Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

Intelligent Flashcard Generation using Natural Language Processing and Spaced Repetition for Personalized Learning

S Catherine Grace[1]
Student,
Department of BCA,
St. Francis College,
Bengaluru, India.

Mary Stella F[2]
Assistant Professor,
Department of BCA,
St. Francis College,
Bengaluru, India.

***Abstract*— Due to the rapid development of digital learning content, there is a need for the development of intelligent systems which would assist learners in organizing and revising educational materials. This paper describes the development of the AI-based smart flashcard application for effective learning. It allows students and self-learners to create interactive flashcards automatically from different learning sources such as PDF documents, Word documents, and text files. For identification of relevant information and key concepts, Natural Language Processing techniques such as text summarization, keyword extraction, and clustering are used. The flashcards and multiple-choice questions are created by the system based on the information extracted from the provided learning material. Thus, it saves time for creating study materials. In order to facilitate knowledge retention, the system uses spaced repetition technique for reviewing difficult flashcards. In addition, learners have an opportunity to edit or delete flashcards using an interactive dashboard and analyze their learning process with the help of graphs and charts. Also, it offers some additional learning functions such as quizzes, practice test, and summaries of the studied information. Flashcards can be exported to the learning platforms, for instance, Quizlet or Anki. Using Artificial Intelligence, NLP techniques, and spaced repetition, this system will help the learners to identify the key concepts and reinforce their understanding of the information. Thus, it facilitates examination preparation.**

***Keywords*— *Artificial Intelligence, Smart Flashcards, Natural Language Processing, Spaced Repetition, Learning Efficiency*.**

I. INTRODUCTION

Students in universities and professional programmes now deal with more digital material than they can hold in mind at once. Course readers, journal articles, technical manuals and lecture slides compete for the same limited working memory [1]. Keeping up means condensing and consolidating several subjects in parallel while also meeting regular coursework deadlines.

Cognitive psychology offers a well-tested answer. Retrieving information from memory, instead of rereading or highlighting it, produces stronger and longer-lasting learning [2]–[4]. This practice, known as active recall or retrieval practice, is thought to work because each successful retrieval strengthens the memory trace that produced it.

Retention improves further when retrieval attempts are spread over gradually widening intervals. This spacing effect counteracts the exponential forgetting first measured by Ebbinghaus [5], [6]: a review placed shortly before a memory becomes inaccessible flattens the decay curve far more efficiently than massed repetition [7], [8]. Retrieval practice and spacing together form the backbone of effective self-regulated study.

Yet many students do not apply these methods consistently, and one reason is the cost of preparing the material. Producing a good deck by hand means reading the source, picking out key propositions, rewording definitions and formatting question–answer pairs, which can absorb hours that could have been spent on recall itself. This overhead is the bottleneck that our work targets.

We present a complete system that takes raw study files as input and delivers scheduled review sessions as output. It cleans and parses documents in several formats, extracts core concepts with a hybrid NLP model, generates cloze-deletion and multiple-choice items automatically, and schedules every card with an algorithm derived from SuperMemo-2 (SM-2) [9]. A dashboard tracks mastery and memory-decay trends for each learner. The contributions of this paper are:

1) a multi-format ingestion and cleaning engine that yields clean running text from PDF, DOCX, plain-text and scanned inputs;
2) a hybrid concept extractor that combines KeyBERT embeddings [10] with TextRank graph ranking [11];
3) an item generator that produces cloze-deletion cards and multiple-choice questions whose distractors are drawn from WordNet [12]; and
4) a 12-week controlled study with 120 undergraduate students that measures preparation time, examination performance and 90-day retention.

The remainder of the paper is organised as follows. Section II gives the learning-science background, Section III reviews related work, Sections IV–VI describe the architecture, the scheduler and the implementation, Section VII reports results,

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

Section VIII discusses limitations and ethical aspects, and Sections IX and X conclude and outline future work.

II. BACKGROUND

An effective study platform needs a sound model of how people learn as well as the right language-processing tools. This section covers the learning-science concepts that motivate the design of our system.

A. Cognitive Load and Deck Authoring

Cognitive Load Theory (CLT), introduced by Sweller [1], starts from the observation that working memory can process only a few novel elements at a time. It separates the load on a learner into three kinds [13]:

- **Intrinsic load:** the inherent complexity of the topic, which cannot be reduced without simplifying the subject matter.
- **Germane load:** the effort devoted to building schemas and encoding knowledge in long-term memory.
- **Extraneous load:** effort wasted on poorly designed instruction or tedious side tasks.

