Évaluation de l'apprentissage incrémental par analogie

Vincent Letard[1, 2, 3] Gabriel Illouz[2, 3] Sophie Rosset[1, 3]
(1) LIMSI CNRS, F-91405 Orsay
(2) Université Paris-Sud, F-91405 Orsay
(3) Université Paris-Saclay
vincent.letard@limsi.fr, gabriel.illouz@u-psud.fr, sophie.rosset@limsi.fr

RÉSUMÉ
Cet article examine l'utilisation du raisonnement analogique dans le contexte de l'apprentissage incrémental. Le problème d'apprentissage sous-jacent développé est le transfert de requêtes formulées en langue naturelle vers des commandes dans un langage de programmation. Nous y explorons deux questions principales : Comment se comporte le raisonnement par analogie dans le contexte de l'apprentissage incrémental ? De quelle manière la séquence d'apprentissage influence-t-elle la performance globale ? Pour y répondre, nous proposons un protocole expérimental simulant deux utilisateurs et différentes séquences d'apprentissage. Nous montrons que l'ordre dans la séquence d'apprentissage incrémental n'a d'influence notable que sous des conditions spécifiques. Nous constatons également la complémentarité de l'apprentissage incrémental avec l'analogie pour un nombre d'exemples d'apprentissage minimal.

ABSTRACT
Incremental Learning From Scratch Using Analogical Reasoning

This paper explores the application of analogical reasoning on incremental machine learning. The learning task is about transferring natural language requests to formal language commands. This work explores two questions for applying analogy in incremental learning situation : How does the analogical reasoning behave in incremental learning situation ? How do the conditions on the learning sequence influence the performance ? To address this issue, an experimental setup is proposed in which two different users are simulated with various learning sequences. We point out that the order of the incoming learning examples has a notable influence only under particular conditions. The experiments also show that incremental learning and analogical reasoning are complementary for a small number of learning examples.

MOTS-CLÉS : apprentissage incrémental, raisonnement analogique, transfert de langage.

KEYWORDS: incremental learning, analogical reasoning, language transfer.

1 Introduction

L'apprentissage incrémental (aussi séquentiel ou en ligne/*online learning*), est le processus par lequel une entité accroît ses connaissances au cours du temps, en même temps qu'elle les utilise. On l'oppose à l'apprentissage hors ligne (*batch learning*) pour lequel l'ensemble d'apprentissage est acquis généralement lors d'une unique phase initiale. Il n'est pas adapté pour l'ajout de connaissances *a posteriori*. Réitérer la phase d'apprentissage est souvent trop coûteux car cela met en jeu des

algorithmes hors ligne. Au contraire, les approches d'apprentissage incrémentales collectent des exemples positifs ou négatifs à chaque action du système et les intègrent aux connaissances avec un coût limité, afin d'améliorer les décisions futures.

Le contexte auquel est appliqué l'apprentissage incrémental concerne la conception d'un assistant capable de transférer des requêtes en langue naturelle vers des commandes en langage formel. L'une des difficultés provient de la relative rareté des exemples. En effet, ils n'arrivent qu'au rythme auquel l'humain qui utilise le système les fournit. Il est donc crucial d'exploiter du mieux possible cet ensemble restreint d'exemples pour produire rapidement des réponses (commandes) pertinentes.

Notre choix pour effectuer ce transfert de la langue naturelle vers un langage formel s'est porté vers le raisonnement analogique car, en tant qu'approche de raisonnement à base de cas, il permet de produire des réponses correctes avec seulement quelques exemples adéquats. D'autre part, le raisonnement à base de cas ne requiert pas non plus de lourd calcul afin d'intégrer de nouvelles connaissances. Ces deux raisons permettent de s'attendre à des performances correctes en situation d'apprentissage incrémental. À notre connaissance, cela n'a pas été évalué jusqu'à présent.

Comment se comporte le raisonnement par analogie dans le contexte de l'apprentissage incrémental ? De quelle manière la séquence d'apprentissage influence-t-elle la performance globale ? Nous proposons dans la suite une série d'expériences pour apporter des réponses à ces questions.

2 État de l'art

Une tâche d'apprentissage est dite incrémentale si *les exemples d'apprentissage utilisés pour la résoudre deviennent disponibles au cours du temps, habituellement un par un*, ainsi que le définit (Giraud-Carrier, 2000). L'auteur met en évidence que certaines catégories de tâches, telles que la conception d'agents intelligents, sont naturellement propices à l'emploi de l'apprentissage incrémental. Bien qu'il suscite un intérêt croissant, c'est encore un paradigme relativement marginal parmi les approches de l'apprentissage artificiel. Cependant, quelle que soit la quantité de données utilisées pour l'entraînement d'un système apprenant, une mise à jour de l'algorithme d'apprentissage (en dehors d'un simple ajustement de paramètres) requerra des données également mises à jour. Cela signifie la constitution d'un corpus contenant les propriétés devenues nécessaires. Cet aspect est particulièrement problématique dans le domaine des agents intelligents, où l'interaction avec l'humain contraint la quantité des données collectées. Sous cette perspective, l'apprentissage incrémental est plus réaliste que l'apprentissage hors ligne, car il n'implique pas l'hypothèse du monde clos.

