

Un outil générique pour l'analyse automatique et la visualisation de productions d'apprenants

Fabien Delorme

Gaëlle Loosli

Laboratoire LITIS – département ASI
INSA de Rouen

Avenue de l'Université - BP 8

76801 Saint-Étienne-du-Rouvray Cedex

{fabien.delorme, gaelle.loosli}@insa-rouen.fr

Résumé

L'analyse automatique de productions d'apprenants est un problème central dans le cadre des EIAH et plus particulièrement des STI (Systèmes Tutoriels Intelligents). Nous proposons dans cet article un outil générique d'analyse assistée de productions d'apprenants. Cet outil doit permettre à un concepteur d'EIAH d'instancier facilement un module d'évaluation dans le cadre d'un EIAH particulier. L'outil instancié permettra alors une analyse automatique de la production de l'apprenant, tout en permettant au tuteur d'avoir une vision globale du niveau des apprenants participant à la formation. L'article présente en détail la démarche mise en œuvre ainsi que quelques exemples d'utilisation de l'outil.

Mots-clés

analyse automatique, évaluation de l'apprenant, algorithme des k Plus Proches Voisins, algorithme du Stochastic Neighbor Embedding.

Abstract

L'analyse automatique de productions d'apprenants est un problème central dans le cadre des EIAH et plus particulièrement des STI (Systèmes Tutoriels Intelligents). Nous proposons dans cet article un outil générique d'analyse assistée de productions d'apprenants. Cet outil doit permettre à un concepteur d'EIAH d'instancier facilement un module d'évaluation dans le cadre d'un EIAH particulier. L'outil instancié permettra alors une analyse automatique de la production de l'apprenant, tout en permettant au tuteur d'avoir une vision globale du niveau des apprenants participant à la formation. L'article présente en détail la démarche mise en œuvre ainsi que quelques exemples d'utilisation de l'outil.

Keywords

automated analysis, learner evaluation, k nearest neighbors algorithm, Stochastic Neighbor Embedding algorithm.

1 Introduction

L'analyse automatique de productions d'apprenants est un problème important dans le cadre des EIAH (Environnements Informatiques pour l'Apprentissage Humain) et plus particulièrement dans celui des STI (Systèmes Tutoriels Intelligents). Une telle analyse peut remplir plusieurs objectifs : adapter le contenu de la formation en fonction des connaissances de l'apprenant [?], valider ou non la maîtrise d'une notion — ou parle alors d'évaluation sommative —, dresser un diagnostic détaillé des connaissances d'un apprenant [?], constituer des groupes, signaler au tuteur des individus en difficulté, déceler les conceptions des apprenants [?], etc. Néanmoins cette analyse pose de nombreux problèmes pratiques et le développement d'un module d'analyse automatique est en général délicat et coûteux en temps. Un outil générique d'analyse permettrait, une fois instancié dans un cadre particulier, un gain de temps important lors du développement d'EIAH dans lesquels il pourrait s'appliquer.

Nous proposons dans cet article le prototype d'un tel outil générique d'analyse assistée de productions d'apprenants. Cet outil doit permettre à un concepteur d'EIAH d'instancier facilement un module d'évaluation dans le cadre d'un EIAH particulier. L'outil instancié permettra alors une analyse automatique de la production de l'apprenant, tout en permettant au tuteur d'avoir une vision globale du niveau des apprenants participant à la formation. Cet outil repose sur l'utilisation de techniques issues de l'apprentissage automatique, et plus particulièrement sur l'utilisation de l'algorithme des k Plus Proches Voisins [?] et de celui du Stochastic Neighbor Embedding [?].

Ce prototype a été instancié et utilisé concrètement dans le cadre de deux EIAH différents, l'un étant issu de nos travaux précédents et s'intéressant à l'évaluation des connaissances conceptuelles des apprenants, l'autre reposant sur le diagnostic des conceptions d'apprenants dans le cadre de la géométrie.

Cet article est constitué de trois parties. Après avoir présenté précisément les contextes dans lesquels notre technique peut être utilisée, nous détaillerons la solution que nous proposons ainsi que les techniques utilisées. Enfin, l'utilisation de cette solution dans le contexte que nous proposons sera abordée.