Copying notes and formatting flashcards by hand is a clear source of extraneous load. It tires the learner before any retrieval practice begins, and hand-built decks can suffer from inconsistent formatting and ambiguous prompts that confuse later review. Automating parsing, concept extraction and question formulation removes this load and leaves more mental capacity for germane processing, that is, for understanding concepts and building schemas.

B. Modelling Memory Retention

Ebbinghaus showed that retention declines roughly exponentially with the time since learning [5], and later replications support the same overall shape [6]. The decline is steepest immediately after the first exposure. The probability of retention can be written as

$$R(t) = e^{-t/S} \tag{1}$$

where t is the time since the last review and S is the memory strength. A larger S means a more durable trace and a slower decline. Each successful retrieval increases S, so the curve flattens with every review, as Fig. 1 illustrates for a hypothetical card. An adaptive scheduler can therefore estimate the window in which a review yields the greatest reinforcement per minute of study.

C. Automated Item Generation

Automated item generation (AIG) uses templates or language models to produce assessment items at scale [14]. A flashcard is a minimal item with a prompt on the front and an answer on the back. Our system generates two item types, cloze-deletion cards and multiple-choice questions, and relies on NLP to decide which concepts are worth asking about.

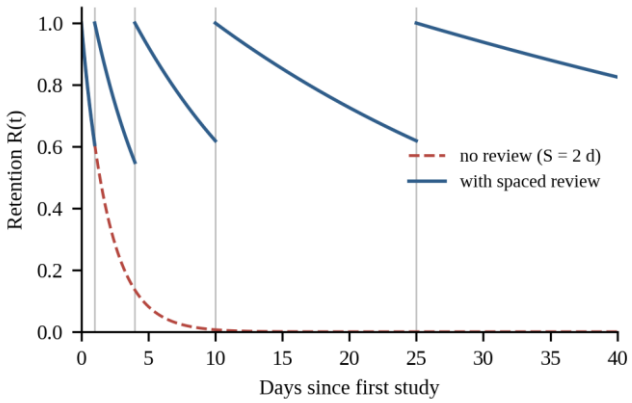


Fig. 1. Illustrative forgetting curves from (1) without review (dashed) and with reviews on days 1, 4, 10 and 25, assuming that S is multiplied by 2.5 after each successful review. The curves are a model illustration, not measured data.

III. RELATED WORK

This section reviews work on educational NLP, keyphrase extraction, question and distractor generation, and computational spaced repetition, tracing the shift from hand-written rules to pre-trained neural models.

A. NLP in Educational Technology

Early educational uses of NLP concentrated on automated essay scoring and grammar checking. As language understanding improved, researchers turned to AIG to help instructors build examinations more quickly [14]. These tools showed that software can lighten teachers' workloads, but student-facing adaptive applications remained comparatively rare.

The first AIG systems relied on hand-written grammatical rules and regular-expression templates. Rule-based generation is precise and cannot hallucinate, yet every new subject area demands considerable engineering effort, and the rules break down on unusual sentence structures and specialised terminology [14].

B. Keyphrase Extraction

Choosing the right concepts is a prerequisite for useful flashcards. Statistical methods such as TF-IDF [15] and RAKE [16] score terms from counts and co-occurrence. They are cheap to run, but they tend to favour generic high-frequency words over essential academic concepts. TextRank [11] added structure by building a graph over the text and ranking its nodes with PageRank-style centrality. KeyBERT [10] goes further and compares candidate phrases with the whole document in a dense embedding space produced by a BERT-family encoder [17], [18], so that technical terms can be recognised even when they occur only a few times.

C. Automated Question Generation

Automated question generation (AQG) turns declarative passages into questions. Early neural sequence-to-sequence models [19] were promising but prone to semantic drift: they could produce fluent questions that contained factual errors or

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

focused on trivial details. Pre-trained transformers such as BART [20], T5 [21] and the GPT family [22], all descended from the transformer architecture [23], produce considerably more coherent and contextually accurate question stems. For multiple-choice items, however, the stem is only half of the task.