L'incrémentalité soulève la question de l'ordre d'apprentissage des exemples. Même si le score final est similaire, des ordonnancements différents font varier les performances intermédiaires du système. Le comportement humain est difficilement prédictible, il a ici une influence directe sur l'ordonnancement. Dans le domaine de l'enseignement algorithmique, (Cakmak & Thomaz, 2014; Balbach & Zeugmann, 2009) tentent de guider l'enseignement de l'humain à la machine grâce à un ensemble d'instructions. L'objectif est d'amener l'humain à proposer des exemples utiles pour le système, qu'il ne proposerait pas naturellement lors d'un enseignement humain-humain. Dans le contexte du raisonnement par analogie, ce guidage ne semble pas nécessaire *a priori*. En effet, l'analogie est un mode de raisonnement habituel de la cognition humaine (Hofstadter & Sander, 2013), un choix d'ordonnancement naturel devrait donc fournir des exemples utiles aux moments où ils sont nécessaires. De plus, notre système interagit plutôt avec un utilisateur qu'avec un enseignant.

Bien que celui-ci soit amené à jouer ces deux rôles, il est préférable de limiter les contraintes imposées à l'utilisateur sur l'enseignement qu'il produit.

3 Raisonnement analogique

Cette section présente le modèle de raisonnement analogique formel ainsi que son application au transfert entre langage d'énoncés vus comme des séquences de tokens. Nous y décrivons également la génération automatique de nouveaux exemples par analogie à partir d'une base restreinte.

3.1 Définition d'une proportion analogique

Une proportion analogique est une relation 4-aire qui peut s'exprimer pour le quadruplet d'objets (x, y, z, t) : "x est à y ce que z est à t". On utilise la notation $[x : y :: z : t]$ pour une proportion analogique, et $[x : y :: z : ?]$ pour une équation analogique. La manipulation de problèmes analogiques peut faire intervenir deux opérations : la **vérification** qu'un quadruplet est en relation analogique, et la **résolution** d'une équation analogique.

Des algorithmes efficaces pour ces deux opérations à l'aide de la programmation dynamique sont décrit dans (Stroppa, 2005; Stroppa & Yvon, 2007). Leur complexité algorithmique est néanmoins en $\mathcal{O}(n^3)$, n étant le nombre d'entités, ou tokens, de la plus longue des quatre séquences. Leur application à l'apprentissage artificiel requiert de rechercher ou de résoudre ces analogies dans la base d'exemples, ce qui induit une boucle supplémentaire due à la recherche de triplets d'exemples dans la base. Étant donnée une base d'exemples B, le parcours naïf a une complexité en $\mathcal{O}(|B|^3)$. Cependant, des optimisations existent, comme l'indexation des exemples dans une structure d'arbre de comptage (Langlais & Yvon, 2008), ainsi que d'autres heuristiques (Lepage & Denoual, 2005). Dans le système utilisé pour nos expériences, nous avons utilisé l'arbre de comptage de Langlais et Yvon.

3.2 Le transfert de langage

Le raisonnement analogique est formalisé sur les monoïdes libres, il est donc applicable en particulier aux séquences munies de l'opération de concaténation. On peut modéliser de manière directe le transfert de langage comme un problème de transfert par analogie sur des séquences de tokens ou de caractères. La résolution d'une équation analogique sur des séquences est effectuée par identification d'un alignement entre des sous-séquences, appelées *facteurs*. Par exemple l'équation analogique ["abx" : "aby" : : "cx" : ?] peut donner lieu à l'alignement, ou *factorisation* : ("ab", "x") ; ("ab", "y") ; ("c", "x"). La factorisation permet de réduire le problème à un ensemble d'équations analogiques triviales de la forme $[x : x :: y : ?]$ ou $[x : y :: x : ?]$. La factorisation donnée en exemple permet d'obtenir les proportions suivantes : ["ab" : "ab" : : "c" : "**c**"] et ["x" : "y" : : "x" : "**y**"] ; par suite, la solution à l'équation initiale est la concaténation "cy".

Les expériences présentées ci-après travaillent à partir d'énoncés vus en tant que séquences de mots sans autre prétraitement, afin que le modèle reste indépendant du langage. La base d'exemple utilisée est composée de paires associant requêtes en langue naturelle à leur expression correspondante en langage de programmation : $B \subset (R \times C)$. Pour une requête $r \in R$ donnée, on parcourt la base d'exemples à la recherche de proportions analogiques afin de générer une commande c en réponse.

À cette fin, on utilise conjointement deux stratégies de recherche d'équations analogiques : directes de la forme $[r_i : c_i :: r : c]$, et indirectes à l'aide de proportions $[r_i : r_j :: r_k : r]$ associées aux équations $[c_i : c_j :: c_k : c]$. Chacune de ces stratégies permet la génération de commandes nouvelles, recombinant le vocabulaire présent dans la base et dans la requête d'entrée. Leur utilisation conjointe vient de leur complémentarité devant la constitution de la requête d'entrée (Letard *et al.*, 2015). Les exemples ci-dessous illustrent les résolutions directe et indirecte (les réponses produites sont en gras) :

supprime le fichier a.txt	:	`rm a.txt`
	::	
supprime le fichier racine	:	**`rm racine`**

compte les lignes de prog.c `wc -l prog.c`	:	compte les lignes du fichier C prog `wc -l prog.c`
	::	
compile prog.c `gcc prog.c`	:	compile le fichier C prog **`gcc prog.c`**

La première permet de gérer les paramètres présents dans la requête d'entrée mais absents de la base, ou plus exactement les tokens à la fois présents dans la requête et dans sa commande associée, sous réserve que la structure de la requête soit, elle, déjà présente dans la base. La seconde permet de gérer les structures nouvelles dans la formulation de la requête d'entrée, à condition que tout paramètre soit déjà présent dans la base.

