ainsi, $d'(YesNoQuestions) = \delta$ où δ est le nouveau poids moyen de l'énoncé récemment actualisé dans K^*.
•**AskAnnotation :** demander à l'utilisateur d'annoter son tour de parole complètement pour mettre à jour K avec de nouveaux exemples positifs. Si cette action est prise, le système tentera en premier lieu une confirmation générale sur la meilleure hypothèse pour un coût de 1. Dans le cas d'un rejet, l'utilisateur devra procéder à l'annotation étiquette par étiquette de l'énoncé. Pour ce faire, nous supposons que pour chacune d'entre elles l'utilisateur informera le système de la localisation dans sa phrase de l'acte de dialogue qu'il s'apprête à annoter ($+1$ sur le coût) puis identifiera consécutivement

l'$acttype$, le $concept$ et la $valeur$.
Par cette action de nouveaux concepts et valeurs pourront donc être ajoutés par l'utilisateur au module SLU (extension du domaine). Si l'hypothèse sémantique de l'énoncé est validée dans son intégrité par l'utilisateur nous considérons que tous les couples forme-de-surface / étiquette-sémantique extraits de la meilleure hypothèse sémantique (plus court chemin) ont été évalués de façon positive par l'utilisateur. Sinon, les m' couples forme-de-surface/étiquette-sémantique annotés par l'utilisateur sont considérés comme un ensemble de tuples $U := ((c_l, T_l, 1)_{1 \leq l \leq m'})$. Dans ce cas, de nouveaux concepts et valeurs peuvent être ajoutés dans les sorties possibles du modèle (ajout d'une colonne dans K^*). En raison du fait que des parties de l'énoncé sont désormais dans K^* en tant qu'exemples positifs, $d'(AskAnnotation) \approx 0$.

Finalement, nous devons définir une fonction de perte telle que le système, par le fait de chercher à l'optimiser, réduira dans le même temps la mesure de l'inefficacité actualisée $d'(i)$ et la mesure de l'effort de l'utilisateur $\phi(i)$. Ainsi, nous proposons de définir la fonction de perte $l(i) \in [0, 1]$ comme étant la combinaison convexe des deux mesures précédemment introduites :

$$l(i) := \underbrace{\gamma d'(i)}_{\substack{\textbf{amélioration} \\ \textbf{du modèle}}} + \underbrace{(1-\gamma)\frac{\phi(i)}{\phi_{max}}}_{\textbf{effort utilisateur}} \tag{2}$$

où $\gamma \in [0, 1]$ permet de régler l'importance de l'amélioration du modèle sur l'effort utilisateur dans le processus d'optimisation. $\phi_{max} \in \mathbb{N}_+$ correspond au nombre maximal d' échanges possibles entre le système et l'utilisateur (dans un même tour de dialogue).

Soit $\mathbf{p} \in \Delta(3) := \{\mathbf{q} \in \mathbb{R}_+^3 \mid \sum_{i \in \mathcal{I}} q(i) = 1\}$ la distribution de probabilité sur les différentes actions. L'objectif d'adaptation du modèle est donc défini comme :

$$\min_{\mathbf{p} \in \Delta(3)} E[l] = \sum_i p(i) l(i). \tag{3}$$

Si nous avions une pleine connaissance de $l(i)$ pour chaque action i, le problème d'adaptation du modèle serait équivalent à celui consistant à résoudre $\min_i\{l(i)\}$. Cependant, dans le scénario considéré, ce cadre ne peut pas être appliqué car la fonction de perte $l(i)$ n'est pas connue explicitement (pas observable pour toutes les actions à tous les instants). De ce fait, les algorithmes de bandit sont les plus adaptés.

3.3 Cas du bandit contre un adversaire

Pour le problème d'adaptation du modèle par une méthode de bandit contre un adversaire les paramètres connus sont l'espace d'actions $\mathcal{I}$ et le coefficient $\gamma \in [0, 1]$. À chaque tour $t = 1, 2, \ldots$, le système reçoit un énoncé utilisateur, en extrait la meilleure hypothèse sémantique et obtient d_t puis choisit une action $i_t \in \mathcal{I}$, éventuellement en ayant recours à une action aléatoire (exploration).

