Heuristics for Evaluating Artificial Intelligence: Appreciating Context in Modelling

Ayse Kok Arslan

Abstract: Often without realizing it, we might develop in our everyday life an unconscious heuristic approach to working with AI and with those tools that make use of AI. The aim of this article is to outline a framework for evaluating artificial intelligence- (AI-) based tools, without the need to understand computing languages such as Python or to have detailed technical knowledge of how they were developed. The study offers a framework for evaluating AI tools based on Long and Magerko's idea of AI literacy andargues that the human users may in some cases be better placed to evaluate the capabilities of a tool than the original developers due to a lack of appreciation of the context in which it would be used. Finally, the study recommends some heuristics for developing non-biased AI tools.

Introduction

Despite the progress in the field, AI has been viewed through widely different perspectives during its long lifetime, dating back over 75 years to the 1950s – from wild optimism to being written off. Unfortunately, both attitudes are wide of the mark, and neither extreme is helpful for a balanced appraisal of AI tools.

There is a common misconception that when algorithms are used, they are the cause of any defects of the tool. Yet, it should be taken into account that the success or failure of AI is as much based on the corpus as it is on the algorithm. If the corpus used has an imbalance of any racial or ethnic origins, then the algorithm will simply replicate that bias.

Narrow AI tools can, if implemented sensibly, greatly enhance our ability to carry out many of the tasks. How these tools are selected and implemented is all important. This study aims to provide guidance on real-world selection, implementation and, finally, appraisal and metrics.

Main Definitions

AI tools make use of what is called 'supervised' or 'semi-supervised' machine learning. Supervised means there is some human involvement in setting up the tool, usually in determining what the correct answers should be.

'Machine learning' (ML) means the use of a computer to follow a pattern, whether or not the pattern is identified by a human.

'Natural language processing' or NLP means the identification of patterns in spoken or written text.

To give a specific example for an AI tool, the search query "How to make a veggie burger?" on YouTube yields thousands of videos, each showing a slightly different technique for the same task. As this can often be time-consuming for a first-time burger maker to go through this plethora of video content, imagine instead, if they could watch a compact visual summary of each video so that a summary of all semantically embedded instructions could provide a quick overview of what the longer video has to offer.

As this example shows, a key facet of human intelligence is the ability to effortlessly connect the visual and auditory world to natural language concepts. Bridging the gap between human perception (visual, auditory and tactile) and communication (via language) is hence becoming an increasingly important goal for AI, enabling tasks such as text-to-visual retrieval [9, 61, 81], image and video captioning [43,79,87], and visual question answering [7,46].

Based on the example above, the narrow, or limited, AI described here is based around a few components:

- a corpus
- a training set
- a test set
- an algorithm.

The 'corpus' is the body of content to be analyzed. The corpus contains some information or characteristic to be extracted based on the choice of the user such as text or collections of images.

The corpus-based approach using a training set uses the process of inductive reasoning. The goal is to use existing evidence to predict a likely inference to achieve good quality results so that results are better than a human could achieve without the tool. ‘Better’ in this sense refers to good quality delivered faster, or better quality with no loss of time compared to a manual process, or both.

The ‘training’ set is a subset of the corpus, which has been tagged in some way to identify the characteristic the user is looking for.

The ‘test set’ refers to the collection of documents to be used for testing the algorithm, to see how successfully it carries out the operation.

The ‘algorithm’ is simply the tool that looks at each item in the corpus and enables a decision to be made. An algorithm may be as simple (and frequently is as simple) as matching a pattern so that the machine is asked to find the closest match between the training set and the test set. Much of computing is based around identifying the most effective algorithm to solve a specific problem, for example, how to sort a collection of numbers into numerical order.

Review of Existing Studies

The term AI is frequently used as a buzzword to give the impression that a tool is more sophisticated than it really is. In practice, the kind of small-scale AI described above is very closely linked to ‘string matching’ or other well-established simple techniques. String matching means the use of a machine to identify instances of a sequence of characters in a text.

According to Russell and Norvig, one should define AI as the creation of intelligent agents. An agent is here defined as a system that can “perceive” its environment and “act” in the pursuit of certain goals. Intelligence, in turn, is defined in terms of what philosophers call instrumental rationality: the capacity to efficiently achieve one’s goals.

On a Turing-inspired definition of AI, artificial intelligence is achieved when one creates machines that can successfully imitate thinking human beings. Russell and Norvig define different types of artificial agents. For example, simple reactive agents can only act in response to specific stimuli that bring forth certain predetermined reactions.