Pour chaque équation analogique trouvée, l'algorithme de résolution renvoie un ensemble de solutions pour la commande C. Il contient généralement un grand nombre de solutions dont la plupart sont incohérentes, bien que toutes soient formellement valides (Somers *et al.*, 2009). Plusieurs stratégies ont été proposées pour sélectionner les solutions les plus naturelles. Nous choisissons ici un intermédiaire entre la stratégie simple du choix de la solution la plus occurrente (Stroppa & Yvon, 2006) et des solutions plus complexes sélectionnant potentiellement plusieurs bonnes solutions (Stroppa, 2005). Nous sélectionnons ici la solution utilisant le moins de *facteurs* dans l'équation analogique initiale [1]. Formellement, il s'agit de la solution issue du quadruplet analogique de plus petit *degré*. Le choix de cette stratégie permet d'éviter une énumération ou un échantillonnage potentiellement coûteux des solutions d'une équation analogique. Ces stratégies permettent d'atteindre une très bonne précision dans les réponses apportées par le système [2]. La performance globale est alors limitée par le taux de réponse du système.

3.3 Générer de la variation

Afin d'augmenter la variation interne à la base d'exemples, l'inférence analogique peut être utilisée pour générer des exemples par analogie d'autres préexistants, comme illustré par l'exemple suivant :

1. *"Envoie le dossier articles sur pc-maison"*
   → `scp -r articles/ pc-maison:`

1. La résolution analogique implique la segmentation des séquences en facteurs, puis l'alignement de ces facteurs. Les solutions les plus fréquentes produites sont généralement identiques à celles requérant le moins de facteurs. Les détails de l'algorithme peuvent être retrouvés dans (Langlais & Yvon, 2014; Stroppa & Yvon, 2007)

2. Référence à un travail antérieur anonymisée pour la rélecture. Elle sera ajoutée pour la version finale.

2. *"Envoie le dossier article sur mon ordinateur personnel"*
   → `scp -r articles/ pc-maison:`
3. *"Vérifie l'espace disque sur pc-maison"*
   → `ssh pc-maison df`
4. *"Vérifie l'espace disque sur mon ordinateur personnel"*
   → `ssh pc-maison df`

Dans cet exemple, les trois premières requêtes sont choisies dans la base pour former une équation analogique qui permet de générer la quatrième en solution. À leur tour, les commandes associées sont également mises en équation. Si cette dernière équation permet de produire une commande solution, la requête et la commande produites sont alors associées et ajoutées à la base.

Cette méthode permet de générer un grand nombre d'exemples comportant beaucoup de variations analogiques de la base initiale. Elle introduit également beaucoup de bruit, car une part importante des paires générées sont grammaticalement incorrectes ou même incohérentes pour un lecteur humain, comme par exemple : "*S'il-te-plaît, supprime le fichier le fichier f.pdf s'il-te-plaît*". Cependant, la présence de ces exemples dans la base générée n'a qu'une influence minime sur la qualité des réponses du système car d'une part, on rappelle que ces réponses ne sont constituées que de commandes en langage formel, l'utilisateur n'est donc pas amené à lire les requêtes générées ; et d'autre part, des requêtes dégénérées telles que l'exemple donné n'entreront que rarement dans la composition d'équations analogiques valides. En effet, l'utilisateur pour sa part soumettra de fait uniquement des requêtes qui font sens pour lui, et pour être valide, une proportion analogique doit satisfaire la propriété de comptage sur les tokens (Stroppa & Yvon, 2005) donnée par :

$$[A : B :: C : D] \Rightarrow \forall t \in \mathcal{L}\ |A|_t + |B|_t = |C|_t + |D|_t$$

$\mathcal{L}$ désigne le lexique des tokens de la base, et $|X|_t$ est le nombre d'occurrences du token t dans la séquence X. La recherche dans la base à l'aide de la structure d'arbre de comptage ne sélectionne que les quadruplets vérifiant cette propriété. Dans le cas des équations analogiques, ne faisant intervenir que des triplets, la propriété se traduit par l'inégalité suivante :

$$\exists\, x\, [A : B :: C : x] \Rightarrow \forall t \in \mathcal{L}\ |A|_t + |B|_t \geq |C|_t$$

Enfin, dans le cas où une requête dégénérée est effectivement employée pour produire une solution, il est probable que la requête qui lui est associée soit malgré tout correcte : la variation au niveau du langage formel est beaucoup plus faible que celle au sein de la langue naturelle, dans la plupart des cas, l'opération de génération d'exemple ne fait que recopier la requête comme en atteste le quadruplet donné plus haut.

De manière plus pratique, le nombre d'exemples générés se révèle trop grand pour qu'une annotation manuelle de la validité des associations ou de la correction grammaticale soit réalisable (voir la description du corpus section 4 ci-après).

3.4 Adaptation à l'incrémentalité

À notre connaissance, le raisonnement analogique formel n'a pas été utilisé dans le contexte de l'apprentissage incrémental jusqu'à présent. Pourtant, l'emploi du raisonnement analogique pour l'apprentissage artificiel est incrémental. En effet, les connaissances de la base d'exemple ne subissent pas de compilation ou pré-calcul coûteux, mis à part la phase d'indexation pour les arbres de comptage

qui a une complexité logarithmique. Chaque nouvel exemple peut alors être directement inclus dans la base pour être réutilisé dans des proportions analogiques dès l'itération suivante.