D. Distractor Generation

A good distractor is topically relevant but clearly wrong. Randomly chosen wrong answers let learners succeed by elimination rather than by knowledge, so they require no real discrimination between concepts. Effective distractors belong to the same semantic category as the answer or reflect a typical misconception. A common technique uses WordNet [12] to find co-hyponyms, that is, words that share a hypernym with the target. Combining this ontological filter with embedding similarity, for example Word2Vec [24] or GloVe [25], and requiring the same part of speech yields distractors that are plausible and diagnostic.

E. Computational Spaced Repetition

Practical scheduling began with SM-2 [9], which multiplies the review interval by a per-card easiness factor. Data-driven models followed. Settles and Meeder [26] proposed half-life regression, which predicts recall probability from a learner's history and lexical features. Tabibian et al. [27] cast review scheduling as a stochastic optimal-control problem, and Ye et al. [28] formulated it as a stochastic shortest-path problem. These methods are more flexible than SM-2, but they are trained on large collections of review logs, which a new platform does not yet have. We therefore start from SM-2 and keep it as a transparent baseline. Table I summarises the techniques discussed in this section.

TABLE I
COMPARISON OF TECHNIQUES RELEVANT TO FLASHCARD GENERATION

Technique	Principle	Main limitation
TF-IDF [15]	Term-frequency weighting	Favours generic frequent terms
RAKE [16]	Word co-occurrence scores	Ignores semantic context
TextRank [11]	Graph centrality of words	No contextual embeddings
KeyBERT [10]	Embedding similarity to the document	Needs transformer inference
Seq2seq AQG [19]	Encoder–decoder generation	Semantic drift, factual errors
BART / T5 AQG [20], [21]	Pre-trained transformers	Output still needs verification
WordNet distractors [12]	Co-hyponym selection	Limited by lexicon coverage
SM-2 [9]	Per-card easiness factor	Heuristic; not learned per user

IV. SYSTEM ARCHITECTURE

Fig. 2 shows the processing pipeline. Uploaded files pass through a sequence of stages: they are parsed and sanitised, analysed semantically, converted into items and finally stored in a relational database from which the scheduler draws cards.

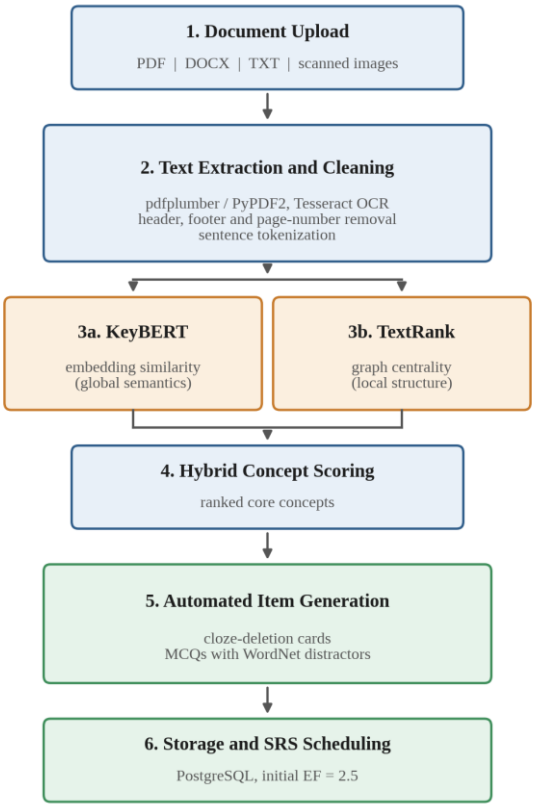


Fig. 2. End-to-end ingestion and processing pipeline.

A. Ingestion and Cleaning

The ingestion stage accepts PDF, DOCX, plain-text and image uploads and normalises their layout. Digital documents are parsed with pdfplumber and PyPDF2. Scanned pages are processed by the Tesseract OCR engine [29] after deskewing and contrast normalisation, which improves character accuracy on low-quality scans. Running headers, footers, page numbers and reference lists are removed so that non-instructional text never enters the item pipeline. The cleaned text is then split into sentences.

B. Hybrid Concept Extraction

Sentences are passed in parallel to the KeyBERT and TextRank extractors. KeyBERT contributes global semantic relevance, because candidates are compared with the embedding of the whole document. TextRank contributes local structural importance, because it measures how central a term is in the sentence graph. Running both ensures that concept scores reflect semantic relevance as well as graph centrality.