2 Contexte de Notre Étude : Quelques Exemples

Dans cette section, nous présentons quelques situations concrètes où s'est manifesté le besoin d'un outil d'analyse automatique de productions d'apprenants. La première concerne nos travaux précédents qui s'intéressaient à l'évaluation des connaissances déclaratives chez les apprenants. La seconde concerne l'évaluation des conceptions chez les apprenants dans le domaine de géométrie et plus particulièrement de la symétrie. Enfin, la dernière évoque l'utilisation des cartes conceptuelles pour évaluer l'apprenant.

2.1 Évaluation des connaissances déclaratives des apprenants

Au cours de ses travaux, N. Delestre a conçu un système nommé MÉTADYNE [?]. Cet outil permet à un enseignant, quel que soit son domaine, de produire un hypermédia adaptatif dédié à l'enseignement d'une ou plusieurs notions.

Un outil d'évaluation basé sur l'utilisation de cartes conceptuelles et sur une méthodologie issue des travaux de B.-M. Barth [?] a été conçu afin d'offrir un diagnostic des connaissances de l'apprenant suffisamment riche pour permettre une réelle adaptation de l'hypermédia au cas par cas [?].

Lors de la phase d'évaluation, l'apprenant est amené à dresser des cartes conceptuelles, chacune définissant une notion importante du cours. Ces cartes sont alors comparées aux cartes de référence produites au préalable par l'enseignant. En fonction du résultat de la comparaison, un certain nombre de cas peuvent être examinés :

- l'apprenant a confondu deux notions,
- il a dressé exactement la définition de référence,
- sa définition ressemble à celle de l'enseignant mais n'est pas exactement identique,
- etc.

Dans chacun de ces cas, un schéma d'adaptation spécifique pourra alors être appliqué et le contenu de l'hypermédia sera adapté en conséquence.

L'outil d'évaluation utilisé dans la nouvelle version de MÉTADYNE a donc pour objectif de classer les productions des apprenants dans une des catégories évoquées ci-dessus. Cet outil est proche de celui que nous proposons dans cet article, qui n'en est qu'une amélioration, comme nous le préciserons à la section 4.

2.2 Évaluation des conceptions des apprenants en géométrie

Dans [?], C. Webber propose un environnement permettant d'évaluer les conceptions d'élèves en mathématiques, concernant la notion de symétrie. Ces travaux reposent sur une analyse didactique poussée et sur un contexte d'enseignement très particulier. En ce sens, il sont le pendant des travaux présentés précédemment.

Le principe de cet environnement est de diagnostiquer les conceptions des apprenants dans le domaine de la géométrie, et plus particulièrement dans le domaine de la symétrie axiale. Pour cela, un problème est posé à l'apprenant. L'analyse du raisonnement mis en œuvre par l'apprenant permet d'identifier un certain nombre d'observables. Ces observables sont autant d'indices permettant de déceler la conception sous-jacente chez l'apprenant. Chaque observable peut en effet être le signe d'une seule ou de plusieurs conceptions différentes.

La solution proposée par C. Webber repose sur l'utilisation d'un système multi-agents. Chaque observable est associé à un agent. Lorsqu'un observable est détecté dans la solution de l'apprenant, l'agent correspondant est activé. Les agents activés votent alors afin de déterminer la conception mobilisée par l'apprenant.

Les diagnostics proposés par le système à partir d'un corpus de productions d'apprenants ont été comparés aux analyses de trois équipes de didacticiens et semblent donner de bons résultats.

2.3 Évaluation de cartes conceptuelles libres

La carte conceptuelle [?] est un outil d'évaluation très souple, qui peut être utilisé dans de nombreux champs disciplinaires. On définira une carte conceptuelle comme un ensemble de concepts étiquetés et reliés à l'aide de liens également étiquetés permettant de former une proposition.

Il à noter que les cartes conceptuelles peuvent être utilisées comme un outil très fermé, comme nous le proposons dans le système MÉTADYNE, ou très ouvert, comme le proposent de nombreux travaux. Dans ce cas-là, l'apprenant est libre de nommer les concepts et liens de sa carte à sa guise. La taille de la carte, c'est-à-dire le nombre de concepts et de liens pouvant y apparaître, n'est pas non plus limitée, de même que la forme même de cette carte. De telles cartes conceptuelles permettent d'évaluer des compétences de haut niveau et peuvent donner accès aux conceptions des apprenants.