4 Évaluation expérimentale

	associations	commandes	cohortes
QUOTIDIEN	512	21	20
génération	54243	48	-
NOUVEAU	77	20	20
génération	1205	23	-
totaux génération	73628	230	-

TABLE 1 – Composition des bases d'exemples

Les nombres sous *associations* donnent le nombre de paires requête-commande de la base. Les lignes *génération* de chaque base donnent le nombre d'éléments uniques et nouveaux. Le comptage des commandes tient compte de toute variation pour différencier deux commandes. Données originales disponibles à http://perso.limsi.fr/letard/ilar/taln2016.zip

Dans cette section, nous étudions, qualitativement principalement, les performances du système d'apprentissage incrémental fondé sur le raisonnement analogique. Nous avons dressé une liste de conditions expérimentales pertinentes pour l'utilisation du système et pour mettre en évidence l'influence de l'apprentissage incrémental. Il s'agit de : l'ordre d'apprentissage des exemples, le nombre d'exemples appris, le changement d'utilisateur et l'utilisation de la génération analogique d'exemples. Les protocoles de test correspondants avec les résultats d'expériences obtenus sont décrits dans les sous-sections qui suivent.

Le tableau 1 résume la composition des bases d'exemples utilisées. La base initiale représente un ensemble d'exemples qui a pu être accumulé auprès d'un groupe d'utilisateurs réguliers, on y réfèrera sous l'expression "base QUOTIDIEN". Elle a été collectée auprès d'un petit groupe de participants en leur remettant la liste des vingt commandes ; la consigne était, pour chaque commande, de rédiger plusieurs formulations de requêtes pour demander naturellement à une personne d'effectuer la commande. Aucune contrainte n'a été donnée concernant ces formulations afin de refléter le plus possible une interaction naturelle. La plupart sont naturellement à l'impératif mais beaucoup aussi sont sous forme interrogative. Quelques unes comprennent des erreurs grammaticales. Les exemples ci-dessous illustrent la forme des associations recueillies :

Crée une copie de foo appelée bar → `cp foo bar`
Va dans dir et récupère les derniers commits → `cd dir; git pull`
Dans foo, peux-tu me dire combien y a de mots ? → `wc -l foo`

La seconde base contient un nombre plus restreint d'exemples que l'on peut attribuer à un nouvel utilisateur du système, susceptible de s'exprimer différemment de ce qui a été appris dans la base QUOTIDIEN. Cette seconde base, nommée "base NOUVEAU" dans la suite, a effectivement été collectée auprès d'un volontaire indépendant de la première. La liste de commandes était identique, mais la consigne comprenait l'instruction supplémentaire de faire varier systématiquement les paramètres des commandes (noms de fichiers, nombre d'éléments, chemins d'accès...). De plus, contrairement à la première collecte, aucun exemple d'association n'a été fourni afin de limiter autant que possible

l'influence sur les formulations du participant. Les associations qu'elle contient concernent cependant le même ensemble de commandes. Le tableau décrit également le nombre d'exemples générés pour chacune des deux bases par la méthode présentée précédemment, en tentant de résoudre les équations analogiques formées par chaque triplet constitué à partir de la base.

La base QUOTIDIEN a été disposée selon différents ordonnancement dans les tests :
— groupée par cohortes
— groupée par cohortes en ordre inverse
— mélangée

Une cohorte est définie par l'ensemble des exemples qui concernent le même type de commande. Par exemple "*Efface le fichier foo.odt*" (`rm foo.odt`) et "*Supprime récursivement le répertoire bar/*" (`rm -R bar/`) appartiennent à la même cohorte, alors que "*Efface la variable $VAR*" (`unset $VAR`) appartient à une cohorte distincte des précédents exemples.

Par souci de lisibilité des graphes présentés dans la suite, la performance de l'apprentissage est représentée par le nombre *cumulé* de bonnes réponses fournies pendant la progression incrémentale. Les caractéristiques intéressantes à relever sont donc la valeur finale ainsi que la forme des courbes. L'aire sous les courbes n'est pas ici une propriété pertinente pour l'interprétation. Un apprentissage idéal produira une courbe ayant une dérivée tendant rapidement vers 1. La droite d'équation $y = x$ a été ajoutée pour comparaison, elle représente la limite supérieure stricte pouvant être atteinte par un système apprenant incrémentalement. En effet, elle correspond à un système donnant exactement une bonne réponse pour chacune des requêtes qui lui sont soumises. En d'autre termes, il s'agit d'un système qui possède déjà toutes les connaissances nécessaires pour répondre à chaque requête, ce qui est donc impossible en commançant avec une base d'exemple vide.

4.1 Influence de l'ordre d'apprentissage

Dans le but d'analyser l'effet de l'ordonnancement des exemples sur la performance, le système a été testé sur la base QUOTIDIEN de manière incrémentale. Pour chaque itération i, la i-ème requête est soumise au système, la réponse comparée à la commande attendue, puis la requête associée à sa commande est ajoutée à la base d'exemples du système. Cet ajout systématique peut être vu en contexte interactif comme un processus de validation : le système demande à l'utilisateur si sa proposition est correcte, dans la positive l'exemple est enregistré, dans la négative le système demande à l'utilisateur de fournir la commande correcte qui aurait dû être proposée et l'association est ajoutée.

