# Categorizing students' questions using an ensemble hybrid approach

Fatima Harrak
Sorbonne Université
CNRS, Laboratoire d'Informatique de Paris 6,
LIP6, F-75005 Paris, France
fatima.harrak@lip6.fr

François Bouchet
Sorbonne Université
CNRS, Laboratoire d'Informatique de Paris 6,
LIP6, F-75005 Paris, France
francois.bouchet@lip6.fr

Vanda Luengo
Sorbonne Université
CNRS, Laboratoire d'Informatique de Paris 6,
LIP6, F-75005 Paris, France
vanda.luengo@lip6.fr

## ABSTRACT

Students' questions categorization is a challenging task as the available corpora are often limited in size (particularly with languages other than English) and require a costly preliminary manual annotation to train the classifiers. Ensemble learning can help improve machine learning results by combining several models, and is particularly efficient to leverage the strengths of very different classifiers. In this paper, we investigate how combining a rule-based annotator (based on keywords identified by an expert) with various machine learning-based approaches and TF-IDF can improve the automated identification of questions asked by 1st year medicine students on an online platform, according to a coding scheme using 4 dimensions. First we evaluated the performance of several models, calculating the kappa between the prediction and the manually labelled dataset, according to each dimension. Then, using a stacking approach, we tried different combinations of them to design a predictive model with a higher performance. The results reveal that the new ensemble models can help to increase the performance for all dimensions of the dataset, in particular those for which the expert rule-based system showed the lowest performance. These results are promising as they indicate that some easy-to-train models can complement more manual approaches, even with a small training set of a few hundreds of annotated questions.

## Keywords

Student's question, ensemble learning, stacking, coding scheme, hybrid method, question categorization

## 1. INTRODUCTION

Categorizing students' questions with limited size of corpora remains a challenging task. The available classification methods require a costly preliminary manual annotation to obtain labeled data, and it is tempting to try training many different classifiers in the hope that combining their predictions would give good performance. One of the most active areas in machine learning has been in studying methods to build good ensembles of classifiers [3]. The premise that ensembles are often much more accurate than the individual classifiers make them more attractive. Ensemble learning helps improve machine learning by combining several models to obtain an overall classifier which prediction accuracy outperforms every single one of them [7, 9]. Among the existing ensemble approaches, stacking [17] is often used. It consists in training a combining classifier (sometimes called a meta-classifier), in addition to the set of individual classifiers, which takes as input the output of the other classifiers.

In this paper, we used a pre-existing coding scheme to annotate students' questions asked by 1st year medicine students on an online platform, and investigated different approaches to improve the automated identification of their questions. We used the stacking approach by combining heterogeneous classifiers such as a rule-based annotator with various machine learning-based approaches and TF-IDF. Our goal was to answer to two research questions: (RQ1) Can combining different approaches improve the performance of the predictive model? (RQ2) What is the best combination of families of classifiers, and in particular, can a hybrid system (mixing expert rules and machine learning) be relevant?

## 2. STATE OF THE ART

Annotating a corpus automatically requires using a coding scheme or a taxonomy of sentences, messages or speech acts. Many taxonomies have been used to characterize the types of questions that students ask. Graesser and Person [4] developed a taxonomy of questions asked during tutoring sessions according to the level of thought. Another well-known taxonomy widely used in education, the Bloom's taxonomy [2] and its revisions [1], was originally created in order to help teachers in formulating questions and therefore tends to be more appropriate for teachers' questions (*e.g.* assessment) than for students' ones. The questions coding scheme used here was defined based on the corpus at hand to match them more accurately, even if some overlap exists with categories of existing taxonomies.

Ensemble learning has been investigated in many studies [11, 10] and consists in weighting several individual classifiers, and combining them in order to obtain a classifier that outperforms every single one of them considered separately. First, different classifiers are generated by training models on different features from the training set. The generated

Fatima Harrak, François Bouchet and Vanda Luengo "Categorizing students' questions using an ensemble hybrid approach" In: *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019),* Collin F. Lynch, Agathe Merceron, Michel Desmarais, & Roger Nkambou (eds.) 2019, pp. 312 - 317

classifiers are then typically combined by some form of majority or weighted voting. In this work, we do not restrict how the individual classifiers are trained, but we deal with different models and not only probabilistic ones which is a prerequisite for some of these techniques.