Bien entendu, de telles cartes conceptuelles sont difficiles à analyser automatiquement car elles nécessitent de recourir à des techniques de traitement automatique de la langue. T. Conlon propose néanmoins une technique intéressante consistant à faire participer l'apprenant à l'analyse de sa propre carte [?]. Cette technique semble donner de bons résultats puisqu'elle permet d'attribuer à la carte d'un apprenant un score proche de celui que donnerait un évaluateur humain. Néanmoins, la technique proposée se limite à donner un score à l'apprenant. Une évaluation formative de la carte serait également intéressante.

3 Vers un Outil Générique pour l'Analyse Assistée

Cette section présente l'outil permettant une analyse assistée des productions d'apprenants que nous proposons. Le problème qui se pose à nous est d'évaluer l'appartenance d'une carte conceptuelle à une classe à partir de peu d'exemples. La plupart des méthodes de classification ont besoin de beaucoup d'exemples pour fonctionner et cela nous a menés à considérer l'algorithme des k Plus Proches Voisins qui n'a pas cette exigence. Par ailleurs, il est connu dans le domaine de l'apprentissage automatique qu'une bonne représentation des données est essentielle à une bonne classification. La représentation de cartes conceptuelles est potentiellement de très grande dimension même si la carte contient peu d'éléments. Cela incite à essayer de réduire l'espace de représentation. Les deux effets recherchés sont d'une part une facilitation de la classification et d'autre part une possibilité de visualisation des données. Après avoir présenté les techniques sur lesquelles nous nous reposons, nous présentons leur utilisation conjointe pour résoudre le double problème de la classification automatique des productions et de la visualisation des données.

3.1 Classification supervisée avec l'algorithme des k Plus Proches Voisins

L'algorithme de recherche des k Plus Proches Voisins [?] est un outil classique de classification qui consiste à classer un point p à l'aide de ses k voisins les plus proches. On procède à un vote parmi les voisins et la classe la plus représentée est attribuée au point considéré. Pour déterminer quels sont les points les plus proches, un simple calcul de distance euclidienne est suffisant. Si p est trop éloigné de tous les points référencés, on pratique alors un rejet en distance, indiquant de la sorte que l'on est incapable de le classer.

Dans la figure 1, les classes et les coordonnées des points a , b , c et d sont connus. On fixe $k = 1$. On cherche à classer les points p_1 , p_2 et p_3 dont on ne connaît que les coordonnées. Le point le plus proche de p_1 est c , donc p_1 se verra attribuer la classe de c . Le plus proche voisin de p_2 est d , il se verra donc

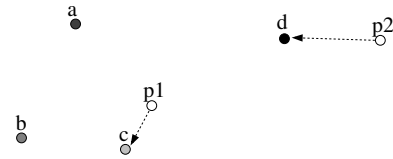


FIG. 1 – Classement des points p_1 , p_2 et p_3 selon l'algorithme des k plus proches voisins

attribuer la classe de ce point. Quant à p_3 , bien que son voisin le plus proche soit d , il est trop éloigné de tous les points et est donc rejeté.

Pour calculer la distance entre deux points p_1 et p_2 , il suffit d'utiliser un produit scalaire et d'en déduire la distance euclidienne entre p_1 et p_2 : $d(p_1, p_2) = \|p_1 - p_2\|^2 = \langle p_1 - p_2, p_1 - p_2 \rangle$

Cet algorithme peut être utilisé pour classer des données de types quelconques, des productions d'apprenant par exemple, pour peu que l'on soit capable de représenter ces données sous la forme d'un vecteur de caractéristiques numériques ou que l'on soit capable de proposer une opération de produit scalaire entre deux éléments.

3.2 Projection en 2 dimensions : l'algorithme du Stochastic Neighbor Embedding

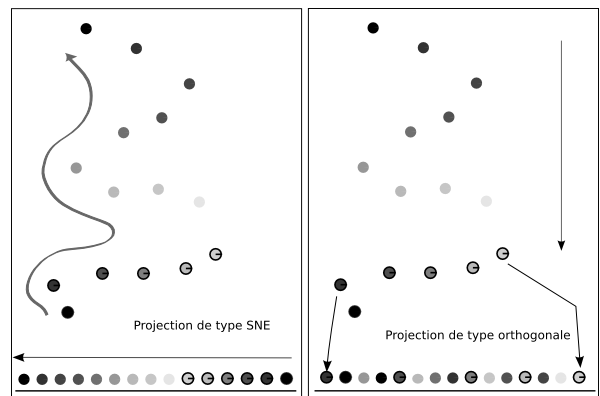


FIG. 2 – Méthodes de projection, avec ou sans prise en compte du voisinage.