In a recent paper, “The Turing Trap,” Brynjolfsson contends that rather than focusing on the idea that algorithms or robots will become substitutes for human-beings, envisioning ways of collaboration between human-beings and AI might be more beneficial for everyone in the longer term.

Augmenting a machine learning model with explanations has been studied in existing literature extensively. For example, in the supervised learning setting, a model can be fine-tuned using human-annotated rationales (Zaidan et al., 2007; Ling et al., 2017b; Narang et al., 2020; Camburu et al., 2018; Cobbe et al., 2021; Chung et al., 2022). Zelikman et al. (2022) proposes better rationale generation by augmenting ground truth answers as hints when predicted answers are incorrect.

In order to overcome such biases, it is important to understand models as a complex reflection of the many various patterns of association between ideas, attitudes, and contexts present among human-beings. Researchers refer to the degree to which a model can accurately reflect these distributions as its degree of algorithmic fidelity. The core assumption of algorithmic fidelity is that the model exhibits underlying patterns between concepts, ideas, and attitudes that mirror those recorded from human-beings with matching backgrounds.

In order for the algorithmic fidelity to establish the generalizability of language models, the following four criteria must be fulfilled:

- Criterion 1 (Social Science Turing Test): Generated responses are indistinguishable from parallel human texts.
- Criterion 2 (Backward Continuity): Generated responses are consistent with the attitudes and socio-demographic information of its input/“conditioning context,” such that humans viewing the responses can infer key elements of that input.
- Criterion 3 (Forward Continuity): Generated responses proceed naturally from the conditioning context provided, reliably reflecting the form, tone, and content of the context.
- Criterion 4 (Pattern Correspondence): Generated responses reflect underlying patterns of relationships between ideas, demographics, and behavior that would be observed in comparable human-produced data.

A lack of fidelity in any one of these four areas decreases confidence in its usability; a lack of fidelity in more than one decreases confidence further.

Algorithmic fidelity gained increasing popularity when it comes to designing neural networks. Research into artificial neural networks has drawn on various disciplines of science and engineering. At its core, deep learning involves three steps:

1) A large dataset of training examples is collected.
2) An expressive neural network is constructed. The neural network is parameterized by the weights of the linear building blocks, and adjusting these weights adjusts the function that the network implements.
3) The error—otherwise known as the loss—of the network over the training examples is evaluated so that the weights are then adjusted according to this gradient so as to reduce the error.

Minimizing error on train examples is often sufficient to attain good performance on previously unseen test examples. Despite this simplicity, some of the most basic questions surrounding optimization and generalization are not resolved.

- Optimization: Given the gradient of a neural network's error, how far and in which direction should the network weights be best adjusted?
- Generalization: Which of the functions that a neural network implements will perform best on unseen data.

A large number of optimization algorithms have been proposed for neural networks (Schmidt et al., 2021), and each has a set of adjustable parameters known as hyperparameters that affect the performance of the method. The learning rate controls how strongly the network weights are adjusted in response to the gradient of the network's error. In the absence of compelling theoretical guidance on how to set the learning rate, best engineering practice is to try a logarithmic grid of possibilities and to see what works best (Goodfellow et al., 2016).

In practice, these questions are usually addressed by trial-and-error over a set of heuristic techniques. Answering this question is important for various reasons. In many applications one is interested in returning the single network that makes the best possible predictions. Besides, in some applications one is interested in obtaining some measure of the uncertainty of the predictions of this best network.

There are many examples of neural networks in use without any mention that an AI tool is being used, although, increasingly, the impact of the AI tool might be too subtle to notice. For example, in a Google blog post about BERT, an ML technique for NLP, the benefit shown was simply the ability to link a preposition with a noun. Whereas earlier search tools tended to ignore prepositions and just focus on nouns, this more sophisticated tool was able to identify a meaningful connection between the 'to' and the 'USA'.

More recently, new capabilities have emerged from Large Language Models (LLMs)- a subcategory of NLP- as they are scaled to hundreds of billions of parameters (Wei et al., 2022a)to perform well on a task having never been trained on with only a handful of examples. Despite the strong reasoning ability of LLMs with or without few-shot examples and self-consistency (Wang et al., 2022b) improving the model performances still requires finetuning on an extensive amount of high-quality supervised datasets.

Transformers used in language models became extremely popular given their capabilities to generate new text. Usually, LMs take text as input, which is then tokenized into discrete tokens by a tokenizer. Each token is then mapped to a learned embedding, which is used as input to a transformer (Vaswani et al., 2017).