Une fois l'action i_t exécutée, le système calcule la nouvelle mesure d'inefficacité $d'_t(i_t)$; l'effort utilisateur à t, $\phi_t(i_t)$, qui correspond au nombre d'échanges effectivement réalisés entre le système et l'utilisateur lors de la réalisation de i_t et la fonction de perte :

$$l_t(i_t) = \gamma d'_t(i_t) + (1-\gamma)\phi_t(i_t)$$

Algorithm 1 Bandit contre un Adversaire, l'algorithme Exp3

Sachant : $\gamma' \in [0, 1]$
Initialiser les poids $w_i(1) = 1$ pour $i = 1, \ldots, M$.
for chaque tour t **do** :
    - calculer $p_i(t) = (1 - \gamma')\frac{w_i(t)}{\sum_{j=1}^{M} w_j(t)} + \frac{\gamma'}{M}$ pour chaque i.
    - déterminer la prochaine action i_t aléatoirement selon la distribution $p_i(t)$.
    - observer la récompense $x_{i_t}(t)$.
    - calculer la récompense estimée $\hat{x}_{i_t}(t) = x_{i_t}(t)/p_{i_t}(t)$.
    - mettre à jour les poids :
    $w_{i_t}(t + 1) = w_{i_t}(t)e^{\gamma' \hat{x}_{i_t}(t)/M}$ et $w_j(t + 1) = w_j(t)$ pour tout autre action j.

Le but sera de trouver $i_1, i_2, \ldots$ tels que pour chaque T, le système minimise la perte cumulée :

$$\sum_{t=1}^{T} l_t(i_t) = \gamma \sum_{t=1}^{T} d'_t(i_t) + (1 - \gamma) \sum_{t=1}^{T} \phi_t(i_t).$$

Aucune hypothèse n'est formulée sur $d'_t(i_t) \in [0, 1]$ et $\phi_t(i_t) \in [0, 1]$. Ainsi, nous ne présupposons pas de l'effet qu'a une action i_{t-l}, avec $l \in \{1, \ldots, t - 1\}$, sur la fonction de perte pour le tour t. Ce choix est justifié par le fait qu'une phrase utilisateur ne peut pas être prédite avec précision par le système sans connaissances a priori robustes.

Parmi les algorithmes de la littérature, nous avons retenu l'algorithme Exp3 (Auer *et al.*, 2002) (voir algorithme 1). Il s'agit d'un algorithme efficace lorsqu'un petit nombre de bras est en jeu. Une preuve mathématique des performances relativement élevées de cet algorithme est notamment donnée dans (Bubeck & Cesa-Bianchi, 2012).

3.4 Simulation

Afin de tester l'algorithme d'apprentissage de la politique d'adaptation du modèle, nous avons choisi dans ces travaux de simuler les réponses de l'utilisateur. Pour ce faire, nous avons mis en place un indicateur à même de déterminer la qualité de la meilleure proposition du ZSSP en fonction d'une référence. En raison du fait que les étiquettes sémantiques $acttype(concept = valeur)$ n'étaient pas alignées aux mots dans le corpus considéré (ici DSTC2) et sachant que ce dernier est une condition préalable pour pouvoir simuler l'annotation en séquence de couples forme-de-surface/étiquette-sémantique nous avons dû donc au préalable procéder à un alignement automatique similaire à celui proposé dans (Huet & Lefèvre, 2011). Ainsi, à chaque tour, nous avons suffisamment d'informations pour être en mesure de répondre avec précision à l'action de la machine (séquences d'actes de dialogue de référence et leurs alignements aux mots). Ici, un sous-ensemble de transcriptions de l'ensemble d'apprentissage de DSTC2 (750 transcriptions d'énoncés utilisateur) est exploité pour évaluer le modèle d'adaptation en ligne.

Dans notre configuration expérimentale, un utilisateur simulé est employé pour répondre aux actions d'adaptation du modèle pour chaque tour de parole dans le dialogue d'origine. Cet utilisateur peut faire usage de trois actions distinctes : *Affirm*, *Negate* et *Inform*. Les actions *Affirm* et *Negate* sont employées pour répondre aux demandes de confirmation liées à l'application des actions d'adaptation

du modèle (AskAnnotation et YesNoQuestions). L'action *Inform* est utilisée exclusivement dans les échanges supplémentaires ayant lieu dans le cadre de l'action système AskAnnotation (par exemple *Inform(acttype=request)*, *Inform(boundaries="austrian food")*). Ici, nous supposons que les sous-dialogues d'annotation peuvent être gérés par un système réel avec un niveau de précision élevé (par exemple en utilisant une grammaire bien calibrée et une logique d'interaction finement réglée). Bien sûr cette hypothèse devra être confirmée en pratique.

4 Expériences et résultats

Notre étude expérimentale a été menée sur une tâche de compréhension de la parole en utilisant les données de la seconde campagne d'évaluation Dialog State Tracking Challenge (DSTC2) (Henderson *et al.*, 2014) qui couvre le domaine de la recherche d'information à propos de restaurants. Nous exploitons ces données (transcriptions, annotations sémantiques...) comme corpus d'apprentissage et de test pour évaluer notre approche d'apprentissage en ligne pour l'étiquetage sémantique. Ainsi, les transcriptions du corpus de test (9890 énoncés utilisateur) seront utilisées pour le test. Un sous-ensemble des transcriptions du corpus d'apprentissage (1472 énoncés) seront exploitées dans notre modèle de raffinement en ligne du modèle.