La figure 1 montre les performances du système selon l'ordre des requêtes d'entrée. On remarque que sur les 512 requêtes de la base, le système produit en condition incrémentale un peu plus de 50% de réponses correctes, bien que le test sur la base mélangée ait environ 15 réponses de retard sur les autres. Malgré la proximité des scores finaux, la proportion de requêtes correctement répondues commune entre les trois tests ne dépasse pas 48%. En effet, la première requête d'une cohorte à être soumise au système ne peut jamais avoir de réponse correcte, et celle-ci n'est jamais la même pour les trois ordonnancement. La stabilité des résultats peut s'expliquer *a priori* par la propriété de symétrie de l'analogie : si le système est capable de répondre correctement à une requête D ayant vu A, B et C, alors il est également capable de répondre correctement à A étant donné B, C et D. Cette symétrie ne s'applique pas à toutes les permutations mais suis la règle ci-dessous, applicable transitivement :

$$[A : B :: C : D] \Leftrightarrow [A : C :: B : D] \Leftrightarrow [C : A :: D : B]$$

Cependant, cette symétrie ne rend les résultats finaux similaires qu'en supposant que toutes les

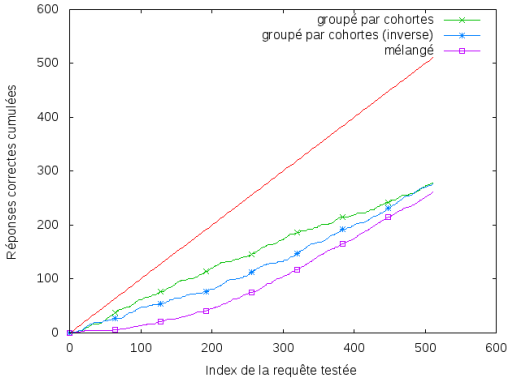

FIGURE 1 – Effet de l'ordre des exemples sur les performances de l'apprentissage incrémental sur la base QUOTIDIEN

proportions de la base sont disjointes or ce n'est pas le cas général, comme discuté plus loin.

La figure 1 montre que l'ordonnancement des exemples a un effet sur la performance. Grouper les exemples donne un meilleur score final et donne des réponses correctes plus régulièrement qu'utiliser la base mélangée. Ce dernier aspect s'explique naturellement par le fait que l'ensemble des connaissances disponibles pour chaque cohorte est acquise séquentiellement, ce qui conduit à produire les bonne réponses qu'elles permettent de trouver immédiatement à l'issue de cette phase locale d'apprentissage. Néanmoins, la courbe donnée par la base mélangée est plus lissée et sa dérivée est toujours croissante. Ce sont des propriétés importantes pour la poursuite de l'apprentissage au delà de ces 512 exemples. Les coefficients directeurs moyens de chacune des courbes sur les 200 dernières requêtes sont, dans l'ordre de la figure, $0,47$, $0,68$ et $0,74$. Si l'exécution se poursuit sur de nouvelles requêtes avec les mêmes taux de réponse, on s'attend à voir la courbe de l'ordonnancement aléatoire dépasser les deux premières. Nous allons à présent analyser de manière plus détaillée les effets de l'ordonnancement en faisant varier le nombre d'exemples appris.

4.2 Arrêt subi de l'apprentissage

Le comportement du système montré au cours des tests avec différents ordonnancements peut aussi être influencé par le nombre d'exemples soumis au total. Même en condition d'apprentissage incrémental, il pourrait être nécessaire d'interrompre l'apprentissage pendant une certaine période, par exemple pour interagir avec un utilisateur final qui n'a pas les connaissances lui permettant d'enseigner les bonnes réponses au système quand une erreur est commise. Nous avons évalué la performance du système en interrompant artificiellement l'incrément de la base à partir de différents états de la base. Les résultats sont donnés par les figures 2 et 3. Elles présentent dix tests différents effectués avec un arrêt de l'incrément aux indices 50, 100, 150 ... jusqu'à 500, ainsi que le test initial (courbe colorée/ponctuée) sans interruption. On peut également assimiler l'arrêt de l'incrément à l'indice n comme une simulation de l'apprentissage hors ligne avec une base initiale composée des $n-1$ premiers exemples.

On note immédiatement une grande différence entre la courbe obtenue sur la base groupée par cohortes (figure 2) et celle obtenue sur la base mélangée (figure 3). En groupant par cohortes, presque

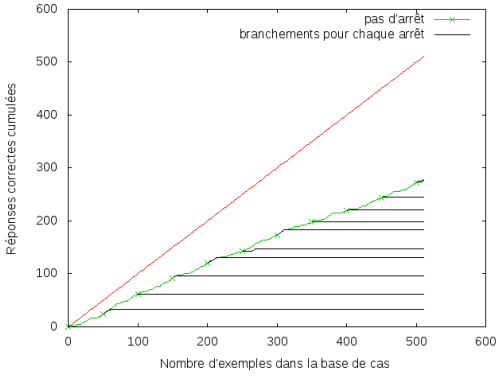

FIGURE 2 – Comportement après interruption de l'apprentissage pour une base groupée par cohortes