The stacking framework introduced by [17], consists in combining multiple classifiers models created by using different learning algorithms on a single dataset. Several variations of this idea have been attempted. Ting and Witten stacked base-level classifiers whose predictions are probability distributions over the set of class values, rather than single class values. The meta-level attributes are thus the probabilities of each of the class values returned by each of the base-level classifiers. Multi-response linear regression, an adaptation of linear regression, is recommended for meta-level learning to predict binary variables [15]. Merz [8] proposed a stacking method called SCANN that uses correspondence analysis to detect correlations between the predictions of base-level classifiers. Todorovski and Dzeroski [16] introduced a new meta-level learning method for combining classifiers with stacking: meta decision trees have base-level classifiers in the leaves, instead of class-value predictions. Researchers in [14] presented a novel bayesian model that relies on combining different models in order to improve the classifier accuracy.

In this paper, we investigated how combining heterogeneous classifiers (derived by different learning algorithms, using different model representations) can help to improve the automated identification of questions using a stacking approach.

## 3. CONTEXT

The dataset considered in this paper is made of questions asked in 2012 by 1st year medicine/pharmacy students from a major public French university. The Faculty of Medicine and Pharmacy has a specific hybrid training system (reading of the material for the class is done at home, and classroom time is dedicated mostly to Q&A) for their 1st year students. The 1st year is divided into two semesters, each of them ending with a competitive exam (in January and May) on the content studied during the period: only a predefined number of students is allowed to continue in the second year. Between the reading session at home and the classroom session, the students can connect to an online platform to either ask a question, or see questions asked by other students and vote for them if they also want an answer to that question. They cannot however propose an answer to those questions. Then, the questions asked online are sent to teachers to help them prepare their Q&A session. The dataset contains 6457 questions overall asked by 429 students.

## 4. QUESTION CODING SCHEME

We chose to consider the nature of questions as defined in the coding scheme introduced in [5], in a process involving multiple human coders and several refinement phases. This coding scheme (summarized in Table 1) consists in tagging each question according to 4 independent dimensions: a main one (dimension 1), which is mandatory, and 3 optional ones (dimensions 2 to 4 - *cf.* Table 1 for the corresponding definition of each dimension). For instance, a question could be a request to re-explain the way something work by providing another example (tagged as Ree (1) on dimension 1, Exa (1) on dimension 2, Man (2) on dimension 3, and nothing (0) on dimension 4, *i.e.* represented as the dimension vector [1,1,2,0]). It could also be a request to verify (Ver = 3) if a schema (Sch = 2) needs to be known for the final exam (Exp = 3) (which would be tagged as [3, 2, 0, 3]). We considered here a corpus of 923 questions manually annotated according to the 4 dimensions of the coding scheme for training and testing the automated annotators. A subsample of 723 questions was used for training the classifiers, and 200 questions were used to test their performance.

**Table 1: Coding scheme**

| ID | Dim1 | Question type | Description |
|---|---|---|---|
| 1 | Ree | Re-explain / redefine | Ask for an explanation already done in the course material |
| 2 | Dee | Deepen a concept | Broaden a knowledge, clarify an ambiguity or request for a better understanding |
| 3 | Ver | Validation / verification | Verify or validate a formulated hypothesis |
| **ID** | **Dim2** | **Explanation modality / Quest. subject** | **Description** |
| 1 | Exa | Example | Example application (course/exercise) |
| 2 | Sch | Schema | Schema application or an explanation about it |
| 3 | Cor | Correction | Correction of an exercise in course/exam |
| **ID** | **Dim3** | **Explanation type** | **Description** |
| 1 | Def | Define | Define a concept or term |
| 2 | Man | Manner (how?) | The manner how to proceed |
| 3 | Rea | Reason (why?) | Ask for the reason |
| 4 | Rol | Roles (utility?) | What's the use/function |
| 5 | Lin | Link between concepts | Verify a link between two concepts, define it |
| **ID** | **Dim4** | **Verification type (optional)** | **Description** |
| 1 | Mis | Mistake / contradiction | Detect mistake/contradiction in course or explanation of teacher |
| 2 | Kno | Knowledge in course | Verify knowledge |
| 3 | Exp | Expected knowledge | Verify expected information in exam or quiz (assessment) |

