A General Multimodal Teaching Video Generation System Based on Storyboard Prediction for Chinese Educational Corpora

Ying Tian[1], Yuhe Ding[1], Wanyue Liu[1,2,*]
[1]School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China
[2]iFLYTEK Co., Ltd., Hefei, Anhui, China
**Corresponding Author*

Abstract: This paper analyzes how to build a video generation system with the help of RF-MTTextCNN model, and can translate the text content into a series of storyboards. In this system, six attributes can be predicted, including scene, difficulty and so on. The existence of these attributes can help the model generate storyboards, adjust the duration, and ensure that the animation rendering can be carried out smoothly. In this study, the system is tested with 4991 samples, and the three random seeds are introduced to find that the system's accuracy reaches 0.837, which is 0.061 higher than the existing model, and the macro-F1 value is 0.788, which is 0.196 higher than the existing model. Three test videos have H.264 video and AAC audio tracks, with a time difference of less than 0.02 seconds. The research results show that the system has strong engineering stability, good interpretability and strong controllability. However, the definition of weak labels is the same as that of rule experts, and the results are only consistent internally, but not consistent with the data marked by human beings.

Keywords: Teaching Video Generation; Storyboard Prediction; Rule Fusion; Multitask Learning; TextCNN; Audio-Video Synchronization

1. Introduction

In the teaching process, teachers will use teaching videos, integrate the concepts they have mastered, use sound, images and other content, set a reasonable time, and use this way to explain the algorithm and formula to students, which can get good results. In the previous research, we deeply realized the value of multimedia teaching, project-based teaching, micro-lessons and so on, which can play an important role in the teaching of network design, programming and so on. However, many teachers need to find materials, design storyboards and produce videos manually, which requires a lot of costs and is not conducive to the reuse of teaching resources.

At present, the application of generative model in the field of video synthesis has achieved positive results, but only to generate a series of visual content can not ensure that the generated video has a strong logic and a large amount of knowledge. The current text-to-video method focuses on visual quality and temporal consistency, but the video content should be updated regularly, the description should be accurate, the rhythm should be appropriate, and the output should be verified. The use of large language model to generate scripts randomly may lead to repeated shots, some fields are not covered, and there are inconsistencies in the visual type. Therefore, the video system must be built on the basis of strong generation ability, and the constraint structure should have strong trainability, which should be evaluated objectively.

This paper has achieved three breakthroughs. First, the educational text of China is transformed into six weakly supervised tasks, namely code requirement, formula requirement, duration requirement, difficulty, visual type and scene type, and the difference between deterministic structural attributes and semantic attributes is analyzed. Second, RF-MTTextCNN is put forward, which can not only use multi-scale convolutional features to predict semantics, but also take into account the code, formula and duration, so that the definition of neural model does not need to be adjusted.Third, the pipeline is built, which can realize text-to-speech, storyboard generation, HTML/SVG rendering and so on. In the process of verification, the video-track information, JSON file and other

contents are retained. It should be made clear that the purpose of this research is not to generate pixels, but to make teaching videos, which can be played and the process is controllable.

In the above research, we regard the six labels as different classification tasks, which will not be mixed. In the process of learning, the visual type should be combined with the specific scene, and the corresponding intent should be expressed, which may lead to prediction errors. The duration level is set according to the length threshold, and the code and formula meet the symbolic pattern. After separating the two categories, the business logic can be interpreted.

2. Related Work

This section reviews the strands most relevant to the proposed system: teaching-media production, controllable video generation, lightweight text classification, and weak-supervision labeling. The aim is to identify the specific gaps in storyboard controllability, traceability, and engineering validation that motivate the present design.

2.1 Teaching Multimedia and Automatic Video Generation

Traditional research on teaching multimedia mainly addresses resource forms and classroom organization. Within computer education practice, multimedia-supported instruction and practical video-rendering workflows have been discussed as ways to make content presentation more concrete and reproducible [1,2]. Their advantage is that teaching objectives are clear and content can be checked by teachers; their limitation is that production depends on manual work and it is difficult to automatically obtain consistent storyboard structures from large-scale course text. This limitation corresponds to the end-to-end structured pipeline proposed in this paper.

Recent text-to-video research includes Make-A-Video, VideoPoet, CogVideoX, and video diffusion models [3-6]. These methods have made progress in open-domain visual generation, but they usually do not directly output educational scene labels, knowledge points, narration, formula requirements, and editable pages; therefore, they cannot easily answer why a particular shot explains the content in a particular way. This paper does not compare against them using FVD or CLIP scores on heterogeneous datasets. Instead, it treats them as part of the visual generation lineage and places all experimental comparisons on the same Chinese storyboard dataset.