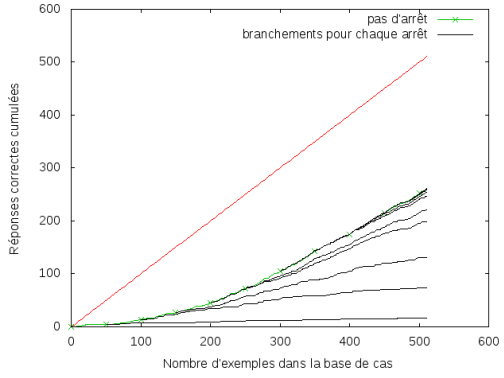

FIGURE 3 – Comportement après interruption de l'apprentissage pour une base mélangée

aucune réponse correcte n'est donnée par le système après l'arrêt de l'incrément, tandis qu'avec la base mélangée, le système continue à donner des bonnes réponses à un rythme moins élevé dépendant de la taille de la base. Cela peut s'interpréter facilement par le fait que les connaissances sont principalement réutilisées localement lorsque la base est groupée, alors qu'elles tendent à être distribuées de manière équiprobable sur l'ensemble de la base dans le cas du mélange.

Ce résultat montre que le score final obtenu sur la base groupée est fortement dépendant de l'apprentissage incrémental. En effet, un apprentissage hors ligne sur des préfixes de la même séquence d'exemples est incapable de généraliser pour donner des performances comparables. D'autre part, cette expérience montre que la condition dans laquelle la base d'exemple est mélangée rend le système beaucoup plus robuste aux interruptions d'apprentissage. En effet, la variété des exemples permet d'acquérir une densité de connaissances plus importante au cours des premières itérations, par opposition à une densité peu variable et plutôt faible dans le cas des exemples groupés. En outre, cette stabilité est bienvenue puisque c'est la situation de l'ordre pseudo-aléatoire des requêtes qu'on peut s'attendre à rencontrer dans un cas réel d'interaction avec des utilisateurs. Enfin, étant donné que le système est multi-utilisateurs, ces résultats suggèrent qu'il est plus intéressant de recevoir peu d'exemples par utilisateur d'un grand nombre d'utilisateurs plutôt qu'un grand nombre d'exemples de seulement quelques utilisateurs. Ce qui soulève la question : comment l'apprentissage incrémental influence-t-il le comportement du système dans le cas d'un utilisateur nouveau ?

4.3 Introduction d'un utilisateur nouveau

Les requêtes soumises au système sont formulées en langue naturelle, connue pour le grand nombre de variations de surface qu'elle permet pour un même contenu sémantique. On peut donc s'attendre à ce que différents utilisateurs du système aient des styles différents pour la rédaction de leurs instructions pour le système. D'autre part, ils ont certainement des besoins différents en termes de paramètres des commandes. Nous avons évalué la capacité de généralisation du système avec et sans incrément, en soumettant au système les requêtes de la base NOUVEAU après avoir ajouté les exemples de la base QUOTIDIEN à ses connaissances. Les résultats de cette expérience sont décrits par la figure 4. Les

performances obtenues précédemment sur la base mélangée (avec incrément), tronquées à 77 requêtes, ont été ajoutés pour comparaison. Les scores finaux obtenus confirment l'utilité de l'apprentissage incrémental : il permet de dépasser l'apprentissage hors ligne de 60%. Le nombre total de réponses correctes est de 16 sur 77 avec incrément, 10 sur 77 sans. Le test précédent sur la base QUOTIDIEN donne seulement 8 réponses correctes sur les 77 premières requêtes, en partant d'une base d'exemples vide. Les coefficients directeurs approchés des courbes avec et sans incrément sont respectivement de $0,35$ et $0,24$ à la fin du test entre les abscisses 60 et 77. Le coefficient pour le test initial sur la base QUOTIDIEN vaut quant à lui $0,18$ sur les mêmes abscisses. On s'attend à ce que le comportement du système avec les requêtes du nouvel utilisateur soit similaire à ce qu'il était avec les requêtes de la base QUOTIDIEN. En effet, bien que différant par leurs styles de rédaction des requêtes, il n'y a pas de raison *a priori* pour que les utilisateurs induisent des performances très différentes pour chacun. Ceci suppose bien entendu que leurs besoins en termes de types de requêtes sont similaires, hypothèse que nous posons ici, mais qui dépend des communautés d'utilisateurs.

D'autre part, on peut considérer l'utilisation de la génération d'exemples par analogie, telle que mentionnée en section 3.3, afin d'augmenter la variation parmi les exemples disponibles.

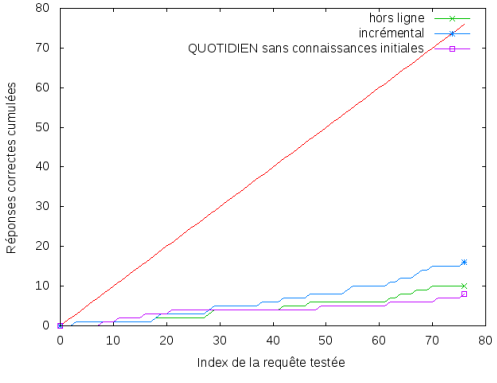

FIGURE 4 – Performances avec et sans incrément pour un test sur la base NOUVEAU, avec la base QUOTIDIEN pour connaissances initiales.
La courbe obtenue précédemment pour le test sur QUOTIDIEN sans connaissances initiales a été ajoutée pour comparaison.

