2020 Conference on Information Communications Technology and Society (ICTAS) 978-1-7281-3770-4/20/$31.00 ©2020 IEEE 10.1109/ICTAS47918.2020.244042

Evidence quality estimation using selected machine learning approaches

Aleksandra Byczyńska, Maria Ganzha
Dept. of Mathematics and Information Science
Warsaw University of Technology
Warsaw, Poland
al.byczynska@gmail.com, M.Ganzha@mini.pw.edu.pl

Marcin Paprzycki
Systems Research Institute
Polish Academy of Sciences
Warsaw, Poland
paprzyck@ibspan.waw.pl

Mikołaj Kutka
Faculty of Medicine
Medical University of Warsaw
Warsaw, Poland
mikolaj.kutka@gmail.com

***Abstract*—Evidence Based Medicine, is a practice, where medical actions/decisions are undertaken on the basis of best available evidence-based recommendations. In this context, we propose a system for automatic grading of evidence.**

Evidence grading is approached as a multi-label classification task. Here, classes represent grades, in a widely used Strength of Recommendation Taxonomy (SORT). Numerous ensemble methods are experimented with. It was found that the most successful one used Support Vector Classifiers, trained on multiple high level features, results of which are used to train a Random Forest Classifier. The best achieved accuracy score was 75.41%, which is a significant improvement over the baseline of 48%, achieved by classifying all instances as the majority class. It was also found that the most important predictor is the *publication type* of articles comprising the body of evidence. The designed system is tuned for use with medical publications and SORT. However, due to it's generality, it can easily be used with other evidence grading systems.

***Index Terms*—evidence based medicine, machine learning, natural language processing, classification, ensemble techniques, stacking classifiers, consensus, game theory, neural networks, multi–label classification**

I. Introduction

In recent years, Evidence Based Medicine (EBM) has become one of leading approaches to support making medical decisions. According to the best practices of EBM, any action concerning patient care should be made according to the best available evidence-based recommendations [1], [2]. Obviously, correct recommendation can only come from well-designed and conducted research. The process of discerning what is the best recommendation consists of the following steps. (1) Medical practitioner gathers all available research on the topic, usually using online medical publications' databases, such as MEDLINE [3] and sophisticated search engines, such as PubMed [4]. (2) Recommendations collected from collected papers are grouped into "bodies of evidence", i.e. sets of articles stating the same recommendation. (3) Quality (strength) of each recommendation is assessed. (4) The highest quality recommendation, from the body of evidence, is followed. Obviously, taking into account the volume of publications, manual assignment of evidence quality is no longer feasible.

Therefore, crucial for current, and future, EBM are computer-based methods for evidence quality grading. Only such methods can, reasonably, provide a systematic way of assessing quality of evidence, present in the deluge of publications. There exists a vast selection of techniques designed specifically for evidence grading. Our current work focuses on Strength of Recommendation Taxonomy (SORT) [5], which was first introduced in 2004, and is widely used to this day. SORT uses a three grade scale to describe the evidence, by addressing its three most important aspects: (i) quality, (ii) quantity and (iii) consistency. It is a patient-oriented (versus disease-oriented) taxonomy, focusing on influence, of a given treatment, on patient mortality and morbidity rates. Grades recognized in SORT are: *A* – evidence of a high quality (consistent, patient–oriented); *B* – evidence of moderate quality (inconsistent, patient–oriented); and *C* – reserved for bodies of evidence of low quality (consensus, usual practice, opinion, disease-oriented, etc).

While assessing quality of a single piece of evidence (e.g. a publication) is of importance, here, we focus our attention on estimating the final grade of the body of evidence, only.

II. Research in the domain of evidence based medicine

Let us start from a brief overview of the most pertinent literature. Of course, there exists much larger body of research focused on facilitating access to information, for the evidence-based medical practice, which was omitted, due to lack of space. Evidence quality estimation, in the field of EBM, can be described as a classification task [6]–[9] and, as such, is often approached by other researchers in the field.

In the work reported by Sarker, Molla-Aliod and Paris [6]–[8], different types of information were extracted from the abstract, title, and metadata of publications, and converted into features. Authors experimented with different classifiers and sets of features, and found that the best accuracy scores, for a simple classification, can be achieved with the *publication type* as the only feature (up to 68% accuracy). However, medical papers rarely encompass explicit information about their publication type. Hence, in an endevour to fully automate the pipeline, a sophisticated rule-based system was developed to extract the publication type from the paper. Using it, accuracy score for a classifier trained on *publication type* feature dropped to 58.5%. This is reasonable, since the quality

978-1-7281-3770-4/20/$31.00 ©2020 IEEE

of publication type estimator itself has influenced the total performance of the system.

In a simple classification scenario, including more features did not significantly improve the classification score. Subsequently, a sequence of classifiers was introduced, each specialised on a single feature set, to maximise the final accuracy. This technique yielded a higher score, of 62.84%.