La configuration du modèle sans données de référence utilisé pour appliquer notre méthode d'apprentissage en ligne correspond à celle présentée dans (Ferreira *et al.*, 2015). Un modèle *word2vec* (Mikolov *et al.*, 2013a) a été utilisé pour apprendre une représentation vectorielle des mots avec 300 dimensions. Ce modèle a été appris avec l'algorithme *Skip-gram* (avec une fenêtre de 10 mots et un softmax hiérarchique) sur une grande quantité de données disponibles librement et présentant une grande couverture thématique.

Ce type de représentation présente certaines régularités avec les propriétés syntaxiques et sémantiques des mots comme celles montrées dans (Mikolov *et al.*, 2013b) ainsi qu'une structure linéaire permettant la combinaison des représentations des mots par une simple addition vectorielle élément par élément. Cette technique est donc utilisée pour projeter nos formes de surface vers leur représentation sémantique vectorielle de type word2vec vue comme une somme des représentations individuelles de chaque mot les constituant.

La base de connaissance utilisée pour notre expérience est initialisée grâce aux descriptions ontologiques fournies dans le cadre du DSTC2 (e.g. listes des concepts/valeurs) ainsi que d'un ensemble d'informations génériques. La sémantique du domaine est constituée de 8 concepts et 215 valeurs. Au total 16 *acttype* sont considérés, il en résulte donc 663 étiquettes sémantiques possibles. Nous avons définis manuellement 53 formes de surface associées aux différents *acttype*s. Par exemple « say again » est utilisé pour représenter l'acte *repeat()*.

Du fait que la technique Exp3 emploie une certaine forme d'exploration stochastique (ici $\gamma' = 0,2$) nous utiliserons une moyenne faite à partir de 20 processus indépendants d'apprentissage en ligne.

La figure 2 donne l'évolution de la probabilité $p_i(t)$ associée à chaque action i telle qu'estimée par l'algorithme Exp3 (avec $\gamma = 0,5$). Nous pouvons observer que chaque action est sélectionnée avec une probabilité comparable au début de la procédure d'optimisation, Exp3 explore. Puis, à mesure que le nombre de tours considérés augmente, on observe que l'influence des deux actions YesNoQuestions et Skip croît. On remarque cependant un avantage clair à l'action Skip lorsqu'il devient plus difficile d'obtenir de nouvelles informations eu égard au coût impliqué pour les collecter.

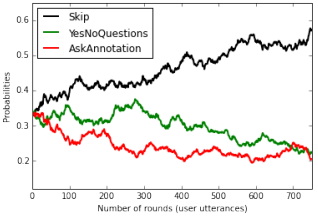

FIGURE 2: Distribution de probabilité estimée par Exp3 au cours du temps sur les différentes actions.

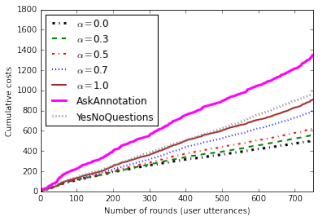

FIGURE 3: Impact de γ sur l'effort utilisateur (coûts) cumulé.

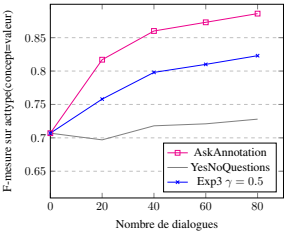

FIGURE 4: Impact du nombre de dialogues employés sur les différentes techniques d'adaptation en ligne en terme de F-mesure.

Dans la figure 3 on compare l'effet de γ sur la stratégie apprise par Exp3 en terme d'effort utilisateur cumulé. Les stratégies **AskAnnotation** et **YesNoQuestions** (stratégies réalisant la même action à chaque tour) sont introduites ici à des fin de comparaison comme méthodes de référence. Nous considérons les performances pour $\gamma \in \{0,0,3,0,5,0,7,1\}$. Nous pouvons observer que la stratégie **AskAnnotation** est la plus coûteuse, suivie par **YesNoQuestions**. Faire varier le paramètre γ semble donc avoir l'effet escompté sur l'apprentissage de la stratégie d'adaptation. Ainsi, plus γ est grand, moins le coût a un impact sur l'apprentissage. De ce fait, lorsque celui-ci est totalement ignoré dans la fonction de perte ($\gamma = 1,0$), l'algorithme Exp3 a tendance à favoriser les actions les plus coûteuses car elles permettent de réduire significativement la mesure d'inefficacité du modèle. Ainsi, γ offre un moyen simple et direct pour régler le compromis entre l'effort de l'utilisateur et l'efficacité du modèle pour une application donnée.