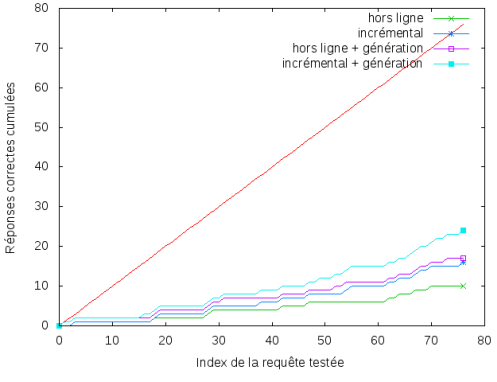

FIGURE 5 – Performances comparées avec et sans utilisation de l'incrément et de la génération sur la base NOUVEAU, avec la base QUOTIDIEN pour connaissances initiales.

4.4 Enrichir automatiquement la base d'exemples

La génération par analogie de nouveaux exemples à partir de la base devrait permettre d'améliorer la performance du système, nous analysons dans cette section son influence relative par rapport à l'apprentissage incrémental. Appliquer la méthode de génération de la section 3.3 dans le contexte de l'apprentissage incrémental revient à effectuer une opération de génération quadratique pour chaque nouvel exemple. Cette opération consiste à former tous les triplets non redondants comprenant le nouvel exemple et deux exemples de la base existante (bases QUOTIDIEN et NOUVEAU confondues, le cas échéant), ce qui correspond à $3 \times |B|^2$ équations analogiques à résoudre. Quand il existe des

solutions, elles sont alors ajoutées à la base. D'après le nombre total d'exemples générés donné plus haut dans le tableau 1, le nombre moyen d'ajouts à la base pour chaque nouvel exemple est de 126, bien qu'il soit généralement croissant et nul pour les deux premiers exemples. On notera qu'une seule étape de génération est appliquée ; les exemples générés n'ont pas été réutilisés dans les générations ultérieures. L'explosion combinatoire du nombre d'exemples générés que cela impliquerait rendrait rapidement les énumérations trop coûteuses.

La figure 5 présente les performances comparées sur la base NOUVEAU avec et sans génération et avec ou sans incrément. La base de connaissances contient initialement l'ensemble des exemples de la base QUOTIDIEN, et les exemples générés associés le cas échéant. L'usage de l'incrément seul ou de la génération seule produisent des résultats similaires. Cependant, plusieurs facteurs influencent la performance obtenue en utilisant la génération, tels que la taille de la base QUOTIDIEN ou la proximité syntaxique entre les requêtes des deux bases, ou plus précisément le nombre de proportions analogiques que l'on peut former entre des requêtes des deux bases. Ces facteurs sont relativement indépendants de l'utilisation ou non de l'incrément, il est donc possible que la proximité des deux résultats ne soit que fortuite. On peut remarquer d'autre part que la génération sans incrément donne de meilleurs résultats que l'apprentissage hors ligne seul, ce qui montre que les connaissances nécessaires pour la production de réponses correctes pour une part des requêtes de la base NOUVEAU étaient déjà contenues implicitement dans la base QUOTIDIEN. La génération par analogie a permis de rendre ces connaissances explicites.

La combinaison de la génération et de l'incrément donne de meilleurs résultats que chacun pris séparément, cela indique que ces approches sont au moins en partie complémentaires. D'une part, il n'est pas vraiment surprenant que la génération ne permette pas de produire toutes les réponses correctes données grâce à l'apprentissage incrémental, mais d'autre part, cela signifie que même en ajoutant un à un tous les exemples rencontrés, certaines solutions ne restent accessibles qu'après une étape de génération analogique.

Malgré les 31% du score final combiné, on peut souligner qu'il représente déjà plus de deux fois le score de base de l'apprentissage hors ligne sans génération. Ce ratio prometteur doit cependant être considéré avec prudence à cause de la petite taille de l'ensemble de test.

	avec génération	sans
groupée par cohortes	61,8%	54,2%
groupée par cohortes (ordre inverse)	61,4%	53,9%
mélangée	60,3%	50,9%

TABLE 2 – Performances finales avec incrément sur la base QUOTIDIEN à partir d'une base vide

Le tableau 2 reporte les résultats finaux comparés, sans (*idem* figure 1) et avec génération de variation. Tous sont obtenus en utilisant l'incrément et avec une base d'exemples initiale vide. On y retrouve également la complémentarité entre génération par analogie et incrément.

5 Discussion

L'utilisation du raisonnement analogique dans le contexte de l'apprentissage incrémental permet, sans connaissance initiale, de dépasser les 50% de bonnes réponses avec seulement 512 exemples. La pente

moyenne sur les 200 dernières requêtes pour la base mélangée atteint $0,74$, ce qui correspond donc à un taux de bonnes réponses de 74% sur ce segment. Alors que les tests sur la base groupée voient leur taux moyen (dérivée des courbes présentées) diminuer ou stagner, le taux de bonnes réponses dans le cas réel d'une base non ordonnée est en constante augmentation. Malgré cela, la performance finale est meilleure pour la base groupée. La raison de cette différence n'est pas triviale. En effet, d'une part, la symétrie de l'analogie permet, comme discuté plus haut, de toujours résoudre au moins une requête parmi les quatre d'une proportion valide dans la base. Mais d'autre part, les proportions entre exemples de la base sont nombreuses et comptent des intersections, des exemples peuvent ainsi jouer dans plusieurs résolutions analogiques. On peut définir un ordre sur les exemples de la base grâce au nombre de proportions dans lesquelles chacun joue un rôle. Un exemple jouant dans plus de proportions qu'un autre sera dit plus informatif. Ainsi, l'ordonnancement optimal d'une base donnée consiste à de placer les exemples les plus informatifs en premier.