Current work of Gyawali, Solorio and Benajiba [9] proposes a different approach. Instead of a pipeline of classifiers, they used an ensemble technique – a stacking classifier, which used classifiers trained on high-level features, as its estimators. Here, six disjoint features were explored: *publication type*, *MeSH* (Medical Subject Headings[1]), abstract *title*, *body*, *methods* and *conclusions* sections. None of base classifiers, tested in their work, yielded an accuracy score higher than 55%. However, experiments with different feature sets, for the ensemble classifier, showed that an accuracy score of 73.77% can be achieved. Specifically, this can be achieved by using five Support Vector Classifiers, each trained on a single feature (excluding the *abstract body* feature), as estimators for the stacking-classifier.

III. Theory, techniques, tools and algorithms for EBM recommendations

Let us now start description of our work, by summarizing experiment's steps and continue to list the selected high-level features, as well as explain the extraction methods and ensemble techniques used in the classification process.

A. Methodology

The goal of our work was to develop a system that will be able to accurately assess the quality of a body of evidence. The starting point was an attempt to reproduce results of the state-of-the-art approach reported in [9]. All experiments were conducted on the same dataset as used by both: Sarker, Molla-Aliod and Paris [6]–[8], as well as Gyawali, Solorio and Benajiba [9]. In order to reproduce findings reported there, the following steps were undertaken:

1) Extraction of high-level features from the publications.
2) Experiments with diverse preprocessing and transforming methods.
3) Training of various classifiers on the extracted features with single-feature trained classifiers.
4) Intermediate feature sets generation.
5) Training stacking classifiers (on intermediate features).

Subsequently, the following techniques were experimented with:

1) Extracting other promising features from publications.
2) Generating intermediate feature-sets, with multi-feature trained classifiers.
3) Experimenting with other ensemble techniques.

Let us now summarize our findings.

B. Selected high-level features

After careful analysis of the available dataset (explained in Section IV-A), and examination of features, which have been used by other researches in the field of evidence quality prediction, eight high-level features have been selected: (1) **abstract text**; (2) **conclusions section**: sentences belonging the *conclusions* section extracted from an abstract; (3) **journal title**; (4) **MeSH** (see, reference, above); PubMed recognizes two types of MeSH: those that are associated with the main topic of the article (with label *majortopic* set to *Y*), and other MeSH, which aren't related to the main topic of a publication (with label *majortopic* set to *N*); (5) **methods section**: sentences belonging the *methods* section of an abstract; (6) **publication type**: the type of medical publication; (7) **publication year** and (8) **title**.

C. Extraction algorithms for the selected features

While some features were used "as is", i.e. abstract text, or title, others had to be extracted. Hence, let us now briefly outline the pertinent extraction processes.

- **Conclusions** and **methods** sections: if an abstract is structured, then sentences belonging to the *conclusions* and *methods* sections are used as features. If the abstract is not structured, then an algorithm, found in [9], is used to extract sentences that should belong to the relative sections, based on their place in the abstract.
- **MeSH**: experiments were run to measure the advantage of using both types of MeSH, and only those with label *majortopic* set to *Y* were selected; since they yield better results. Note that the approach found in [9] also uses only these MeSH.
- **Publication type**: Sarker, in [8], mentions examples of regular expressions, used to determine *publication type* of a medical paper. Those expressions were used also in our work, alongside other expression developed to extract the *publication type* with better precision. A re-engineered algorithm, used in [8], was used for extraction.

  After preliminary experiments, it became apparent that development of a system that would accurately extract *publication type* from the abstract and metadata is a very complex task. This information is rarely stated explicitly in the source texts and often various *publication types* are mentioned. Therefore, since this issue is not the focus of our work, a different strategy was chosen.

  Most of evidence items has a human-annotated sentence explaining why a specific grade in Strength of Recommendation Taxonomy (SORT) [5] was assigned (these sentences are present in a sibling dataset [10], available from [11] and they are not used in the state-of-the-art work). These human-annotated explanations usually mention publication types of the articles. An example sentence may be of a form: "*based on multiple high quality randomized controlled trails*". Sarker, et.al. in [6]–[8], use these sentences in their experiments and achieved an accuracy score of up to 68%, using only this feature.

[1] https://www.nlm.nih.gov/mesh/meshhome.html

However, since such information is not present in "real-life", we do not use it in our later experiments. Nevertheless, when these, human-annotated sentences were used, when present, alongside the abstract, title and metadata of publications during the publication type assignment, the accuracy scores of up to 70% was observed.