Enfin, dans la figure 4 Exp3 ($\gamma = 0,5$) est comparé à **Askannotation** et **YesNoQuestion** en terme de F-mesure sur les transcriptions du corpus de test. Comme prévu **AskAnnotation** obtient les meilleures performances. En effet, l'utilisation des nouvelles annotations permet au modèle ZSSP de couvrir dynamiquement des actes de dialogue supplémentaires grâce à la mise à jour de K avec des exemples robustes. Du fait que l'objectif de l'algorithme Exp3 est de trouver un compromis entre réduire l'effort de l'utilisateur et l'efficacité du modèle, cette méthode est capable d'atteindre à plus faible coût des performances proches de celles obtenues avec **AskAnnotation** et bien meilleures que celles observées pour **YesNoQuestion** (cette dernière ne pouvant pas capturer de nouveaux concepts).

5 Conclusion

Dans ce papier une approche de bandit contre un adversaire a été employée pour optimiser la stratégie d'adaptation d'un modèle d'analyse sémantique sans données de références et permettre de résoudre le problème d'une couverture initiale limitée sur la sémantique de domaine spécifique. Il a été montré que cette technique est efficace et à même de fournir un moyen pratique de formaliser un compromis entre l'effort de supervision fourni par l'utilisateur et l'amélioration de l'efficacité du système. La généralisation de l'approche d'optimisation proposée ainsi qu'une comparaison plus poussée avec d'autres algorithmes de bandit fera l'objet de futurs travaux.

Références

AUER P., CESA-BIANCHI N., FREUND Y. & SCHAPIRE R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, p. 48–77.

BAYER A. & RICCARDI G. (2013). On-line adaptation of semantic models for spoken language understanding. In *ASRU*.

BUBECK S. & CESA-BIANCHI N. (2012). Regret analysis of stochastic and nonstochastic multiarmed bandit problems. *Foundations and Trends in Machine Learning*, **5**(1), 1–122.

CELIKYILMAZ A., TUR G. & HAKKANI-TUR D. (2011). Leveraging web query logs to learn user intent via bayesian latent variable model. In *ICML*.

DEORAS A. & SARIKAYA R. (2013). Deep belief network based semantic taggers for spoken language understanding. In *INTERSPEECH*.

FERREIRA E., JABAIAN B. & LEFÈVRE F. (2015). Online adaptative zero-shot learning spoken language understanding using word-embedding. In *ICASSP*.

GOTAB P., DAMNATI G., BÉCHET F. & DELPHIN-POULAT L. (2010). Online slu model adaptation with a partial oracle. In *INTERSPEECH*.

HAHN S., DINARELLI M., RAYMOND C., LEFÈVRE F., LEHNEN P., DE MORI R., MOSCHITTI A., NEY H. & RICCARDI G. (2010). Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE TASLP*, **19**(6), 1569–1583.

HECK L. & HAKKANI-TUR D. (2012). Exploiting the semantic web for unsupervised spoken language understanding. In *SLT*.

HENDERSON M., THOMSON B. & WILLIAMS J. (2014). The second dialog state tracking challenge. In *SIGDIAL*.

HUET S. & LEFÈVRE F. (2011). Unsupervised alignment for segmental-based language understanding. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*.

JABAIAN B., BESACIER L. & LEFÈVRE F. (2013). Comparison and Combination of Lightly Supervised Approaches for Language Portability of a Spoken Language Understanding System. *IEEE TASLP*, **21**(3), 636–648.

LEFÈVRE F., MOSTEFA D., BESACIER L., ESTEVE Y., QUIGNARD M., CAMELIN N., FAVRE B., JABAIAN B. & ROJAS-BARAHONA L. (2012). Robustness and portability of spoken language understanding systems among languages and domains : the PORT-MEDIA project. In *LREC*.

LORENZO A., ROJAS-BARAHONA L. & CERISARA C. (2013). Unsupervised structured semantic inference for spoken dialog reservation tasks. In *SIGDIAL*.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

MIKOLOV T., YIH W. & ZWEIG G. (2013b). Linguistic regularities in continuous space word representations. In *NAACL-HLT*.

RALAIVOLA L., FAVRE B., GOTAB P., BÉCHET F. & DAMNATI G. (2011). Applying multiclass bandit algorithms to call-type classification. In *Workshop on Automatic Speech Recognition & Understanding, ASRU*.

TUR G., HAKKANI-TUR D., HILLARD D. & CELIKYILMAZ A. (2011). Towards unsupervised spoken language understanding : Exploiting query click logs for slot filling. In *INTERSPEECH*.