## 5. AUTOMATED ANNOTATORS

### 5.1 Approach 1: expert rule-based annotator

We used first a previously developped custom annotator relying on keywords manually identified and associating them a weight [6]. To design it, the human annotator identified from a separate dataset of questions in the corpus the keywords that were indicative of each dimension value (*e.g.* in Dimension1, for the dimension value “Re-explain”, some of the keywords identified were “re-explain”, “restate”, “redefine”, “retry”, “repeat”, “revise”, ”get back on”, *etc.*). For each question, for each dimension, the question was tagged in the dimension according to the value that had the highest number of keywords associated to it (*e.g.* for dimension 1, a question with two keywords associated to the value ”re-

**Table 2: Kappa between the standalone expert rule-based annotator and the reference manual annotation**

| Dim1 | Dim2 | Dim3 | Dim4 |
|---|---|---|---|
| 0.76 | 0.69 | 0.70 | 0.65 |

explain" and one keyword associated to the value "validation" would be tagged as a "re-explain" question).

The automatic annotator is using a set of weights associated to each keyword of each dimension (*e.g.* "explain": 7, "what/how": 3), and defined using the set of 723 questions. Those weights were determined in two steps: first, the human annotators empirically associated a weight between 1 and 9 to each keyword, depending on whether they thought they were very marginally (1), significantly (5) or very significantly (9) associated to a given dimension. Then in a second step, the automatic annotator was used on the 723 manually annotated questions, and weights were manually adjusted (adding or removing 1) on some keywords for questions for which the manual and automatic annotation were different, iterating until full agreement was obtained on almost all segments from the 723 questions. Finally, the question identified by the values associated to each dimension, is represented as a dimension vector.

The Kappa values per dimension for the annotations coming from human expert and the automatic annotator are given in Table 2.

## 5.2 Statistical approaches

### 5.2.1 Data coding: from questions to words vectors

First, we used the French version of WordNet [12], a lexical database linking semantic concepts to each other in an ontology according to a variety of semantic relations (*e.g.* synonyms and hyperonyms). The aim was to transform different synonyms into the same expression in the questions (*e.g.* for dimension value "Reason" in Dim3, the synonym words "cause", "reason" and "motif" were replaced in the text by "why") to reduce the lexical diversity and consolidate a particular expression for the following treatment. Then, the classical preprocessing steps are used on the corpus of 923 questions: punctuation removal, stemming, tokenization, and stopwords filtering. After extracting the unigrams and bigrams for all questions in the corpus, the weights for the words are computed using two different methods: (1) TF-IDF (described in the next section), (2) counting occurencies ('1' if the word is in the question, '0' otherwise). Each of the 723 questions was represented by a word vector according to (1) or (2). We finally reduced the number of extracted keywords to keep the most important and significant ones using a feature selection technique (removing less frequent and correlated unigrams and bigrams).

### 5.2.2 Approach 2: TF-IDF

We used TF-IDF [13] to compute term weights. The goal of TF-IDF is to estimate how the words in a given document are representative of that document when compared to a larger set of documents. It combines two complementary metrics: the term frequency (TF), and the invert document frequency (IDF). TF thus gives a higher weight to the commonly occurring terms and a lower weight to rare terms. The drawback is that some words that are common in a given document but also common in all documents could end up with a weight that is over-representing their real importance. IDF fixes this issue by adjusting the weight with the general importance of the term. Equation 1 describes the method to compute individual TF-IDF weight values for each term (word). We made two different calculation measures of TF-IDF on the corpus of 723 questions.