D. Feature's preprocessing and transforming

All extracted features can be divided into three main categories: text, categorical and numerical. The text features are: (1) text of *abstract*, (2) *conclusions*, and (5) *methods* sections, (3) *journal title,* and (8) publication *title*. These features have to undergo a natural language preprocessing, before they can feasibly be used as features for the classifiers. Thus, the following steps were undertaken: (a) stop words and number removal, (b) stemming/lemmatization and (c) Meta Mapping. Meta Map [12] is a semantic understanding tool, used to decrease the vocabulary, by generalizing some domain-related concepts. Multiple experiments were performed, in order to determine whether the vocabulary should be processed in this way, and to what extent. For obvious reasons, this step is omitted for the journal title. For the remaining features, three levels of Meta Mapping were chosen:

1) no Meta Mapping
2) replacement of disease or syndrome related terms by a general tag **diseaseorsyndrome**. Note that Gyawali, Solorio and Benajiba ([9]) also choose to perform such Meta Mapping. Example of a sentence with the replacement can be seen in Table I.

TABLE I
DISEASE OR SYNDROME TERMS META MAPPED

Original	Meta Mapped
In a prospective, double–blind, randomized trial the efficacy of a heparinoid in ointment form was assessed in treating **superficial thrombophlebitis** developing after continuous intravenous infusion.	In a prospective, double–blind, randomized trial the efficacy of a heparinoid in ointment form was assessed treating **diseaseorsyndrome** developing after continuous intravenous infusion.

3) all terms with medical connotation are replaced by their general group tags. This transformation is similar to the previous one, but this time it is applied to **all** pertinent terms. Example of a sentence with this replacement can be found in Table II.

TABLE II
MOST TERMS META MAPPED

Original	Meta Mapped
In a prospective, double–blind, randomized trial the efficacy of a heparinoid **in ointment** form was assessed in treating **superficial thrombophlebitis** developing after continuous **intravenous infusion**.	In a prospective, double–blind, randomized trial the efficacy of a heparinoid **biomedicalordentalmaterial** form was assessed in treating **diseaseorsyndrome** after continuous **therapeuticorpreventiveprocedure**.

Next, *ngrams* were created, using text preprocessed by previous operations. Finally, *count vectorization* and *term frequency–inverse document frequency* (tf-idf) was applied.

The categorical features are: (3) *MeSH*, and (6) *publication type*. Since these features are already vectors of keywords, preprocessing, such as stop words removal, is not necessary. The only transforming steps are *count vectorization* and *tf-idf.*

Publication year is the only numerical feature. The mean value of all publication years, of publications belonging to the body of evidence, was used to proportionately scale the data.

E. First–level classification

A first step, in an ensemble classification, is training simple classificators that can be then used as estimators in the ensemble setting. High-level features obtained within pipelines, described in the previous section, were used to train the simple classifiers. Each training instance is a body of evidence: a set of publications (described by reference ids), and a SORT grade. A classifier was trained on a union of features of all publications belonging to a body of evidence.

A variety of classifiers has been trained. For each classifier, a grid search was performed to find the best set of parameters, and the results were evaluated by a 10-fold cross validation on the train and test sets. Chosen classifiers were:

- Support Vector Classifier (SVC) [13]
- Gradient Boosting Classifier [14]
- Extra Trees Classifier [15]
- Random Forest Classifier (RFC) [16]
- Decision Tree Classifier (DTC) [17]
- Gaussian Process Classifier [18]
- MLP Classifier [19]
- K–nearest Neighbors Classifier [20]
- AdaBoost Classifier [21]
- Gaussian Naive Bayes [22]

For all classifiers, the Scikit-learn [23] library was used. The selected scoring function was accuracy.

F. Ensemble methods

Once simple classificators were tested and the most accurate ones were selected, their classifications' probability vectors were used to generate intermediate feature sets. These were then used as the input for the ensemble methods.

1) Data preparation for the ensemble methods: Classifiers yielding the highest accuracy scores, in first-level classification, were used to generate intermediate features, which were then used as an input to ensemble classifiers. The intermediate features are three-value vectors. The *ith* value in a vector states the probability that this particular instance belongs to *ith* class.

Typically, the dataset would be divided into three data sets: train, development (dev), and test. Here, first-level classifiers would be trained on the train set, and probabilities would be generated by the trained first-level classifiers for the dev and test sets. Subsequently, ensemble classifiers would be trained on the dev set, and evaluated on the test set.

However, because the total available dataset is relatively small, a k-fold cross validation-like method was used to generate intermediate features. Here, the entire train set was divided into *k* folds, each comprising of 10 elements. Each *i*th fold was treated as a test set, while the classifier was trained on the remaining elements. Then, it was asked to return probabilities of class belonging for the *i*th fold. Using this procedure, probabilities for all instances of the train set have been generated. These were serialised for further usage. Subsequently, all train data was used to train a classifier, which generated predictions for the test set. These were also serialized. This process ensures that there is no "information leak" between the train and the test set, while maximizing the utility of the small dataset.