La projection d'un espace de grande dimension à un espace de plus petite dimension pose souvent le problème dit du *repliement* : deux points distants l'un de l'autre peuvent se retrouver projetés au même endroit (voir dessin de droite figure 2). Pour palier cet inconvénient, il existe des méthodes qui cherchent à préserver le voisinage des points, telles que

deux points proches restent proches lors de la projection et deux points distants restent distants (voir dessin de gauche figure 2).

L'algorithme Stochastic Neighbor Embedding met cela en œuvre [?]. Nous avons besoin de deux matrices P et Q qui vont représenter respectivement une notion de voisinage dans l'espace de départ et dans l'espace d'arrivée. Sur la figure

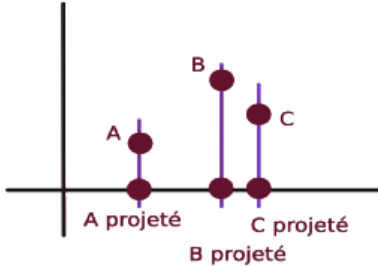


FIG. 3 – Exemple de projection qui conserve le voisinage. Cet exemple sert d'illustration à la définition des matrices de distances P et Q .

3, les points A , B et C sont disposés dans un espace à deux dimensions. Dans cet espace de départ, nous pouvons calculer la distance entre les points (P). De même, les points $A_{projete}$, $B_{projete}$ et $C_{projete}$ représentent la projection dans un espace à une dimension (une droite) et nous pouvons dans cet espace d'arrivée donner la distance entre les points (Q).

$$P = \begin{bmatrix} & A & B & C \\ A & 0 & 4 & 6 \\ B & 4 & 0 & 2 \\ C & 6 & 2 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} & A_{projete} & B_{projete} & C_{projete} \\ A_{projete} & 0 & 3 & 4 \\ B_{projete} & 3 & 0 & 1 \\ C_{projete} & 4 & 1 & 0 \end{bmatrix}$$

Dans notre cas, nous cherchons à définir la meilleure projection possible donc nous ne connaissons pas la matrice Q . Le but est par conséquent de trouver la transformation qui gardera le plus fidèlement possible les proportions de la matrice P avec pour contrainte de réduire arbitrairement le nombre de dimension pour représenter les points d'arrivée.

Plus formellement, la matrice P est calculée selon une distance d_{1ij} (euclidienne ou donnée par le problème) entre les points i et j dans l'espace de départ :

$$p_{ij} = \frac{\exp(-d_{1ij}^2)}{\sum_{i \neq k} \exp(-d_{1ik}^2)}$$

La matrice Q est calculée selon une distance d_{2ij} euclidienne entre les points i et j dans l'espace d'arrivée :

$$q_{ij} = \frac{\exp(-d_{2ij}^2)}{\sum_{i \neq k} \exp(-d_{2ik}^2)}$$

Les matrices P et Q représentent chacune dans leur espace la probabilité de chaque point d'être voisin de chaque autre. P est fixée par le problème à résoudre. L'objectif de l'algorithme est de trouver le meilleur agencement de points projetés dans l'espace d'arrivée pour que Q représente les mêmes probabilités de voisinage. Ainsi, si x_1 est le point de plus proche de x_2 dans l'espace de départ, y_1 (l'image de x_1) devra être le point le plus proche de y_2 dans l'espace d'arrivée.

On résout ce problème par une méthode de descente de gradient. Le principe est de chercher le minimum d'une fonction de coût C qui représente la divergence entre P et Q . Cet algorithme est itératif, chaque étape rapproche la solution courante de la solution finale (figure 4). L'itération s'arrête lorsque la solution courante n'est plus modifiée.

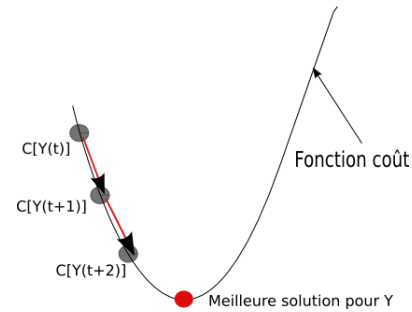


FIG. 4 – Illustration de la descente de gradient.