Wei et al. (2022b) propose a language model to generate a series of natural-language-based intermediate steps, and show it can help language models better solve complex and multi-step reasoning tasks. Zhou et al. (2022a) further decompose the questions into multiple sub-questions, and ask the language model to solve each sub-question sequentially. Jung et al. (2022) show that inducing a tree of explanations and inferring the satisfiability of each explanation can further help judge the correctness of explanations.

The study of language models also includes the phenomena of emergent abilities which are defined as abilities that are present in larger models. Emergent abilities may arise in prompted tasks in which a pre-trained language model is given a prompt for a task framed as next word prediction, and it performs the task by completing the response. A prompted task emerges when it unpredictably surges from random performance to above-random at a specific scale threshold.

The second class of emergent abilities encompasses prompting strategies that augment the capabilities of language models.

One example of an emergent prompting strategy is called "chain-of-thought prompting", for which the model is prompted to generate a series of intermediate steps before giving the final answer. Chain-of-thought prompting enables language models to perform tasks requiring complex reasoning, such as a multi-step math word problem.

In pursuit of a better user experience, delivering personalized content for each individual user as real-time response is a common goal of these models. To this end, information from a user's latest interaction is often used as the primary input for training a model, as it would best depict a user's portrait and make predictions of user's interest and future behaviors. However, efforts to leverage the power of deep learning are constantly encountered with problems arising from the unique characteristics of data derived from real-world user behavior mainly in two aspects:

(1) The features are mostly sparse, categorical and dynamically changing;
(2) The underlying distribution of training data is non-stationary.

To overcome such challenges, there is a need for a set of skills for evaluating new AI tools and understanding what parameters to measure and possible pitfalls to avoid when introducing a new utility. In other words, AI literacy is required to make use of AI tools.

Long and Magerkodefine AI literacy as 'a set of competencies that enable individuals to critically evaluate AI technologies, communicate and collaborate effectively with AI, and use AI as a tool'. They further define over 30 relevant factors, of which the first five are essential skills for the assessment and recommendation of AI tools:

1. The ability to distinguish between tools that use or do not use AI
2. Analyze differences between human and machine intelligence
3. Identify various technologies that use AI
4. Distinguish between general and narrow AI
5. Identify problem types that AI excels at and problems that are more challenging for AI

To be specific, the skills outlined here may not necessarily require the ability to code. Once developed, AI tools can be extended to domains where their validity is greatly reduced. There are several examples of this overextension of AI tools, with predictably irrelevant or meaningless results.

For many AI tools, a methodology based on the A/B test is feasible, if complex, to provide a solid assessment. This process is similar to building or adjusting websites in which A/B tests are often used. An A/B test is a randomized trial widely used in website development, in which two versions of a variable, such as a web page layout, are shown to different groups of users, and their resulting behavior is measured. A/B tests provide the following benefits:

- evaluations are made with data rather than by guesswork
- the test gives the response of real users
- the results enable the estimation of metrics of success – what is an acceptable goal (rather than an absolute goal).

As for human evaluation, all human-beings are not equal at this task. It is a well-established principle in website design that the best way to evaluate software is to test it with real users, not with the software team who built the tool, or with the people tasked with managing the delivery of the tool.

Without a solid analytical framework, human-beings tend to rely on instinct, which could be described as an internal assessment mechanism – they instinctively trust (or do not trust) a familiar methodology, or tools they have used before. The role for the information professional of a framework in all this is providing credibility: providing users with external validations that enable them to trust a tool and to deploy it with confidence.

The AI toolkit: A framework for evaluation of AI tools

Within the light of this information the following toolkit can be used to evaluate AI tools. Although making use of Long and Magerko's idea of AI literacy, the requirements here are much more specific.

Goal

- What is a realistic goal?

Expecting perfection for an AI utility is impossible as AI tools based on a training set cannot have 100% accuracy. Nevertheless, the accuracy they provide should be considerably greater than using human-beings for the same task.

Corpus

- Is the corpus large enough? Is the training set large enough?
- What are the start and end dates for the data in the corpus? Does this matter?
- Who chose the corpus, when was it chosen and for what purpose? Details of the corpus used, like the data for a research article, should be publicly stated and accessible.
- What is the corpus bias?
- Is the tool likely to raise diversity, equality and/or inclusion issues?
- Is personal data captured and reused?

Algorithm

- Have the developers provided a single-sentence summary of the methodology behind the algorithm?