2) Selected machine learning techniques: After input data for the ensemble techniques was prepared, a varied selection of ensemble classification methods were experimented with. There belonged to four main groups:

- Classifiers – a variety of different classifiers from the Scikit-learn library were chosen.
- Neural networks – several models were designed and tested and the architecture that performed best in the preliminary experiments was selected. Chosen model, depicted on Fig. 1, has an input layer with I_n nodes where n depends on the number of features used. Both hidden layers have 512 nodes and the output layer has 3 nodes, each representing the probability score associated with a SORT grade. Each of the layers is densely connected. First two layers have activation function *Rectified Linear Units (RELU)*, whereas the output layer has a *Softmax* activation function. As the objective function, *categorical crossentropy* was chosen. *Keras* [24] library was used to design the neural network. All experiments were run with six different optimizers: *Adadelta* [25], *Adagrad* [26], *Nadam* [27], *Adam* [28], [29], *Adamax* [28] and *Rmsprop* [30].

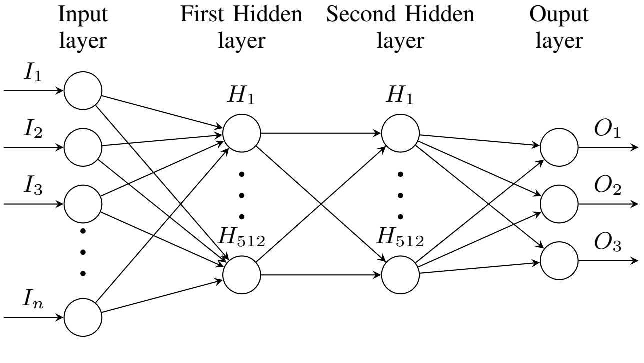

Fig. 1. Representation of neural network used in experiments

- Game theory – an approach that has been proven to be successful in similar problems [31], [32], where multiple experts (agents) can be distinguished. Each agent represents probability vectors (confidence scores) of an element belonging to a class. This approach can be used when experts have different "opinions" on the label that should be assigned. In order to classify an instance, a game is played by the agents. A game consists of rounds, which are played between two agents at a time. There are as many rounds as it is necessary for all agents to agree, or for only one agent to remain in the game.
- Consensus methods – similar to the game theory method, also an agent-based technique [31], [32].

IV. Experiments

Let us now summarize results of performed experiments. As it could have been noted, we have run much larger number of experiments, but we are reporting only the most interesting/important ones.

A. Dataset

The dataset used in this work was created for the ALTA 2011 Shared Task competition [33]. It was used in both [6] and [9]. Although this dataset is not freely available online, its authors were kind enough to provide it. The dataset consists of three files: *train*, *dev* and *test*, containing bodies of evidence (references) and assigned SORT grades. Each line comprises of an evidence id, SORT grade and a list of references (PubMed ids). The dataset also has publications (as XML files). Table III represents bodies of evidence distribution across SORT grades. In this work *train* and *dev* sets were merged into one set, which is further referred to as the *train* set. This was done because the dataset is relatively small and, hence, prone to overfitting. Creating an even smaller *train* set would be counter productive.

TABLE III
Distribution of bodies of evidence in respect to SORT grades

	Numeric				**Percentage (%)**			
	Train	***Dev***	***Test***	***All***	***Train***	***Dev***	***Test***	***All***
A	212	48	56	316	31.3	27.0	30.6	30.4
B	311	80	89	480	45.9	45.0	48.6	46.2
C	154	50	38	242	22.8	28.1	20.8	23.3
All	677	178	183	1038				

B. Evaluation process

For every experiment, numerous metrics were calculated. The key metric was accuracy, however balanced accuracy and AED (which corresponds to the closeness of the predicted to the actual label), were measured as well. They were used to make more informed decisions and understand more fully, which techniques (do not) work and why.

When conducting experiments, the following steps were performed, in order to make sure the best feature set-classifier pair was found:

1) Feature subset generation – either subsets that were the most probable to yield the highest scores were generated, or, when possible, all subsets were generated.
2) Classifier selection – numerous different classifiers (classification techniques) were selected and tried.
3) For every feature set and classifier pair, grid search for the best parameters was performed, with 10-fold cross

validation on the train set. For settings where there were too many possibilities, grid search was performed only for the best performing pairs.

4) When evaluating on the test set, mean accuracy was calculated with 95% confidence intervals (CI) over 10 runs, to show that the reported score is reproducible.

Let us now describe in detail results of our experiments.

C. Experiments with first-level classification

First, experiments with different feature preprocessing steps were run. For each step, the best performing method was selected for each feature individually. *Stemming*, generally, gave better results than *lemmatization*, and was used for all text features, except for the *abstract*-related ones (*abstract*, *conclusions* and *methods* sections). No decisive difference in performance using different Meta Mapping levels was detected. However, other researchers emphasised importance of this step, because it reduces the feature sparseness. Hence, we have generated three parallel features for *abstract*, *methods*, *conclusions* and *title*, with three levels of Meta Mapping and used them in the experiments. Number and stop words removal were used for all text features. Ngrams of length 1-4 were generated for *abstract* and unigrams and bigrams for *methods*, *conclusions*, *journal title* and *title*.