$$W_{ik} = TF_{ik} \cdot \log\left(\frac{N}{n_k}\right) \tag{1}$$

Where:

$W_{ik}$ = TF-IDF weight for term $k$ in document $Q_i$
$TF_{ik}$ = frequency of term $k$ in document $Q_i$
$IDF_{ik}$ = inverse document frequency of term $k$ in document $Q_i = \log\left(\frac{N}{n_k}\right)$
$N$ = total number of documents in the questions corpus
$n_k$ = number of documents in the corpus that contain the term $k$

The first version consists in calculating four separate TF-IDF on each of the four dimensions, to extract the words that differentiate each category on each dimension. For a given dimension, all the questions manually annotated in each category (*e.g.* "Re-explain") were considered as documents (*e.g.* on dimension 1, document1 is the union of questions annotated as "Ree"). Each document (set of questions) is converted into a corresponding word-weight vector, where each word-weight represents the TF-IDF measure for the word in the document. TF-IDF weight ($W_{ik}$) was attributed for each term $k$ in document $i$ ($i$ is the number of documents in that dimension, *e.g.* $i$ varying from 1 to 3 for dimension 1). In order to classify new questions, we used the TF-IDF weights calculated on each dimension value from the sample of 723 questions. We attributed TF-IDF weights calculated on the training sample for the corresponding words on the test sample of 200 questions. Then, we chose the simplest ranking function which consists in summing the TF-IDF weights for each question on each dimension value. Therefore, for each question, for each dimension, we tag the question in that dimension according to the value that has the maximum weights. Finally, we calculated the Kappa values between the values found by this model [**TF-IDF+MAX**] for that dimension, and the corresponding values found by the manual annotation (*cf.* first column of Table 3). Two versions were tested: one where the questions were preprocessed using WordNet (*cf.* previous section) and one where they were not. The results obtained were similar in terms of performance, so we decided to keep the version including the preprocessing with WordNet, as it intuitively should generalize better to variations of existing questions.

In the second version, TF-IDF was calculated on the corpus of 723 questions without distinguishing the different dimensions. The questions were not grouped by dimension value, but instead, each question in the corpus was considered as a document (*i.e.* 723 documents overall). The document

**Table 3: Kappa between automatic annotation obtained by standalone TF-IDF + different ML methods and the reference manual annotation**

| | TFIDF + | | | | | | |
|---|---|---|---|---|---|---|---|
| **Dim.** | **Max** | **GLM** | **GBT** | **NB** | **KNN** | **DT** | **RI** |
| Dim1 | 0.66 | 0.69 | **0.71** | 0.47 | 0.62 | 0.46 | 0.61 |
| Dim2 | 0.39 | **0.73** | 0.69 | 0.12 | 0.56 | 0.49 | 0.36 |
| Dim3 | **0.66** | 0.59 | 0.60 | 0.43 | 0.58 | 0.37 | 0.52 |
| Dim4 | 0.58 | **0.71** | 0.63 | 0.37 | 0.60 | 0.19 | 0 |

is then converted into a corresponding word-weight vector, where each word-weight represents the TF-IDF measure for the word in the question. Finally, we used the word vectors as the input for machine learning techniques to predict the value associated to the question in that dimension (described in section 5.2.3).

### 5.2.3 Approach 3: ML-based annotator

We tried 6 machine learning (ML) classification techniques (Generalized Linear Model [**GLM**], Gradient Boosted Trees [**GBT**], Decision Tree [**DT**], K Nearest Neighbors [**K-NN**], Rule Induction [**RI**] and Naïve Bayes [**NB**]) for each dimension separately. The appropriate hyper-parameters (such as k for K-NN) were chosen in each case to obtain the highest value and may differ from one table to another. For each classifier, the input was the word vectors and the label to predict was the value associated to the question in that dimension. We considered the corpus of 923 questions as labeled data. Then, we trained the classifiers on the 723 questions and evaluated their performance on an independent sample of 200 questions, to ensure a good estimation of the performance on unseen data. Finally, we calculated the Kappa values between the values found by the classification model for that dimension, and the corresponding values found by the manual annotation. We considered two versions for comparison here as well: the first one using the corpus processed using WordNet, and the second one without the processing with WordNet.