Evaluation and Metrics

- Who to evaluate: end users or subject-matter experts, or both? Internal or external?
- What metrics will be used to evaluate the tool? The F1 score, if used, must be interpreted in context.

Sanity check

- Sanity check/common sense: Have the developers built in ‘common-sense’ limitations to prevent the algorithm being applied too widely? Am I asking a meaningful question? Is this a feasible exercise?
- Does the tool provide feedback when a question is out of scope?
- Based on the checks above, is the tool fit for purpose?

Dissemination

- Is there easy-to-read documentation and guidance for new users that explains in simple terms how to use the tool and how it improves on current processes?

Feedback

- Does the tool provide a feedback loop so it can be improved over time?

Recommendations

The following recommendations might be useful when it comes to developing reliable and non-biased AI toolkits. It is the author’s opinion that these can also be considered as heuristics of developing frameworks for non-biased AI.

1. Take the time to understand your data

Doing an exploratory data analysis and looking for missing or inconsistent records would be useful to do before training a model in order to reflect if any bias exist in the data.

2. Don’t look at all your data

It is important that one does not make untestable assumptions that will later feed into the model. It is fine to make assumptions, but these should only feed into the training of the model, not the testing. So, one should avoid looking closely at any test data in the initial exploratory analysis stage. Otherwise one might, consciously or unconsciously, make assumptions that limit the generality of a model in an untestable way.

3. Do make sure you have enough data

If there is no enough data, then it may not be possible to train a model that generalizes.

If the signal is strong, then one can get away with less data; if it’s weak, then one needs more data.

Data augmentation is also useful in situations where there is limited data in certain parts of your data set, e.g. in classification problems where there are less samples in some classes than others — a situation known as class imbalance.

If there is limited data, then it’s likely that one might need to limit the complexity of the ML models in use, since models with many parameters, like deep neural networks, can easily overfit small data sets.

4. Think about how a model will be deployed

If the model is going to be deployed in a resource-limited environment, such as a sensor or a robot, this may place limitations on the complexity of the model. Another consideration is how the model is going to be tied into the broader software system within which it is deployed.

5. Don't allow test data to leak into the training process

There are a number of ways that information can leak from a test set. For instance, during data preparation, using information about the means and ranges of variables within the whole data set to carry out variable scaling — in order to prevent information leakage, this kind of thing should only be done with the training data.

The best thing to prevent these issues is to partition off a subset of data at the start of your project, and only use this independent test set once to measure the generality of a single model at the end of the project.

6. Try out a range of different models

Generally speaking, there's no such thing as a single best ML model similar to the 'No Free Lunch' theorem, which shows that no ML approach is any better than any other when considered over every possible problem [Wolpert, 2002]. So, the aim is to find the ML model that works well for your particular problem

7. Don't use inappropriate models

A simple example of this is applying models that expect categorical features to a dataset containing numerical features, or vice versa.

Other examples of inappropriate model choice include using a classification model where a regression model would make more sense (or vice versa).

8. Don't assume deep learning is best

Non-linear functions are hard to follow at the best of times, but when you start joining them together, their behavior gets very complicated very fast. Whilst explainable AI methods can shine some light on the workings of deep neural networks, they can also mislead you by ironing out the true complexities of the decision space.

A deep neural network is unlikely to be a good choice if there is limited data, if domain knowledge suggests that the underlying pattern is quite simple, or if the model needs to be interpretable.

9. Do optimize model hyperparameters

Many of the hyperparameters significantly effect the performance of the model, and there is generally no one-size-fits-all.

It's much better to use some kind of hyperparameter optimization strategy, such as random search and grid search,

It is also possible to use AutoML techniques to optimise both the choice of model and its hyperparameters, in addition to other parts of the data mining pipeline — see He et al. [2021] for a review.

However, when carrying out both hyperparameter optimization and feature selection, it is important to treat them as part of model training, and not something more general that is done before model training.

10. Do evaluate a model multiple times

Many ML models are unstable. This means that a single evaluation of a model can be unreliable, and may either underestimate or overestimate the model's true potential. For this reason, it is common to carry out multiple evaluations. Crossvalidation (CV) is particularly popular, and comes in numerous varieties [Arlot et al., 2010].

11. Don't assume a bigger number means a better model

There are various reasons why a higher figure does not imply a better model. For instance, if the models were trained or evaluated on different partitions of the same data set, then small differences in performance may be due to this. If they used different data sets entirely, then this may account for even large differences in performance.