Second, experiments with different classifiers and regressors were run, in order to maximize the accuracy score. For each feature (Table IV), different classifiers were experimented with and each classifier's parameters were tuned.

TABLE IV
FEATURES AND THEIR IDS

No.	Feature	No.	Feature	No.	Feature
1a	Abstract (M0)	3	Journal title	7	Publication year
1b	Abstract (M1)	4	MeSH		
1c	Abstract (M2)	5a	Methods (M0)	8a	Title (M0)
2a	Conclusions (M0)	5b	Methods (M1)	8b	Title (M1)
2b	Conclusions (M1)	5c	Methods (M2)	8c	Ttile (M2)
2c	Conclusions (M2)	6	Publication type		

M0 – No Meta Mapping; **M1** – Disease or syndrome terms Meta Mapped; **M2** – Most medicine related terms Meta Mapped

The best performing feature was *publication type*, with an accuracy score of 70.038%. It was not surprising, since previous works also showed that this feature was the most important, in the classification process. Other features yielded scores in a range 48-55%. *MeSH* and *publication year* yielded the lowest accuracy scores, and, what became evident from the confusion matrices, in these cases, all elements from the test set were assigned a majority class label *B*.

The next phase of experiments involved classifiers trained on multiple features. First, all subsets of disjoint features were generated and used to train various classifiers. "Disjoint" refers to the *abstract*-related and *title* features, as these features can exist in three levels of the Meta Mapped form. For each feature 1, 2, 5 and 8, only one of variants *a-c* could be selected. Moreover, since *conclusions* and *methods sections* are disjoint subsets of sentences of an abstract, these two could be chosen at the same time, but neither of them could be selected alongside the *abstract*. This approach minimizes the number of subsets, (hence reducing experiments time), and prevents giving artificially more importance to *abstract*-related (or *title*) features.

Experiments with different parameters were conducted and whilst most were classifier-specific, approaches where parameter $class\_weight = balanced$ was used, on average, yielded higher accuracy scores. SVC performed the best of all tested classifiers. The highest performing feature sets (accuracy score of up to 69%) included *publication type*, *journal title* and *MeSH* features, and did not include the *publication year* feature. The accuracy almost always rose after removing the *publication year* or adding *journal title* or *MeSH* to the feature sets.

It was a surprising finding, as *journal title* was believed to be an unimportant feature, and did not yield better than random results on its own. Also, feature sets with *conclusions* and *methods sections* (either both or one of them) achieved higher scores than those with the *abstract*. Approaches with *methods* yielded higher accuracy scores than those with *conclusions* in their feature set.

D. Experiments with ensemble techniques

In previous works, researchers focused on using classifiers trained on single features only (in the ensemble setting). However, since multi-feature trained classifiers performed significantly better than the single-feature trained ones, we have decided to experiment with a novel approach: ensemble technique involving multi-feature trained classifiers. This was tried in other domains with satisfying results ([31], [34]).

Moving in this direction, above mentioned best-performing classifiers became source of intermediate features. The accuracy of created datasets was measured by checking the probabilities returned by first-level classifiers against true values of instances.

Since it would not have been feasible to use all possible subsets of high-level features, only feature sets that yielded an accuracy score higher than 60% were selected. Table V shows the resulting combinations. However, for *abstract*, *conclusions*, *methods* and *title*, different levels of Meta Mapping were also used, so the number of intermediate features was 18 and not 10 as the Table suggests. This information was stripped for clarity and conciseness.

First, the experiments for the consensus ensemble technique were run. In these, scaling agent's confidence score by their feature set's accuracy score gave only slightly better results than not scaling. That means the feature sets with higher accuracy had more impact on the final score. The best scores achieved subsets of size 3 and 2. The highest accuracy score of 73.22% was for a 3-item subset.

Subsequently, experiments with a game theory ensemble technique were run. The parameters experimented with were:

- scaling agent's confidence scores vs. no scaling,
- adjusting the dropout score – when the highest confidence score of an agent drops below a certain value, the agent

TABLE V
ENSEMBLE TECHNIQUES – FEATURE SETS USED

No.	Feature
1	abstract, journal title, MeSH, publication type, title
2	conclusions, journal title, MeSH, methods, publication type, title
3	conclusions, journal title, MeSH, publication type, title
4	journal title, MeSH, methods, publication type, title
5	MeSH, publication type
6	MeSH, publication type, publication year
7	MeSH, publication type, publication year, title
8	publication type
9	publication type, title
10	publication type, MeSH, publication type, publication year, title

leaves the round (experimented with values between 0-30%),

- the number of times an agent can choose the *change* action (default: 2; experimented with values 1–10).