En pratique, l'utilisation de cette méthode ne requiert pas de forme particulière pour les données d'entrée. Il suffit de savoir donner une distance entre les points pour pouvoir l'utiliser. La taille de l'espace de sortie est arbitraire, il est par conséquent aisé de fixer à deux pour obtenir une visualisation des points.

3.3 Mise en œuvre de notre solution

L'outil que nous proposons repose sur l'utilisation conjointe de ces deux techniques. Dans un premier temps, l'algorithme Stochastic Neighbor Embedding permet de projeter dans un espace à deux dimensions les productions des apprenants, représentées sous forme de données auxquelles est associée une opération de calcul de distance. Puis l'algorithme des k Plus Proches Voisins est utilisé avec $k = 1$ pour classer ces projections par rapport à des cas connus.

Afin d'utiliser notre outil, il faut au préalable :

- représenter les productions des apprenants sous la forme d'un vecteur de caractéristiques numériques ou proposer une opération de calcul de produit scalaire entre deux productions,
- classer manuellement, préalablement à l'analyse automatique, un certain nombre d'éléments.

Les avantages de cette solution sont multiples. Tout d'abord, la projection en 2 dimensions préalable à la classification permet d'en améliorer les résultats. De tels

résultats ont été vérifiés expérimentalement [?]. Cette technique permet en outre à l'enseignant de visualiser le processus de classification. Une telle transparence fait habituellement défaut aux techniques issues de l'apprentissage machine, ce qui rend parfois difficile l'acceptation du résultat proposé par le système.

Les possibilités de visualisation permettent également à l'enseignant d'obtenir des informations qu'il n'aurait pas forcément pu détecter de lui-même.

Notre outil peut donc être utilisé dans les cas où l'on souhaite :

- positionner la production d'un apprenant parmi un ensemble de cas connus,
- observer la répartition des apprenants d'une formation au sein de ces divers cas.

Les cas connus au préalable peuvent être des cas de référence (bonnes réponses, conceptions erronées), des productions d'apprenants classées à la main ou un mélange des deux.

Techniquement, notre prototype a été conçu en utilisant le langage Python et la librairie Numeric dédiée au calcul scientifique. Pour l'instancier dans un EIAH particulier, il suffit de créer une classe (au sens de la programmation objet) représentant les productions des apprenants et disposant d'une méthode `produit_scalaire`.

4 Mise en Œuvre Concrète

Dans cette section, nous indiquons comment nous avons appliqué l'outil présenté ci-dessus dans le cadre des exemples évoqués à la section 2.

4.1 Dans le contexte de MÉTADYNE

Au sein de MÉTADYNE, les apprenants construisent leur définition des notions importantes du cours sous la forme d'un ensemble de propositions que l'on nomme *attributs* pour respecter la terminologie proposée par Barth dont nous nous sommes inspirés.

Ainsi, la définition de l'algorithme du tri récursif pourra être constituée des attributs suivants :

- est un *tri*,
- a pour sous-procédure *diviser en deux parties*,
- a pour sous-procédure *trier les deux parties*,
- a pour sous-procédure *combiner les deux parties*.

Il est possible de déterminer un produit scalaire entre deux ensemble d'attributs de la manière suivante : $\langle A_1, A_2 \rangle = |A_1 \cap A_2|$. Ainsi, deux définitions seront proches si elles ont beaucoup d'attributs en commun.

Les problèmes propres à l'analyse du langage naturel (synonymie, ambiguïtés syntaxiques, ...) ne se posent pas dans notre contexte : lors de la réalisation de leur définition, les apprenants doivent construire les attributs de leur définition

à partir d'un ensemble d'éléments fixé au préalable par l'enseignant.

Une fois cette opération de produit scalaire déterminée, il est possible d'instancier notre outil d'analyse et de tenter de classer les données issues d'une séance d'évaluation. Nous avons donc tenté de l'utiliser à partir des données issues d'une expérimentation que nous avons mise en œuvre. Cette expérimentation réunissait des étudiants-ingénieurs du département ASI (Architecture des Systèmes d'Information) de l'INSA de Rouen dans le cadre d'un cours sur l'informatique répartie. Après avoir identifié les définitions de référence puis classé à la main les cartes d'un apprenant afin d'augmenter la précision de la classification, nous avons projeté les cartes des autres apprenants.