D'autre part, les résultats des dernières expériences (figures 4 et 5) rappellent qu'un nombre minimal d'exemples pour chaque cohorte est nécessaire avant d'observer une augmentation sensible du score. La courbe sur la base non ordonnée de la figure 1 semble atteindre sa dérivée finale entre la requête 200 et la requête 250, ce qui correspond à avoir vu en moyenne entre 10 et $12,5$ exemples par cohorte (20 au total). Cela explique pourquoi les résultats sur les 77 requêtes de la base NOUVEAU n'atteignent pas un taux de bonnes réponses similaire.

6 Conclusion et perspectives

Dans cet article, nous avons étudié l'adéquation de l'apprentissage incrémental avec le raisonnement par analogies formelles au travers de quatre expériences principales. Nous avons montré que l'utilisation de l'apprentissage incrémental rend le système robuste aux variations sur l'ordonnancement des exemples d'entrée. L'ordonnancement pseudo-aléatoire attendu dans le cas réel donne des résultats un peu moins tôt que sur une base groupée, mais plus stables. Nous avons également observé que l'incrément a un effet non négligeable sur la performance du système face à un nouvel utilisateur. Enfin, la combinaison de l'apprentissage incrémental et de la génération analogique de variation dans les connaissances permet d'augmenter sensiblement les résultats face à un nouvel utilisateur, et d'atteindre 60% de bonnes réponses sur 512 requêtes sans connaissance préalable.

Ce résultat prometteur ne doit pas faire perdre de vue l'influence potentielle du bruit sur des tests avec une base d'exemples de cette taille. Il serait très intéressant à court terme de comparer les résultats avec des données similaires sur d'autres langages et en augmentant le nombre de commandes différentes. À moyen terme, une comparaison avec d'autres approches sera informative également. Certains travaux proposent d'appliquer l'apprentissage incrémental à des méthodes statistiques (Polikar *et al.*, 2001; Poggio & Cauwenberghs, 2001), mais ils concernent des problèmes de classification. Il s'agirait d'étudier si leur adaptation au problème de transfert de langage est efficace, et si elle est comparable à l'approche analogique, notamment pour une petite quantité de données.

D'autre part, la génération analogique de variations à partir de la base d'exemples, qui permet une amélioration des résultats, reste une technique très coûteuse car elle a une complexité cubique dans la taille de la base, ce qui rend impossible son application pour plus d'une itération. Des pistes pour limiter ce coût computationnel sont la recherche d'une heuristique efficace pour élaguer ce vaste espace de recherche, ou encore de réduire fortement le coût de l'opération unitaire de la génération : la résolution d'équations analogiques.

Références

BALBACH F. J. & ZEUGMANN T. (2009). Recent developments in algorithmic teaching. In *Language and Automata Theory and Applications*, p. 1–18. Springer.

CAKMAK M. & THOMAZ A. L. (2014). Eliciting good teaching from humans for machine learners. *Artificial Intelligence*, **217**, 198–215.

GIRAUD-CARRIER C. (2000). A note on the utility of incremental learning. *AI Communications*, **13**(4), 215–223.

HOFSTADTER D. R. & SANDER E. (2013). *L'analogie au coeur de la pensée*. Odile Jacob.

LANGLAIS P. & YVON F. (2008). Scaling up analogical learning. In *COLING (Posters)*, p. 51–54.

LANGLAIS P. & YVON F. (2014). Issues in analogical inference over sequences of symbols : A case study on proper name transliteration. In H. PRADE & G. RICHARD, Eds., *Computational Approaches to Analogical Reasoning : Current Trends*, p. 59–82. Springer-Verlag Berlin Heidelberg.

LEPAGE Y. & DENOUAL E. (2005). Purest ever example-based machine translation : Detailed presentation and assessment. *Machine Translation*, **19**(3-4), 251–282.

LETARD V., ILLOUZ G. & ROSSET S. (2015). Analogical reasoning for natural to formal language transfer. In *ICTAI*, Vietri sul Mare, Italy.

POGGIO T. & CAUWENBERGHS G. (2001). Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, **13**, 409.

POLIKAR R., UPDA L., UPDA S. S. & HONAVAR V. (2001). Learn++ : An incremental learning algorithm for supervised neural networks. *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, **31**(4), 497–508.

SOMERS H., DANDAPAT S. & NASKAR S. K. (2009). A review of ebmt using proportional analogies. In *3rd International Workshop on Example-Based Machine Translation*.

STROPPA N. (2005). *Analogy-Based Models for Natural Language Learning*. Thèse, Télécom ParisTech.

STROPPA N. & YVON F. (2005). *Analogical learning and formal proportions : Definitions and methodological issues*. Rapport interne, ENST Paris.

STROPPA N. & YVON F. (2006). Du quatrième de proportion comme principe inductif : une proposition et son application à l'apprentissage de la morphologie. *Traitement automatique des langues*, **47**(1).

STROPPA N. & YVON F. (2007). Formal models of analogical proportions. hal-00145148.