12. Do consider combinations of models

Different ML models explore different trade-offs; by combining them, you can sometimes compensate for the weaknesses of one model by using the strengths of another model, and vice versa.

13. Don't generalize beyond the data

One common issue is bias, or sampling error: that the data is not sufficiently representative of the real world. Another is overlap: multiple data sets may not be independent, and may have similar biases. So, in short, don't overplay your findings, and be aware of their limitations.

14. Be careful when reporting statistical significance

A positive test doesn't always indicate that something is significant, and a negative test doesn't necessarily mean that something isn't significant.

15. Look at your models

For relatively simple models like decision trees, it can also be beneficial to provide visualizations of your models, and most libraries have functions that will do this for you. For complex models, like deep neural networks, consider using explainable AI (XAI) techniques to extract knowledge.

Conclusion

Often without realizing it, we might develop in our everyday life an unconscious heuristic approach to working with AI and with those tools that make use of AI. Making an uncritical use of AI tools would risk discrediting a whole area of new technology. By using the framework suggested in this study, interested parties, without being developers, can become qualified to assess AI tools with confidence.

By asking the right questions based on a framework and implementing the suggested heuristics, those responsible for recommending and assisting in the take-up of AI tools can ensure a much higher success rate for this technology.

References

[1]. Sebbaq, H., & Faddouli, N. E. E. (2021, January). MOOCs semantic interoperability: Towards unified and pedagogically enriched model for building a linked data repository. *International Conference on Digital Technologies and Applications*. Springer, 621-631.

[2]. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805. https://arxiv.org/abs/1810.04805

[3]. Abduljabbar, D. A., & Omar, N. (2015). Exam questions classification based on Bloom's taxonomy cognitive level using classifiers combination. *Journal of Theoretical and Applied information Technology*, *78*(3), 447. http://www.jatit.org/volumes/Vol78No3/15Vol78No3.pdf

[4]. Conole, G. (2014, April). The 7Cs of learning design: A new approach to rethinking design practice. *Proceedings of the Ninth international Conference on Networked Learning* (pp. 502-509). Edinburgh, Scotland. https://www.lancaster.ac.uk/fss/organisations/netlc/past/nlc2014/abstracts/pdf/conole.pdf

[5]. Davis, D., Seaton, D., Hauff, C., & Houben, G. J. (2018, June). Toward large-scale learning design: Categorizing course designs in service of supporting learning outcomes. *Proceedings of the Fifth Annual ACM Conference on Learning at Scale* (pp. 1-10). https://doi.org/10.1145/3231644.3231663

[6]. Kopp, M., & Lackner, E. (2014). Do MOOCs need a special instructional design? *Proceedings of Sixth international Conference on Education and New Learning* (EDULEARN14; pp. 7138-7147). Barcelona, Spain. https://library.iated.org/view/KOPP2014DOM

[7]. Molenda, M. (2003). in search of the elusive ADDiE model. *Performance improvement*, *42*(5), 34-37. http://www.damiantgordon.com/Courses/DT580/in-Search-of-Elusive-ADDiE.pdf

[8]. Osman, A., & Yahya, A. A. (2016). Classifications of exam questions using linguistically-motivated features: A case study based on Bloom's taxonomy. https://www.researchgate.net/publication/298286164_Classifications_of_Exam_Questions_Using_Linguistically-Motivated_Features_A_Case_Study_Based_on_Blooms_Taxonomy

[9]. Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980. https://arxiv.org/abs/1412.6980

[10]. Major, C. H., & Blackmon, S. J. (2016). Massive open online courses: Variations on a new instructional form. *New Directions for institutional Research*, *2015*(167), 11-25. https://doi.org/10.1002/ir.20151

[11]. Das, S., Das Mandal, S. K., & Basu, A. (2020). identification of cognitive learning complexity of assessment questions using multi-class text classification. *Contemporary Educational Technology*, *12*(2), Article ep275. https://doi.org/10.30935/cedtech/8341

[12]. Haris, S. S., & Omar, N. (2012, December). A rule-based approach in Bloom's taxonomy question classification through natural language processing. *Seventh international Conference on Computing and Convergence Technology* (iCCCT; pp. 410-414). institute of Electrical and Electronics Engineers. https://ieeexplore.ieee.org/abstract/document/6530368

[13]. Conole, G. (2016). MOOCs as disruptive technologies: Strategies for enhancing the learner experience and quality of MOOCs. *Revista de Educación a Distancia*, *50*(2). http://dx.doi.org/10.6018/red/50/2