The best setup for this experiment was to scale agents' confidence scores and assign a high value for the dropout criterion. It is evident that this method needs a lot more features to generate accurate predictions than the consensus technique, as the highest performing subsets contained 4-11 feature sets. The highest accuracy score was 73.77%.

A neural network was also trained in the ensemble setting. Although the highest accuracy scores produced by this method rose up to 74% on the test set, when cross-validated (10-fold cross-validation on the train set), mean accuracy scores turned out not to have exceeded 61.5% with a very high standard deviation (around 8-10%). Here, the best subsets used an *adadelta* optimizer.

The final experiment involved a stacking classifier. Random Forest Classifier yielded the best results. Almost all accuracy scores of more than 70% were obtained. Other tree-based classifiers also yielded comparatively high scores. This approach gave the highest accuracy scores, with the best one being 75.41%, which is higher than 73.77% reported in [9].

V. ANALYSIS OF RESULTS

Let us now summarize the key observations made on the basis of obtained results.

A. First-level classification

Experiments with single- and multi-feature trained classifiers were run. Different preprocessing steps, classifiers and balancing techniques were tried. Dataset balancing did not improve the accuracy scores, however it did improve the *balanced accuracy* score. When intermediate feature sets were generated using this method, and different ensemble techniques were tried, returned results were inferior by around 10% for every technique, compared to the approach, which did not use balancing. This is a statistically significant difference. However, in most experiments, using a $class\_weight = balanced$ parameter, did improve the classification. This parameter artificially changes the weights of a classifier by the ratio of class representatives in the train set.

In all experiments, *abstract*, *conclusions*, *methods* and *title* features were tested in three different levels of Meta Mapping. After analysing the results, the only conclusion is that usage of the Meta Map tool did not, in any statistically significant way, improve the accuracy of classification.

For the first-level classification, different classifiers, and neural networks were used. Classifiers yielded superior results compared to the neural networks. SVC produced better results than other classifiers, for most feature sets. An SVC was also used by other researchers, who achieved similar results on this problem.

Neural network produced accuracy scores that were only slightly inferior. Moreover, standard deviation of their scores was significantly higher (around 10%, compared to around 2-5% for classifiers). An attempt to generate intermediate feature sets with neural networks was undertaken. However, the resulting accuracy scores were significantly inferior to the ones generated by classifiers. Hence, this endeavor was abandoned.

B. Ensemble techniques

When comparing all results of ensemble techniques, across the two intermediate feature sets, it was evident that techniques employed on the feature set generated from multi-feature classification performed better than those that used feature sets generated from a single-feature classification.

The most significant difference can be seen between results of the consensus and game theory techniques, where the highest accuracy scores varied between 66% to 73%, and 68% to 73% respectively. The higher results achieved in the latter case may be easily explained. These algorithms, contrary to classifiers or neural networks, cannot infer "hidden qualities" of the data – they are simply only "as good as their inputs".

The difference between the performance of ensemble classifiers and ensemble neural networks, in the two settings, was more evident than expected. One could only assume, that running the experiments on the intermediate features, which are more accurate, would yield significantly higher scores, but the difference between the approaches varied by only around 3-4% between the settings, when comparing the highest achieved scores.

Classifiers rendered highest accuracy scores on the test set, in all run experiments. While SVC was the highest performing classifier in the first phase of the experiments: when dealing with high-level features, Random Forest Classifier performed significantly better when trained on the intermediate feature sets (ensemble approach).

The correlation between high scores on the train and test sets, across the different methods was also examined. The technique with the highest correlation ratio was the consensus method. Next was the game theory technique, which also had quite high correlation between the scores. Unfortunately, classifier-feature sets pairs that scored particularly high on the train set, performed significantly poorer on the test set (74% and 66% for the best performing pair), whereas the pairs that achieved highest scores on the test set, yielded low scores on the train set (65% and 75%).

C. Comparison of results to other works

First set of conducted experiments were meant to reproduce findings from [9]. The experiment setup was reproduced by performing the same extraction and preprocessing of features. Tables VI and VIII compare both result-sets. Here, let us note that Gyawali, Solorio and Benajiba [9] did not report exact scores for their base classifiers, instead present results in a figure – those were approximated. For the sake of comparison, scores reported by Sarker, Molla-Aliod and Paris ([6]–[8]) are also depicted in Table VI.