## 5.3 Results

The kappa values found with the three automated annotators taken individually (expert rule-based, TF-IDF and ML) are provided in Tables 2, 3 and 4 respectively for each dimension. We note that the expert rule-based annotator clearly outperforms both ML-based annotator and TF-IDF only on dimension 1, whereas they almost have similar performances on dimension 3. TF-IDF with the classifier GLM gives the best performance on dimension 4. Furthermore, the ML-based annotation without WordNet performs better than the classifiers using WordNet for all dimensions and particularly on dimension 2.

## 6. ENSEMBLE HYBRID APPROACH

Our next step consists in building a predictive model with a higher performance to improve the automated identification of questions according to the coding scheme provided in Table 1. Using the aforementioned stacking approach, we tried different combinations of models regardless of which classifier is the best one. Moreover, it does not require any of the classifiers to be probabilistic; they can even be human experts. Our goal was not only to obtain the best classifier

**Table 4: Kappa between automatic annotation obtained by standalone different ML methods and the reference manual annotation**

| **Dim.** | **GLM** | **GBT** | **NB** | **K-NN** | **DT** | **RI** |
|---|---|---|---|---|---|---|
| Processing using WordNet | | | | | | |
| Dim1 | 0.69 | 0.70 | 0.28 | 0.60 | **0.73** | 0.69 |
| Dim2 | 0.10 | 0.74 | 0.10 | 0.50 | **0.79** | 0.37 |
| Dim3 | **0.68** | 0.64 | 0.37 | 0.61 | 0.59 | 0.60 |
| Dim4 | 0.63 | **0.66** | 0.34 | 0.60 | 0.48 | 0.63 |
| Processing without WordNet | | | | | | |
| Dim1 | 0.73 | 0.69 | 0.33 | 0.56 | **0.74** | 0.66 |
| Dim2 | 0.58 | 0.81 | 0.12 | 0.48 | **0.85** | 0.29 |
| Dim3 | **0.70** | 0.65 | 0.35 | 0.60 | 0.57 | 0.62 |
| Dim4 | 0.63 | **0.67** | 0.46 | 0.59 | 0.10 | 0.47 |

performance, but also to do so using a fairly small training set of annotated questions and see if a good performance could be obtained nonetheless.

## 6.1 Method for stacking

In the first phase, a set of 20 base-level models have been created (1 expert rule-based annotation, 7 TF-IDF annotation and 12 ML-based annotation). In this second phase, we want to train a meta-level classifier that combines the outputs of the base-level models. In other words, we have 20 predictions for each dimension for each of the 200 question segments in the testing set, as well as the 20 manual annotations for these 200 segments that provide a grounded truth, and we want to train a classification model using some subsets of these 20 features. We trained the meta-level classifier using the same aforementioned 6 classification techniques (GBT, GLM, NB, K-NN, DT, and RI) for each dimension separately, using a 10-fold cross validation to ensure a good estimation of the performance (*i.e.* training the models on 180 segments and testing on 20). Finally, for each model we calculated the Kappa values between the values found by that meta-model for that dimension, and the corresponding values found by the manual annotation. Regarding the set of features we considered, we wanted to consider combinations that mixed different set of approaches, and we therefore considered six meta-learning combinations described below. For each of them, the training was performed four times (once for each of the four dimensions - *cf.* Figure 1).

**(1) Stacked TF-IDF models:** We combined the outputs of the methods using each individiual TF-IDF classifier to compute keywords weights (*i.e.* 7 features for each classifier, *cf.* Table 3).

**(2) Stacking TF-IDF with expert rule-based annotation:** We combined the outputs of the TF-IDF models with the output of the expert rule-based annotator (*i.e.* 8 features for each classifier, *cf.* Tables 3 and 2).

**(3) Stacked ML techniques:** We combined the outputs of the machine learning-based annotation with the two combinations: processing using WordNet and without it (*i.e.* 12 features for each classifier, *cf.* Table 4).