2.2 Text Classification and Multitask Learning

Foundational work on multitask learning uses shared representations to provide inductive transfer across related tasks [7]. Kim showed that convolutional networks can effectively extract local sentence patterns [8], and Vaswani et al. proposed the Transformer to model long-range dependencies through self-attention [9]. Transformers are highly expressive, but with 4,991 weakly labeled samples and 8 GB of GPU memory, model capacity is not the only goal. A lightweight TextCNN trains quickly and has clear task heads, making it more suitable as an interpretable constraint module for this system. Existing classification studies often predict only a single label, whereas this paper simultaneously predicts what function to teach, what visual form to use, how long to explain, how difficult it is, and whether formulas or code are needed. This is exactly where the multitask structure is used.

2.3 Chinese Educational Corpora and Weak Supervision

The OpenCSG Chinese Corpus provides a series of high-quality Chinese datasets for large language model training. Such corpora contain educational text features such as concept explanations, step descriptions, and formula statements, but they do not naturally include video storyboard labels. This paper uses rules and regular expressions to construct weak labels, following the general principle of transforming heuristic labeling functions into trainable signals [10]. The advantage is low cost and reproducibility; the limitation is also direct: the model may learn keyword rules rather than genuine instructional design. Therefore, the experiments include a character TF-IDF baseline, category-distribution analysis, and confusion matrices, and they identify weak-label bias as the primary limitation in the conclusion.

In the process of weak supervision, we usually use heuristic functions to achieve the purpose of training unlabeled samples, but the existence of heuristic functions may lead to conflicts and result in coverage bias. The case analyzed in this paper is relatively typical: some labels are not generated by rules, but have the same

engineering definition in the system. For example, the setting of short, medium and long levels is not the evaluation of teachers, but an important variable that needs to be controlled in the process of speech synthesis and layout, and rule experts need to be arranged in this process, which can not be avoided, but it can make the control interface, inference logic and label source consistent. The task of external generalization and validation is semantic, such as scene, visual type and difficulty.

3. Method

The method is to integrate weakly supervised storyboards with deterministic video assembly, set labels as the starting point, introduce a rule-based multitask predictor, and use speech to drive the duration back.

3.1 Overall Pipeline

Input course text is first cleaned, segmented, and weakly labeled. RF-MTTextCNN outputs six storyboard attributes: the semantic branch provides scene, visual type, and difficulty predictions, while the rule experts output duration, formula requirement, and code requirement. These attributes and the original text are jointly fed into a constrained large language model to generate a Storyboard conforming to a JSON Schema. Then the page module converts visual_type and visual_plan into HTML, SVG, and GSAP animations; the speech module synthesizes Chinese narration for each storyboard shot; finally, the page duration is written back according to the actual speech duration, and HyperFrames and FFmpeg package the result as an MP4 file. The overall system pipeline is shown in Figure 1.

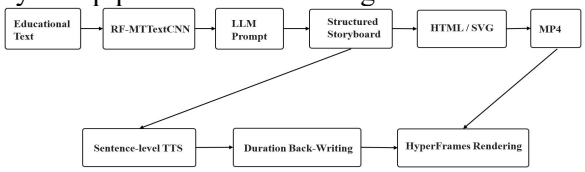

Figure 1. Multimodal Teaching Video Generation Pipeline Driven by Chinese Educational Text

Let the i-th storyboard shot be S_i, containing a title, scene type, visual type, knowledge point, narration, visual plan, and duration. The complete script is not free text but an ordered structure:

$$S = [S_1, S_2, ..., S_m] \tag{1}$$

This structured representation connects model prediction with generation and rendering: the former provides discrete constraints, while the latter adds natural language and visual details. If any module fails, the problem can also be located layer by layer through the Storyboard, pages, audio, and timeline.