TABLE VI
COMPARISON WITH OTHER RESEARCHERS – FIRST-LEVEL CLASSIFICATION (SINGLE FEATURE-TRAINED CLASSIFIERS)

No.	Our Results		Sarker, Molla-Aliod and Paris		Gyawali, Solorio and Benajiba
	Accuracy (%)	95% CI	Accuracy (%)	95% CI	Accuracy (%)
1a	**51.293**	50.9–51.7	-	-	-
1b	50.893	50.6–51.2	49.7	42–57	40.0
1c	50.528	50.3–50.8	-	-	-
2a	**50.893**	50.3–51.5	-	-	-
2b	50.492	49.4–51.6	-	-	47.5
2c	50.237	50.0–50.5	-	-	-
3	**49.727**	49.7–49.7	47.3	42–53	-
4	**48.889**	48.5–49.3	-	-	48.5
5a	51.913	51.5–52.3	-	-	-
5b	50.783	50.5–51.1	-	-	**52.5**
5c	51.148	50.3–52.0	-	-	-
6	**70.038**	70.0–70.0	58.5	51–66	50.0
7	**48.525**	48.3–48.8	47.6	42–53	-
8a	**55.628**	55.3–55.9	-	-	-
8b	50.128	49.9–50.4	52.5	45–60	54.0
8c	53.588	53.3–53.9	-	-	-

The accuracy scores per feature, even though not very high, are in line with results reported by other researchers, with the only exception for the *publication type*. In the extraction process for this feature, data (human-annotated explanation text) from outside of the competition dataset was used. Sarker, Molla-Aliod and Paris also used this data in some of their experiments; Table VII compares these results. It is divided into two sections: the first section contains results where they used human-annotated data; results in the second part are from experiments using a fully automatic approach and only data from the competition dataset. Gyawali, Solorio and Benajiba [9] did not use the human-annotated information at all, and reported an even lower accuracy score of 50% for the *publication type* feature.

We have made an attempt to directly compare the stacking approach to the results reported in [9]. For that, another set of experiments was performed. Experimental setting was almost the same as before, but this time human-annotated metadata was not used for the *publication type* extraction from the article. Instead, a fully automated approach was used. The accuracy score of a classifier trained on a *publication type* feature was only 56%, which was still higher than results reported in [9](50%). This feature was later used to generate an intermediate feature set and used in the stacking approach. Surprisingly, even though first-level classifiers depicted in this work yield higher accuracy scores the accuracy of stacking classifiers did not exceed 64% (compared to 73.77% reported in [9]). The cause for the discrepancy has not been diagnosed and will be further investigated in the near future.

TABLE VII
COMPARISON WITH SARKER, MOLLA-ALIOD AND PARIS – FIRST–LEVEL CLASSIFICATION (MULTIPLE FEATURE-TRAINED CLASSIFIERS)

Feature	Our Results		Sarker, Molla-Aliod and Paris	
sets	Classifier	Accuracy (%)	Classifier	Accuracy (%)
6	DTC	71	KNNC	69
3, 6, 7, 8	SVC	60	DTC	64
6, 7	SVC	58	DTC	67
6, 8	SVC	63	DTC	67
3, 6,	RFC	60	DTC	64
3, 7, 8	SVC	48	SVC	51
3, 7	SVC	46	SVC	46
6	DTC	71	DTC	59
1, 6, 8	SVC	54	SVC	60

First part of the table compares results achieved in this work to the ones where human-annotated data was used; Second part of the table compares results of this work to the fully automatic approach
DTC – Decision Tree Classifier; **SVC** – Support Vector Classifier; **KNNC** – K-nearest Neighbors Classifier; **RFC** – Random Forest Classifier;

TABLE VIII
COMPARISON WITH GYAWALI, SOLORIA AND BENAJIBA – FIRST-LEVEL CLASSIFICATION (MULTIPLE FEATURE-TRAINED CLASSIFIERS)

Feature	Our Results		Gyawali, Soloria and Benajiba	
sets	Classifier	Accuracy (%)	Classifier	Accuracy (%)
1, 4, 6, 8	SVC	65	SVC	46
2, 4, 5, 6, 8	SVC	68	SVC	50
4, 6	RFC	61	SVC	53

SVC – Support Vector Classifier; **RFC** – Random Forest Classifier

Compared to the work of Sarker, Molla-Aliod and Paris, our results are in line and slightly superior. Similar accuracy scores on the base features are reported, and the difference between the highest scoring feature and the end score is similar – around 5%.

We have to admit that their approach has a significant advantage, because they develop an automated system to extract *publication types* from abstracts and metadata, which makes their system fully automatic. This work shows that if a system that would extract *publication type* with a high accuracy (around 70% at least, compared to their 58%) was developed, it would be possible to use system described in this work in an automatic way as well. That would potentially yield even better results.

VI. DISCUSSION AND LESSONS LEARNED

The overarching goal of this work was to help medical practitioners make better informed decisions in the limited time they have for each patient, by helping them focus only on high quality research. In this context we have investigated methods that would predict evidence quality from raw publication abstracts and metadata by classifying the evidence into three grades: *A*, *B* or *C*.