**(4) Stacking ML techniques with expert rule-based annotation:** We combined the outputs of the machine learning-based annotation (with and without WordNet) with

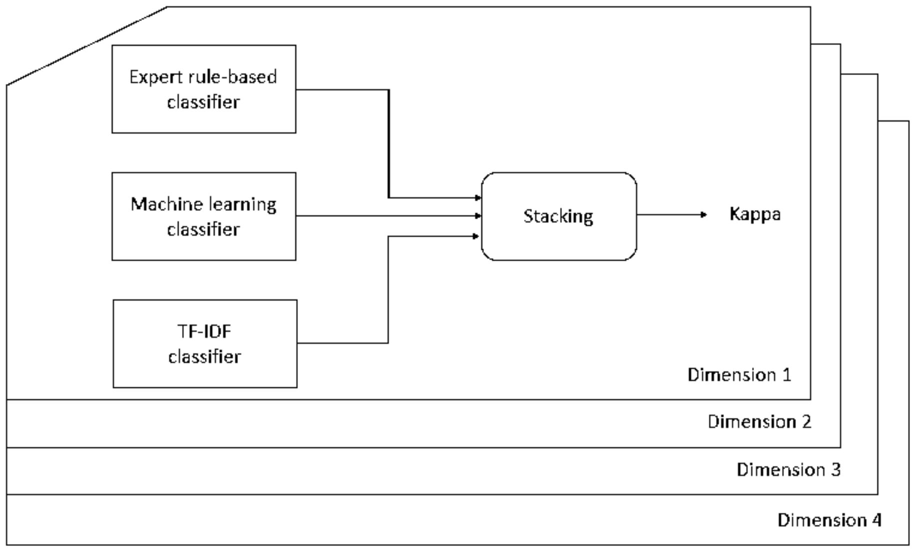

**Figure 1: The overall stacking process**

the output of the expert rule-based annotation (*i.e.* 13 features for each classifier, *cf.* Tables 4 and 2).

**(5) Stacking ML techniques with TF-IDF:** We combined the outputs of the machine learning-based annotation (with and without WordNet) with the output of TF-IDF based annotation (*i.e.* 19 features for each classifier, *cf.* Tables 4 and 3).

**(6) Stacking ML, TF-IDF and expert rule-based annotation:** We combined the outputs of all the existing classifiers: the machine learning-based annotation (with and without WordNet) with the output of TF-IDF and expert rule-based annotation (*i.e.* 20 features for each classifier, *cf.* Tables 4, 3 and 2).

## 6.2 Results and discussion

The kappa values found with the 6 classification techniques for each dimension are provided in Table 5. Each stacking model was trained individually on each dimension and the highest value obtained for each dimension among the 6 classifiers is tagged in bold, for each set of features considered. For instance, on the first row, we see that when combining the 7 TF-IDF classifiers that predict dimension 1, the best stacking result is obtained with a decision tree (0.75), which outperforms the best individual TF-IDF classifier (0.71 with GBT, *cf.* Table 3). We can notice that Naive Bayes is often the best ensemble classifier among the 6 tested, giving better performance on a small dataset. The best overall performance between the 6 set of comparisons are marked with a star (*): for dimension 1 and 4, it is Naive Bayes combining the ML and the expert rule-based classifiers, for dimension 2 it is Naive Bayes combining TF-IDF and the expert rule-based classifiers, and for dimension 3 it is GBT combining also TF-IDF and the expert rule-based classifiers.

When considering the combinations involving TF-IDF, we see that the combination of several TF-IDF outperforms the base-level TF-IDF on dimension 1 and 3. The kappa values are overall lower on dimensions 2 and 4, which is probably due to the unbalanced training data in these dimensions (it also explains why sometimes a classifier would obtain a kappa of 0 on these dimensions in the various tables). Moreover, the various TF-IDF classifiers combined with expert-rule based annotator outperforms both the TF-IDF base-level and expert-rule based annotator, as well as the combination of several TF-IDF. Similar results were found for several TF-IDF combined with machine learning, with a slightly better performance than individual classifiers. Overall, if one had to choose only one set of features, the best option is an hybrid ensemble (TF-IDF with expert rule-based annotator), which outperforms on average the model combinations with an average kappa of 0.77 (from the classifiers giving the best performance on each dimension, *i.e.* NB on dimensions 1 and 2, GBT on dimension 3 and K-NN on dimension 4).