[14]. González-Carvajal, S., & Garrido-Merchán, E. C. (2020). *Comparing BERT against traditional machine learning text classification*. arXiv preprint arXiv:2005.13012. https://arxiv.org/abs/2005.13012

[15]. Merrill, M. D. (2012). *First principles of instruction*. John Wiley & Sons. https://digitalcommons.usu.edu/usufaculty_monographs/100/

[16]. Margaryan, A., Bianco, M., & Littlejohn, A. (2015). instructional quality of massive open online courses (MOOCs). *Computers & Education, 80,* 77-83. https://doi.org/10.1016/j.compedu.2014.08.005

[17]. Xing, W. (2019). Exploring the influences of MOOC design features on student performance and persistence. *Distance Education, 40*(1), 98-113. https://doi.org/10.1080/01587919.2018.1553560

[18]. Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into Practice, 41*(4), 212-218. https://doi.org/10.1207/s15430421tip4104_2

[19]. Nevid, J. S., & McClelland, N. (2013). Using action verbs as learning outcomes: Applying Bloom's taxonomy in measuring instructional objectives in introductory psychology. *Journal of Education and Training Studies, 1*(2), 19-24. https://doi.org/10.11114/jets.v1i2.94

[20]. Omar, N., Haris, S. S., Hassan, R., Arshad, H., Rahmat, M., Zainal, N. F. A., & Zulkifli, R. (2012). Automated analysis of exam questions according to Bloom's taxonomy. *Procedia: Social and Behavioral Sciences, 59*, 297-303. https://doi.org/10.1016/j.sbspro.2012.09.278

[21]. Mohammed, M., & Omar, N. (2020). Question classification based on Bloom's taxonomy cognitive domain using modified TF-iDF and word2vec. *PLOS ONE 15*, e0230442. https://doi.org/10.1371/journal.pone.0230442

[22]. Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for multi-class classification: An overview*. arXiv preprint arXiv:2008.05756. https://arxiv.org/abs/2008.05756

[23]. Pardos, Z. A., & Schneider, E. (2013). *AiED 2013 workshops proceedings* (Vol. 1). http://people.cs.pitt.edu/~falakmasir/docs/AiED2013.pdf

[24]. Quintana, R. M., & Tan, Y. (2019). Characterizing MOOC pedagogies: Exploring tools and methods for learning designers and researchers. *Online Learning*, *23*(4), 62-84. https://doi.org/10.24059/olj.v23i4.2084

[25]. Rosselle, M., Caron, P. A., & Heutte, J. (2014, February). A typology and dimensions of a description framework for MOOCs. in *Proceedings of European MOOCs Stakeholders Summit 2014,* (eMOOCs 2014; pp. 130-139). Lausanne, Switzerland. Proceedings document published by Open Education Europa (www. Open education europa. eu). https://hal.archives-ouvertes.fr/hal-00957025/

[26]. Sebbaq, H., El Faddouli, N. E., & Bennani, S. (2020, September). Recommender system to support MOOCs teachers: Framework based on ontology and linked data. *Proceedings of the 13th international Conference on intelligent Systems: Theories and Applications,* Article 18. https://doi.org/10.1145/3419604.3419619

[27]. Ting Fei, Wei Jyh Heng, Kim Chuan Toh, & Tian Qi. (2003). Question classification for e-learning by artificial neural network. *Proceedings of the Joint Fourth international Conference on information, Communications and Signal Processing, and the Fourth Pacific Rim Conference on Multimedia* (pp. 1757-1761). institute of Electrical and Electronics Engineers. Singapore. https://doi.org/10.1109/iCiCS.2003.1292768

[28]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, i. (2017). Attention is all you need. *31st Conference on Neural information Processing Systems* (NiPS 2017; pp. 5998-6008). Long Beach, USA. https://arxiv.org/abs/1706.03762

[29]. Yousef, A. M. F., Chatti, M. A., Schroeder, U., & Wosnitza, M. (2014, July). What drives a successful MOOC? An empirical examination of criteria to assure design quality of MOOCs. *14th international Conference on Advanced Learning Technologies* (pp. 44-48). institute of Electrical and Electronics Engineers. https://doi.org/10.1109/iCALT.2014.23

[30]. Yusof, N., Hui, C. J. (2010). Determination of Bloom's cognitive level of question items using artificial neural network. *10th international Conference on intelligent Systems Design and Applications* (iSDA; pp. 866-870). institute of Electrical and Electronics Engineers. Cairo, Egypt. https://doi.org/10.1109/iSDA.2010.5687152