3.2 Weakly Supervised Storyboard Label Construction

Table 1. Six Storyboard Prediction Tasks

Task	Example-Valuese	Role
scene typee	concept-introduction/-formula explanation/-algorithm flow/-case /-code /-comparison/ summarye	Determines the-pedagogical functionof a storyboard shote
visual typee	concept cards/-curve/-flowchart/-network/matrix/.code trace-/-bar chart/ timelinee	Selects-theright-side. visualization structure
duration level	short/-medium-/-longe	Provides-an-initialduration priore
difficultye	basic/intermediate/ advancede	Constrainsterminology and.explanation depthe
need formulae	0/1	Controls the formulaarea
need codee	0/1	Controls.the-code area

There are 10 Parquet shards in the local data. The text is segmented by Chinese punctuation into fragments of 80-420 Chinese characters, and 4991 items are retained after deduplication. Scene_type and visual_type are selected according to keyword scores. Need_formula and need_code are judged according to regular expressions for formulas and code. Duration_level is divided according to length threshold. According to the terms proof, derivation, complexity, training, etc., the difficulty is divided. The j-th weak label can be written as:

$$y_j = g_j(x; R_j), j = 1, ..., 6 \tag{2}$$

Among them, g_j is the rule function, and R_j is a set of keywords, length thresholds or regular expressions. It should be made clear that these labels are only weak supervision signals, not gold standard. It can be used for engineering pipeline and structural verification, but it can not be used as a substitute for visual expression and instruction.

In the process of completing the six tasks, the class distribution is not balanced. For example, in the scene_type task, the number of samples of formula_explain is only 232, while the number of samples of concept_intro is 1429. In the

visual_type task, the number of samples of concept_cards is 1305, while the number of samples of matrix is only 148. The imbalance of the duration_level task is the most prominent, with 4,646, 282 and 63 samples respectively, which means that the performance of the majority class will be overestimated if only the accuracy is considered. Therefore, this paper introduces macro-F1 and analyzes the error by normalizing the confusion matrix. Table 1 summarizes the six storyboard prediction tasks.

3.3 RF-MTTextCNN Storyboard Prediction

The network structure is shown in the figure below. After a certain sequence x is input into the 128-dimensional embedding layer, it is processed by three branches with kernel widths of 3, 5 and 7 respectively, and semantic segments, clauses and phrases are captured respectively. The convolution feature can be expressed as follows:

$$c_t = \mathrm{ReLU}(W_k \bullet E_{t:t+k-1} + b_k) \tag{3}$$

Each branch performs global max pooling and is then concatenated:

$$z_k = \max(c_t, z = [z_3; z_5; z_7]) \tag{4}$$

The six tasks share the embedding layer and convolutional encoder, but each uses an independent linear head:

$$p_j = \mathrm{softmax}(w_j z + b_j) \tag{5}$$

$$L = \sum_{j=1}^{6} \lambda_j CE(y_j, p_j) \tag{6}$$

The purely neural branch is first jointly trained on the six weak labels, and task-specific gating is then applied at inference. Let J_s = {scene_type, visual_type, difficulty} denote the set of semantic tasks and J_r = {duration_level, need_formula, need_code} denote the set of structural tasks. The final output can be written as:

$$\hat{y}_j = \begin{cases} \arg\max p_{jc} & j \in J_s \\ g_j(x; R_j) & j \in J_r \end{cases} \tag{7}$$

In the process of formulating the design plan of the model, we need to deepen our understanding of its architecture and divide it into two components, which are deterministic rule experts and learned semantic prediction modules. After entering the character sequence, the matrix is embedded to form a vector representation of each character in a continuous manner. Through the parallel processing of three one-dimensional convolutional layers, the kernel sizes are 3, 5 and 7 respectively, which can capture the local pattern in a larger context. Through global maximum pooling, the extracted features can be concentrated, and the semantic task head can be used to predict the difficulty, visual type and scene type. At the same time, the input data is described in the form of structure, which is provided for rule experts, so as to make a reasonable decision on the code requirements and formula requirements. In the process of division, the probability learning model focuses on the semantic attributes, and the structural constraints are defined explicitly. In Figure 2, the structure of RF-MTTextCNN and the relationship between the two parts are described.

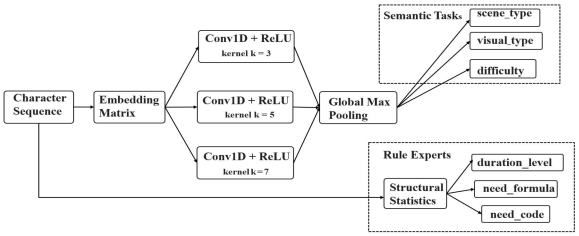

Figure 2. RF-MTTextCNN Structure for Storyboard Attribute Prediction

At the initial stage of the model design, the pooling operation was followed by a fully connected layer, but the ablation test results showed that the addition of this layer reduced the average macro-F1 score of the model by 0.066, which means that the semantic task head can use the pooled representation directly, without the need for additional transformation.The structural experts have no training parameters, and their computational cost comes from two aspects: one is to calculate the sequence length, and the other is to match the regular expressions.Therefore, this design does not guarantee that all tasks can be carried out equally well, but separates the deterministic engineering definition from the probabilistic prediction, and focuses on the learnable capacity of the semantic tasks.