When considering the combinations involving ML-based classifiers, the ML-based annotator combined with expert rule-based outperforms slightly the base-level machine learning on dimensions 1, 3 and 4 compared to the other ML combinations. Similarly to TF-IDF, the hybrid ensemble (ML with expert rule-based annotator) gives an average kappa of 0.77 instead of 0.74 for the base-level ML.

The combination of the three types of approaches obtains a performance similar or lower than the two other previously mentioned hybrid ensembles.

## 7. CONCLUSION

In this paper, we have shown that even with a small training set (less than 1000 questions), it can be useful to add ML-based approaches to complement a manually crafted annotator using a stacking approach to combine classifiers with each other. Using an hybrid ensemble of machine learning-based (or TF-IDF-based) annotators with a previously existing annotator seems to be the best approach, leveraging

**Table 5: Kappa values between the ensemble models and the reference manual annotation**

| **Stacked TF-IDF models** | | | | | | |
|---|---|---|---|---|---|---|
| **Dim.** | **GLM** | **GBT** | **NB** | **K-NN** | **DT** | **RI** |
| Dim1 | 0.73 | 0.74 | 0.72 | 0.73 | **0.75** | 0.70 |
| Dim2 | 0 | 0.35 | **0.67** | 0.49 | 0.51 | 0 |
| Dim3 | 0.62 | **0.70** | 0.66 | 0.67 | 0.68 | 0.66 |
| Dim4 | 0.55 | 0.67 | 0.68 | **0.69** | **0.69** | 0.67 |
| **Stacking TF-IDF + expert rule-based** | | | | | | |
| **Dim.** | **GLM** | **GBT** | **NB** | **K-NN** | **DT** | **RI** |
| Dim1 | 0.73 | 0.72 | **0.76** | 0.72 | 0.68 | 0.71 |
| Dim2 | 0 | 0.30 | **0.80*** | 0.66 | 0.48 | 0 |
| Dim3 | 0.70 | **0.79*** | 0.76 | 0.77 | 0.75 | 0.67 |
| Dim4 | 0.60 | 0.66 | 0.72 | **0.73** | 0.67 | 0.65 |
| **Stacked ML models** | | | | | | |
| **Dim.** | **GLM** | **GBT** | **NB** | **K-NN** | **DT** | **RI** |
| Dim1 | 0.76 | 0.73 | **0.80** | 0.76 | 0.71 | 0.68 |
| Dim2 | 0.30 | 0.48 | **0.77** | 0.59 | 0.62 | 0 |
| Dim3 | 0.62 | 0.71 | 0.71 | **0.72** | 0.70 | 0.65 |
| Dim4 | 0.58 | 0.65 | **0.72** | 0.67 | 0.68 | 0.57 |
| **Stacking ML + expert rule-based** | | | | | | |
| **Dim.** | **GLM** | **GBT** | **NB** | **K-NN** | **DT** | **RI** |
| Dim1 | 0.77 | 0.77 | **0.80*** | 0.76 | 0.70 | 0.69 |
| Dim2 | 0.16 | 0.48 | **0.77** | 0.60 | 0.62 | 0 |
| Dim3 | 0.64 | **0.76** | 0.71 | 0.73 | 0.66 | 0.64 |
| Dim4 | 0.60 | 0.66 | **0.74*** | 0.69 | 0.63 | 0.59 |
| **Stacking ML + TF-IDF** | | | | | | |
| **Dim.** | **GLM** | **GBT** | **NB** | **K-NN** | **DT** | **RI** |
| Dim1 | **0.77** | 0.73 | **0.77** | 0.76 | 0.71 | 0.68 |
| Dim2 | 0.30 | 0.52 | **0.78** | 0.61 | 0.62 | 0 |
| Dim3 | 0.66 | 0.75 | 0.71 | **0.72** | 0.70 | 0.62 |
| Dim4 | 0.60 | 0.64 | **0.71** | 0.71 | 0.64 | 0.61 |
| **Stacking ML + TF-IDF + expert rule-based** | | | | | | |
| **Dim.** | **GLM** | **GBT** | **NB** | **K-NN** | **DT** | **RI** |
| Dim1 | 0.77 | 0.75 | **0.78** | 0.76 | 0.72 | 0.68 |
| Dim2 | 0 | 0.56 | **0.78** | 0.58 | 0.62 | 0 |
| Dim3 | 0.65 | **0.77** | 0.72 | 0.73 | 0.67 | 0.61 |
| Dim4 | 0.61 | 0.63 | **0.70** | 0.69 | 0.63 | 0.61 |