4.2 Dans le contexte de Baghera

Le système Baghera a nécessité le développement d'un système multi-agents qui pourrait être réutilisé dans d'autres contextes, mais n'est pas aussi générique que la solution que nous proposons. En outre, il n'offre pas la possibilité de visualiser la manière dont les productions des apprenants ont été projetées, ce qui ne permet par exemple pas d'avoir une vision globale du niveau d'un groupe d'apprenants, une classe par exemple.

L'utilisation de notre outil a été possible directement avec les données proposées au sein du document [?]. Pour cela, nous avons considéré que chacun des observables permettant de qualifier la production d'un apprenant pouvait être le signe d'une seule, de deux ou des trois conceptions étudiées. Ainsi, chaque observable peut être représenté sous la forme d'un vecteur v en 3 dimensions, chacune indiquant la ou les conceptions candidates, en posant $\sum_{i=1}^3 v_i = 1$.

Chaque production sera alors également représentée sous la forme d'un vecteur en 3 dimensions qui sera la somme de tous les vecteurs représentant les observables qui la composent.

Les résultats obtenus sont d'aussi bonne qualité que ceux proposés par la méthode reposant sur des agents. La figure 5 montre le résultat de la projection. Chaque carré représente la production d'un apprenant. Les couleurs correspondent à la conception que les didacticiens avaient majoritairement attribuée à cette production. Idéalement, il faudrait donc que tous les points d'une même couleur soient projetés dans la même zone. On peut observer visuellement que cette situation idéale est globalement vérifiée, à l'exception notable d'un des points. Il est à noter que ce point pose également problème dans la solution reposant sur la technique multi-agents et il semble manifestement que la représentation des données soit en cause.

5 Conclusion et Perspectives

L'outil que nous avons développé a pu être utilisé dans des contextes relativement variés. Nous envisageons de

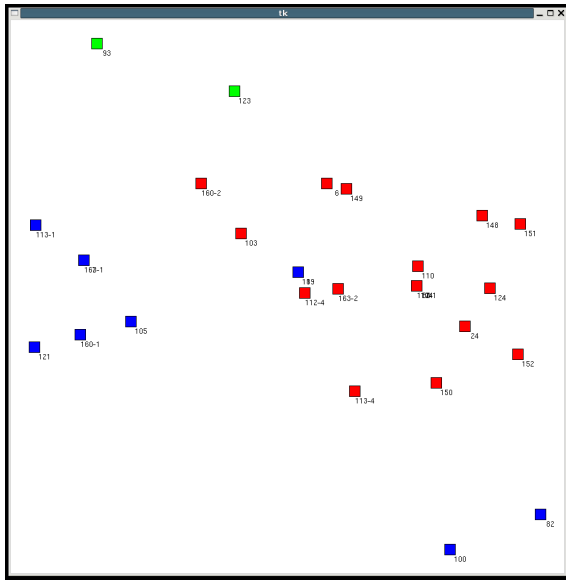


FIG. 5 – Projection des données issues de [?] à l'aide de notre outil

l'utiliser dans d'autres contextes encore, afin d'en éprouver la généralité et de pouvoir déterminer avec précision les cas dans lesquels il peut être appliqué. Notons qu'à l'heure actuelle, il couvre les niveaux les plus bas de la taxonomie des objectifs cognitifs proposée par Bloom [?]. Ainsi, nous allons prochainement tenter de le mettre en œuvre dans le cadre de la construction de cartes conceptuelles par l'apprenant sans donner de contraintes à ce dernier. Une telle application pose bien entendu le problème de l'analyse du langage naturel au sein des cartes. Ce problème n'a pas encore été résolu, mais nous pensons travailler sur les techniques proposées par T. Conlon et qui reposent sur la participation de l'apprenant au cours de l'analyse de sa carte [?].

Le logiciel que nous avons développé n'est en outre qu'un prototype et il est tout à fait envisageable d'y ajouter de nouvelles fonctionnalités. Le temps nécessaire à la projection d'une production d'apprenant étant de quelques secondes dans la plupart des cas, il est ainsi tout à fait envisageable de mettre en œuvre un suivi synchrone des apprenants en indiquant à l'enseignant, en temps réel, l'état de leurs productions.

Enfin, il reste tout à fait envisageable d'utiliser d'autres techniques de projection en 2 dimensions ou de classification supervisée si de telles techniques permettent d'obtenir de meilleurs résultats, que ce soit en temps de calcul ou en précision.