However, the results of this study can not be explained too much. Because all the tags are obtained according to the same rules, the results are close to perfect, which is enough to show that the definition of the rule system is consistent, but can not show that the teacher can assign labels independently. When there are more and more test sets in the future, expert outputs should be evaluated with the help of

neural models, and if there are shortcomings, they should be supplemented or replaced with a learnable rule calibration layer.

3.4 Constrained Storyboard Generation

The predicted labels are not the final script, but prompt constraints. The system writes the course topic, original text, and six predicted labels into the prompt and requires the large language model to generate fixed fields. A JSON Schema checks field types, the number of storyboard shots, and required keys. If visual_type is curve, visual_plan must provide axes, data points, and key positions; if it is network, it must provide nodes, links, and signal flow. Page generation can be abstracted as:

$$P_i = \text{Render}(T_{vis(i)}, V_i) \tag{8}$$

where $T_{vis(i)}$ is the page template selected according to visual type and V_i is the visual_plan. This design does not require the LLM to directly generate uncontrollable video pixels. Instead, it generates checkable page parameters, so teachers or developers can still modify formulas, labels, and animation rhythm.

3.5 Speech Synthesis and Duration Back-Writing

The VITS model proposed by Kim et al. [11] combines variational inference, normalizing flows, and adversarial training, providing a high-quality framework for end-to-end text-to-speech synthesis. This system focuses on engineering synchronization for teaching videos: narration is split by Chinese punctuation such as commas, periods, and semicolons, each sentence is sent to local TTS, and pauses of different lengths are inserted between short sentences. Let a_i be the speech duration of the i-th storyboard shot, v_i be the minimum complete duration of the page animation, and delta be the safety margin. The final storyboard duration is:

$$d_i = \max(a_i + \delta, v_i) \tag{9}$$

$$T = \sum_{i=1}^{m} d_i \tag{10}$$

The biggest advantage of this method is to make up for the shortcomings of the previous course design, the video length is fixed at 60 seconds, the narration time is too long, and the ending part is not played. The system can convert the page frame into H.264 format, the speech into AAC format, and detect the track and time with ffprobe.

4. Experiments and Analysis

This section introduces the system from multiple aspects, using unified data splitting and random seeds. It introduces the experimental protocol, error analysis, and end-to-end MP4 verification.

4.1 Experimental Setup

The experiments use storyboard_dataset.csv from a real course project, containing 4,991 samples. The data are randomly split into training and validation sets at an 8:2 ratio. To reduce the contingency of a single split, three random seeds, 42, 52, and 62, are used. Each group is trained for eight epochs, with a maximum sequence length of 180, a batch size of 64, the AdamW optimizer, a learning rate of 2 * 10^-3, and weight decay of 10^-4. The implementation uses PyTorch 2.8.0+cu129 on Windows with an NVIDIA GeForce RTX 5060 Laptop GPU and 8 GB of memory [12].

The evaluation metrics are Accuracy and Macro-F1 for each task, and the arithmetic mean over the six tasks is also reported. Let the number of classes be C. Macro-F1 is defined as:

$$\text{MacroF1} = \frac{1}{c}\sum_{c=1}^{c}\frac{2P_cR_c}{P_c + R_c} \tag{11}$$

In the process of comparison, all data, indicators, labels and metrics are unified. The TF-IDF + LR baseline uses 2-5 characters and one-vs-rest logistic regression. Single-scale convolutional k = 5. Independent task TextCNN has to build a complete network for each of the six tasks. Both MS-MTTextCNN and RF-MTTextCNN are neural, the latter can be used with three structural experts, and the two comparison results are homologous and comparable.

4.2 Quantitative Comparison

In order to compare the computational efficiency and prediction performance of all models in a unified way, the training time, average accuracy and other contents are listed in Table 2.