C. Item Generation and Storage

The item generator produces two card formats:

- **Cloze-deletion cards:** a key term is replaced by a blank while the surrounding wording of the source passage is preserved.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

- **Multiple-choice cards:** a definition stem is presented with one correct answer and three WordNet-derived distractors [12].

Generated cards are written to a PostgreSQL database with an initial easiness factor (EF) of 2.5, ready for scheduling.

V. Spaced Repetition Scheduling

The platform schedules reviews with a state machine derived from SM-2 [9], shown in Fig. 3, so that intervals adapt to each learner's retention speed and to each card's difficulty. After seeing the back of a card, the learner grades their recall from 0 (complete blackout) to 5 (perfect recall). Table II lists the scale and its effect on the easiness factor.

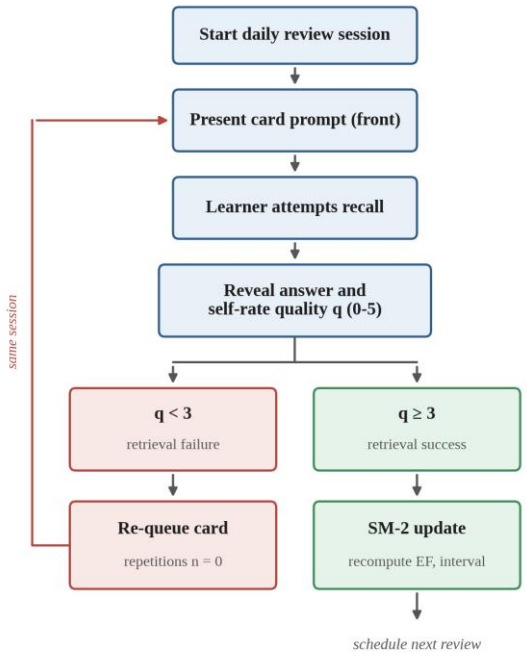


Fig. 3. Spaced-repetition review state machine.

TABLE II
SM-2 Recall Grades and Their Effect

q	Recall quality	ΔEF	Result
5	Perfect response	+0.10	Schedule next review
4	Correct after hesitation	0.00	Schedule next review
3	Correct with serious difficulty	−0.14	Schedule next review
2	Incorrect; answer seemed easy to recall	–	Re-queue, $n = 0$
1	Incorrect; answer seemed familiar	–	Re-queue, $n = 0$
0	Complete blackout	–	Re-queue, $n = 0$

A grade below 3 signals retrieval failure. The card's repetition count n is reset to zero and the card is re-queued for immediate review in the same session, so unmastered items cannot slip through. Grades of 3 or higher update the easiness factor as

$$EF' = EF + 0.1 - (5 - q)(0.08 + 0.02(5 - q)) \quad (2)$$

and the next interval is computed as

$$I(n) = I(n-1) \cdot EF', \quad n > 2 \quad (3)$$

with $I(1) = 1$ day and $I(2) = 6$ days, following SM-2. In the original specification the easiness factor is also bounded below at 1.3. For example, a perfect grade raises EF by 0.10 whereas a grade of 3 lowers it by 0.14, so cards that are repeatedly graded as difficult come back more often. This feedback loop avoids over-reviewing easy items and reinforces fragile ones shortly before they are forgotten [26], [27].

VI. IMPLEMENTATION

The platform follows a decoupled microservice architecture so that heavy document parsing and real-time user interaction can proceed concurrently without interface latency.

A. Technology Stack

- **Frontend:** React with Tailwind CSS, providing responsive review decks, flip animations and analytics dashboards. Local state keeps offline review progress and synchronises it with the backend API after reconnection.
- **Backend:** Python FastAPI, with Celery task queues and Redis message brokers handling document processing in the background, so long parsing jobs never block user HTTP requests.
- **NLP engines:** Hugging Face Transformers [30], KeyBERT, spaCy and NLTK for sentence parsing, embedding computation and lexical extraction.
- **Storage:** PostgreSQL for generated cards and their scheduling state.

The dashboard presents each learner's memory-decay patterns and mastery rate over time, which closes the loop between scheduled practice and visible progress.