As a result of our work, some discoveries about the features, previously mentioned by other researchers, were confirmed, namely

- the importance of accurate *publication type* assignment,
- lack of importance of *publication year* as a feature,
- observation that relevant parts (*methods* and *conclusions* sections) extracted from the abstract are better predictors than whole *abstract* texts.

An observation contradictory to the existing research was also made: *journal title*, previously believed to be insignificant, was often among the feature sets yielding high(er) accuracy scores.

A novel approach was also proposed – to use different, high accuracy yielding feature set-classifier pairs, as estimators for different ensemble techniques. The ensemble approach performed significantly better than the baseline (75% compared to 48%). Out of all tested ensemble approaches, classifiers yielded the highest accuracy score (75%), whilst consensus method gave the most consistent results, with only a slightly lower accuracy (72%). Here, upon reflection, one may note that better consistency may, in fact, be more useful in some real-life scenarios.

Due to the generality of the approach, it can easily be used for evidence quality estimation in other grading systems than SORT, though that would require a new dataset with annotated examples, which, to our best knowledge, does not exist. Future research would also benefit from enlarging the existing dataset, as the currently available dataset is quite small (while access to digital libraries of medical texts is very expensive). Another possible point of interest for researchers in the domain may be focusing on improvement of semantic tools such as MetaMap, which could be useful in finding relevant information in the publications, such as study group size or publication type.

References

[1] D.L. Sackett, W.M.C. Rosenberg, J.A. Gray, bhaynes@mcmaster.ca Haynes, and W.S. Richardson. Evidence based medicine: What it is and what it isn't. *BMJ (Clinical research ed.)*, 312:71–2, 02 1996.

[2] D.L. Sackett, W.S. Richardson, W. Rosenberg, and bhaynes@mcmaster.ca Haynes. Evidence-based medicine. how to practice and teach ebm. evidence-based medicine. *Churchill Livingston*, 2, 01 2005.

[3] https://www.nlm.nih.gov/bsd/medline.html. [Online; accessed 10-November-2019].

[4] https://www.ncbi.nlm.nih.gov/pubmed/. [Online; accessed 10-November-2019].

[5] Mark Ebell, Jay Siwek, Barry Weiss, Steven Woolf, Jeffrey Susman, Bernard Ewigman, and Marjorie Bowman. Strength of recommendation taxonomy (sort): A patient-centered approach to grading evidence in the medical literature. *The Journal of the American Board of Family Practice / American Board of Family Practice*, 17:59–67, 02 2004.

[6] Abeed Sarker, Diego Molla Aliod, and Cécile Paris. Automatic evidence quality prediction to support evidence-based decision making. *Artificial Intelligence in Medicine*, 64, 04 2015.

[7] Diego Molla Aliod and Abeed Sarker. Automatic grading of evidence: the 2011 alta shared task. 12 2014.

[8] Abeed Sarker. *Automated Medical Text Summarisation to Support Evidence-based Medicine*. PhD thesis, 01 2014.

[9] Binod Gyawali, Thamar Solorio, and Yassine Benajiba. Grading the quality of medical evidence. pages 176–184, 06 2012.

[10] Diego Molla Aliod and María Santiago-Martínez. Development of a corpus for evidence based medicine summarisation. *The Australasian medical journal*, 5:503–6, 10 2012.

[11] https://sourceforge.net/projects/ebmsumcorpus/. [Online; accessed 10-November-2019].

[12] Alan Aronson. Effective mapping of biomedical text to the umls metathesaurus: The metamap program. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, 2001:17–21, 02 2001.

[13] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[14] Jerome Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29, 11 2000.

[15] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63:3–42, 04 2006.

[16] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[17] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[18] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[19] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.

[20] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. [Online; accessed 10-October-2019].

[21] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

[22] Tony Chan, Gene Golub, and Randall LeVeque. Updating formulae and a pairwise algorithm for computing sample variances. 01 1979.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] François Chollet et al. Keras. https://keras.io, 2015.

[25] Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701, 2012.

[26] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, July 2011.

[27] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1139–III–1147. JMLR.org, 2013.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[29] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. 05 2018.

[30] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[31] M.C. Abreu and Anne Canuto. Analyzing the benefits of using a fuzzy-neuro model in the accuracy of the neurage system: an agent-based system for classification tasks. pages 2959–2966, 01 2006.

[32] Maria Ganzha, Marcin Paprzycki, and Jakub Stadnik. Combining information from multiple search engines-preliminary comparison. *Inf. Sci.*, 180(10):1908–1923, May 2010.

[33] https://www.alta.asn.au/events/sharedtask2011/. [Online; accessed 10-November-2019].

[34] Yunan Zhang, Qingjia Huang, Xinjian Ma, Zeming Yang, and Jianguo Jiang. Using multi-features and ensemble learning method for imbalanced malware classification. pages 965–973, 08 2016.