the benefits of each approach. Combining TF-IDF and ML-approaches, however, does not seem as relevant. In our case, the hybrid ensemble models helped in increasing the performance for almost all dimensions, thus replying positively to our two initial research questions. It is worth noting though that the use of WordNet to reduce the vocabulary did not help in increasing the classifiers performance in our case.

One of the limits of this paper is that we considered only a single coding scheme and dataset. The increase in kappas can also be sometimes seen as modest, but this is to be put in perspective with the fact that human coders using this coding scheme rarely can reach a kappa superior to 0.75 on such a task. Moreover, one should note that the dimensions that were improved were the ones that were the furthest from the human coder performance. To conclude, we believe our result can open the perspective to easily improve the performance of various speech act and message annotators which often only rely on expert rules annotators.

## 8. REFERENCES

[1] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock. A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives, abridged edition. *White Plains, NY: Longman*, 2001.

[2] B. S. Bloom and M. B. Engelhart. Furst, EJ. *Hill, WH, & Krathwohl, DR Taxonomy of educational objectives. The classification of educational goals. Handbook I: Cognitive domain. New York: Longmans Green*, 1956.

[3] T. G. Dietterich. Machine-Learning Research. *AI Magazine*, 18(4):97–97, Dec. 1997.

[4] A. C. Graesser and N. K. Person. Question asking during tutoring. *American educational research journal*, 31(1):104–137, 1994.

[5] F. Harrak, F. Bouchet, and V. Luengo. Identifying relationships between students' questions type and their behavior. In *10th International Conference on Educational Data Mining*, pages 402–403, 2017.

[6] F. Harrak, F. Bouchet, V. Luengo, and P. Gillois. Profiling Students from Their Questions in a Blended Learning Environment. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, pages 102–110, New York, NY, USA, 2018. ACM.

[7] Y. Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009):1–10, 2009.

[8] C. J. Merz. Using Correspondence Analysis to Combine Classifiers. *Machine Learning*, 36(1):33–58, July 1999.

[9] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *European Conference on Computer Vision*, pages 57–70. Springer, 2010.

[10] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

[11] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, Feb. 2010.

[12] B. Sagot and D. Fišer. Building a free French wordnet from multilingual resources. In *OntoLex*, 2008.

[13] G. Salton. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*, 1989.

[14] E. D. Simpson, M. Venanzi, S. Reece, P. Kohli, J. Guiver, S. J. Roberts, and N. R. Jennings. Language Understanding in the Wild: Combining Crowdsourcing and Machine Learning. In *Proceedings of the 24th International Conference on World Wide Web*, pages 992–1002, Geneva, Switzerland, 2015.

[15] K. M. Ting and I. H. Witten. Issues in Stacked Generalization. *Journal of Artificial Intelligence Research*, 10:271–289, May 1999.

[16] L. Todorovski and S. Džeroski. Combining Multiple Models with Meta Decision Trees. In D. A. Zighed, J. Komorowski, and J. Żytkow, editors, *Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Computer Science, pages 54–64. Springer Berlin Heidelberg, 2000.

[17] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, Jan. 1992.