VII. RESULTS AND DISCUSSION

We evaluated the system along three dimensions: keyphrase extraction accuracy against traditional baselines, expert ratings of the generated items across disciplines, and learning outcomes in a controlled classroom study.

A. Keyphrase Extraction Benchmark

Table III and Fig. 4 compare the proposed KeyBERT-based extractor with three baselines. KeyBERT reaches an F1 score of 0.77, which is 0.28 above TF-IDF (0.49), 0.24 above RAKE (0.53) and 0.14 above TextRank (0.63). Its ROUGE-L of 0.80 is likewise the highest. The statistical baselines frequently missed key terms that occur only a few times, whereas KeyBERT recognised their contextual importance regardless of frequency [10], [11].

TABLE III
Benchmark Comparison of Extraction Models

Model	Precision	Recall	F1 score	ROUGE-L
TF-IDF	0.48	0.51	0.49	0.52
RAKE	0.52	0.54	0.53	0.55
TextRank	0.61	0.65	0.63	0.66
KeyBERT (ours)	**0.76**	**0.78**	**0.77**	**0.80**

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

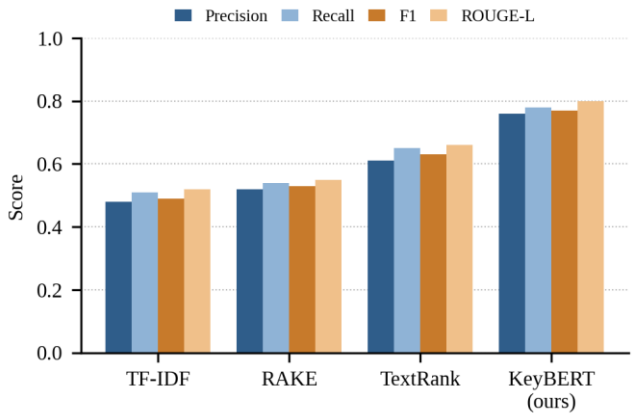


Fig. 4. Extraction accuracy of the compared models (data from Table III).

B. Expert Quality Assessment

Domain experts rated the generated items on a 1–5 scale in computer science, biology and history (Table IV, Fig. 5). Factual correctness received the highest mean rating (4.80) and distractor quality the lowest (4.40), which is consistent with the difficulty of choosing plausible but incorrect options discussed in Section III-D. Ratings stayed within 0.2 points across the three disciplines: biology was rated highest for correctness, fluency and distractor quality, and history highest for educational utility. The factual-correctness score supports the reliability of the item-generation pipeline.

TABLE IV
EXPERT QUALITY ASSESSMENT (1–5 SCALE)

Dimension	CS	Biology	History	Mean score
Factual correctness	4.8	4.9	4.7	4.80
Grammatical fluency	4.6	4.7	4.5	4.60
Distractor quality	4.4	4.5	4.3	4.40
Educational utility	4.5	4.4	4.6	4.50

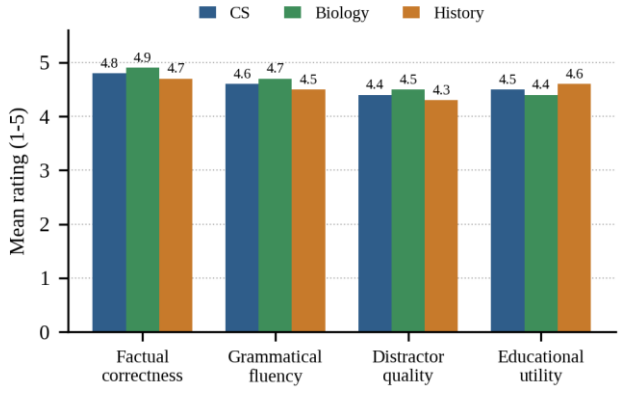


Fig. 5. Expert ratings by evaluation dimension and discipline (data from Table IV).

C. Controlled Classroom Trial

A 12-week controlled study involved 120 undergraduate computer-science students, split equally into a control group (n = 60) and an experimental group (n = 60). Both groups received the same syllabus, lecture slides and textbook readings. The control group studied with conventional methods (manual note-taking, highlighting and self-authored flashcards), while the experimental group used the proposed platform. Table V and Fig. 6 summarise the outcomes.

TABLE V
12-WEEK STUDENT TRIAL METRICS

Metric	Control	Experimental	Net Δ
Weekly prep time	4.2 h/wk	1.8 h/wk	−57.1%
Weekly quiz mean	74.2%	88.5%	+14.3 pts
Final exam score	72.4%	90.9%	+18.5 pts
90-day concept retention	51.3%	84.2%	+32.9 pts

pts = percentage points.

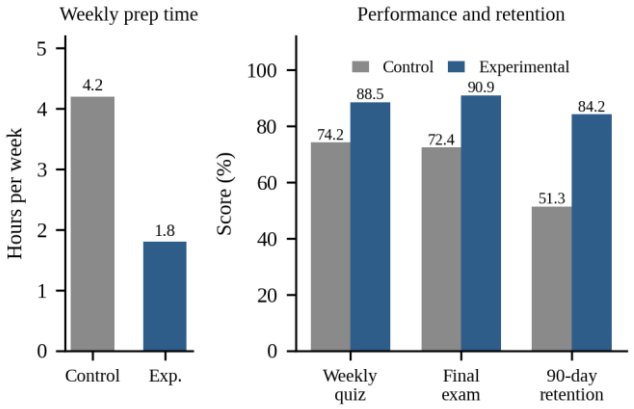


Fig. 6. Preparation time (left) and learning outcomes (right) for the two groups (data from Table V).

Experimental students spent 57.1% less time on weekly preparation (1.8 versus 4.2 hours) and scored 18.5 percentage points higher in the final examination (90.9% versus 72.4%), a relative gain of about 26%. Weekly quiz means were 14.3 points higher. In an unannounced retention test 90 days after the course, the experimental group retained 84.2% of concepts against 51.3%, a difference of 32.9 percentage points, or about 64% in relative terms. These figures suggest that cutting the cost of deck creation, combined with scheduled retrieval practice, carries over into longer-lasting learning [2], [3].

D. Elimination of Creation Friction

The main reason for the gains appears to be the removal of creation friction. In survey responses, control-group students described mental burnout during deck authoring that led to irregular study schedules, and many abandoned flashcards altogether in the middle of the semester because of time pressure. Experimental students, by contrast, had a finished, structured deck within seconds of finishing a lecture and could use short idle periods, such as commutes or breaks, for brief review sessions. The shift from occasional, stressful cramming to steady daily recall is the most plausible mechanism behind the examination gains [3], [4].

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

VIII. LIMITATIONS AND ETHICAL CONSIDERATIONS

A. System Limitations

- **OCR degradation:** heavily noisy scans can introduce character errors into generated card prompts.
- **Mathematical notation:** dense proofs and equations need a dedicated LaTeX-aware parser to keep their formatting intact.
- **Scope of assessment:** flashcards are well suited to atomic recall, whereas synthesis and argumentative writing require complementary open-ended assessment tools.

B. Validity Considerations

The trial involved a single cohort of undergraduate computer-science students, so replication in other disciplines and institutions is needed before the effect sizes are generalised. Because the control group used conventional self-study rather than another software tool, the comparison measures the complete workflow, namely automated generation together with adaptive scheduling, and cannot separate the contribution of each component. Survey responses on study habits may also reflect subjective impressions rather than measured behaviour.

C. Ethical and Privacy Considerations

Because learners upload their own material, the platform may process copyrighted texts or documents that contain personal information. Deployments should therefore restrict access to each learner's own files, encrypt stored content and support deletion on request. Automatically generated cards can contain errors, so each card should link back to its source passage and remain editable by the learner, and generated items should be presented as study aids rather than authoritative statements.

IX. CONCLUSION

This paper presented an intelligent flashcard generation system that unifies multi-format document parsing, hybrid KeyBERT–TextRank concept extraction, automated cloze and multiple-choice item generation, and SM-2-based adaptive scheduling. By removing the manual authoring step, the platform lets learners spend their time on retrieval practice. In a 12-week controlled study the experimental group reduced weekly preparation time by 57.1%, scored 18.5 percentage points higher in the final examination and retained 32.9 percentage points more concepts after 90 days. Together with the extraction benchmark and the expert ratings, these findings indicate that automated item generation can narrow the gap between cognitive-science recommendations and everyday study habits.

X. FUTURE WORK

Future research will proceed in four directions:

- **Multimodal extraction:** vision-language models such as CLIP [31] could turn diagrams, charts and chemical structures in textbooks into visual flashcards.
- **Personalised decay modelling:** the fixed SM-2 parameters could be replaced by learned models, for example reinforcement-learning-based schedulers [28] trained on review logs [26], [27], to fit each learner's own memory curve.
- **Generative explanation:** an AI tutoring back end could explain the correct answer immediately whenever a learner marks a card as failed [22].
- **Broader validation:** larger multi-institution trials across several disciplines, with inferential statistics, would test how well the observed gains generalise.

REFERENCES

[1] Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive Science, 12(2), 257–285.

[2] Roediger, H. L., & Karpicke, J. D. (2006). Test-enhanced learning: Taking memory tests improves long-term retention. Psychological Science, 17(3), 249–255.

[3] Karpicke, J. D., & Blunt, J. R. (2011). Retrieval practice produces more learning than elaborative studying with concept mapping. Science, 331(6018), 772–775.

[4] Dunlosky, J., Rawson, K. A., Marsh, E. J., Nathan, M. J., & Willingham, D. T. (2013). Improving students' learning with effective learning techniques. Psychological Science in the Public Interest, 14(1), 4–58.

[5] Ebbinghaus, H. (1913). Memory: A Contribution to Experimental Psychology (H. A. Ruger & C. E. Bussenius, Trans.). Teachers College, Columbia University. (Original work published 1885).

[6] Murre, J. M. J., & Dros, J. (2015). Replication and analysis of Ebbinghaus' forgetting curve. PLoS ONE, 10(7), e0120644.

[7] Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., & Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. Psychological Bulletin, 132(3), 354–380.

[8] Kang, S. H. K. (2016). Spaced repetition promotes efficient and effective learning: Policy implications for instruction. Policy Insights from the Behavioral and Brain Sciences, 3(1), 12–19.

[9] Wozniak, P. A. (1990). Optimization of learning. Master's thesis, University of Technology in Poznań.

[10] Grootendorst, M. (2020). KeyBERT: Minimal keyword extraction with BERT. Zenodo.

[11] Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 404–411.

[12] Miller, G. A. (1995). WordNet: A lexical database for English. Communications of the ACM, 38(11), 39–41.

[13] Sweller, J., van Merriënboer, J. J. G., & Paas, F. G. W. C. (1998). Cognitive architecture and instructional design. Educational Psychology Review, 10(3), 251–296.

[14] Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. International Journal of Artificial Intelligence in Education, 30(1), 121–204.

[15] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513–523.

[16] Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. In Text Mining: Applications and Theory (pp. 1–20). Wiley.

[17] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proc. NAACL-HLT, 4171–4186.

[18] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proc. EMNLP-IJCNLP, 3982–3992.

[19] Du, X., Shao, J., & Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. Proc. 55th Annual Meeting of the ACL, 1342–1352.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

[20] Lewis, M., Liu, Y., Goyal, N., et al. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Proc. 58th Annual Meeting of the ACL, 7871–7880.

[21] Raffel, C., Shazeer, N., Roberts, A., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1–67.

[22] Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901.

[23] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008.

[24] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[25] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. Proc. EMNLP, 1532–1543.

[26] Settles, B., & Meeder, B. (2016). A trainable spaced repetition model for language learning. Proc. 54th Annual Meeting of the ACL, 1848–1858.

[27] Tabibian, B., Upadhyay, U., De, A., Zarezade, A., Schölkopf, B., & Gomez-Rodriguez, M. (2019). Enhancing human learning via spaced repetition optimization. Proceedings of the National Academy of Sciences, 116(10), 3988–3993.

[28] Ye, J., Su, J., & Cao, Y. (2022). A stochastic shortest path algorithm for optimizing spaced repetition scheduling. Proc. 28th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, 4381–4390.

[29] Smith, R. (2007). An overview of the Tesseract OCR engine. Proc. 9th Int. Conf. on Document Analysis and Recognition (ICDAR), 629–633.

[30] Wolf, T., Debut, L., Sanh, V., et al. (2020). Transformers: State-of-the-art natural language processing. Proc. EMNLP: System Demonstrations, 38–45.

[31] Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. Proc. 38th Int. Conf. on Machine Learning (PMLR 139), 8748–8763.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)