Table 2. Baselines and Structural Comparison Under a Unified Setting

Methode	AverageAccuracye	AverageMacro-F1e	ParametersDimensions	TrainingTime/s
Majority Classe	0.624±0.005	0.264±0.001	0	0
Character TF-IDF+LR	0.749±0.006	0.584±0.002	40,000 dimensions	2.23
Single-Scale-Convolution	0.779±0.000	0.586±0.015	748.000	7.13-

Independent-TaskTextCNN	0.772±0.017	0.585±0.020	6,352,000	14.72
MS-MTTextCNN (Ours)	0.776±0.005	0.592±0.014	919,000	6.60
RF-MTTextCNN (Ours)e	0.837±0.003	0.788±0.004	919,000	6.60

As shown in Table 2, RF-MTTextCNN achieves the best performance, with an average accuracy of 0.837 and an average Macro-F1 score of 0.788. Compared with MS-MTTextCNN, it substantially improves both metrics while maintaining the same parameter size and training time. This result indicates that incorporating rule experts can enhance prediction quality without introducing additional trainable parameters or computational costs.The overall comparison of baselines and the proposed variants is presented in Figure 3.

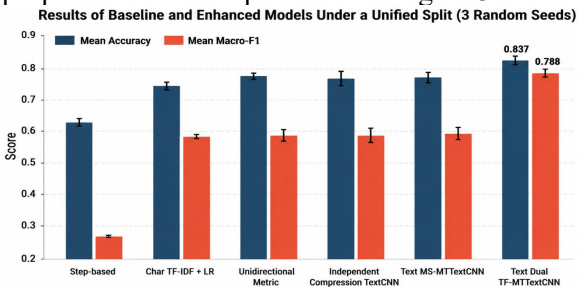

Figure 3. Overall Results of Baseline and Enhanced Models

Figure 3 shows that the performance of the two evaluation indicators of RF-MTTextCNN is relatively ideal, especially in terms of Macro-F1, which can be compared with a variety of storyboard attributes.Compared with MS-MTTextCNN, the method of this paper can better deal with structural definition tasks, and the neural component can get higher semantic prediction.

The most critical reason for the improvement is that the inference mechanism and task definition are unified, and the role of neural encoder is not significant. The Macro-F1 of character TF-IDF+LR is 0.584, which can prove that weak labels have a strong connection with keywords. In other words, the current research results can be used to prove that the system can reproduce the story board control rules, but whether it can achieve the level of teachers in teaching design is not yet clear. When calculating the aggregated indicators, this difference must be taken into account.

4.3 Per-Task Results and Error Analysis

To provide a more detailed analysis of task-level performance, this study compares the Macro-F1 scores of the baseline models and RF-MTTextCNN across individual prediction tasks. The comparison includes both learned semantic tasks and rule-based structural tasks, allowing the contributions of the two components to be examined separately. The task-wise results are presented in Figure 4.

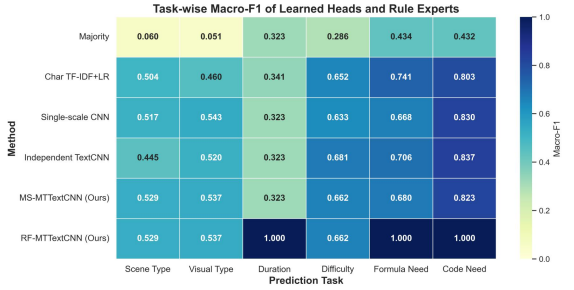

Figure 4. Macro-F1 Comparison by Task

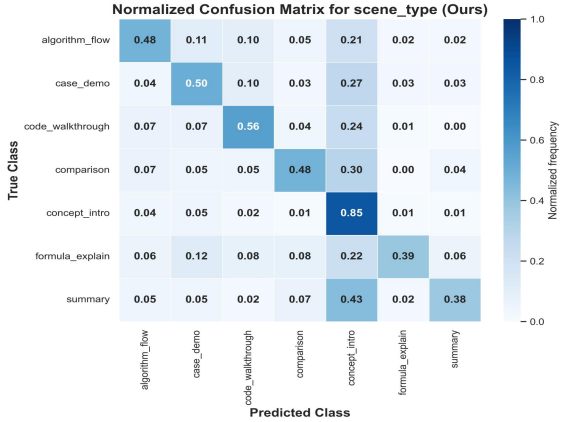

Figure 5. Normalized Confusion Matrix of scene_type

It can be seen from Figure 4 that the RF-MTTextCNN model has achieved a score of 1.000 in the three aspects of need_code, need_formula and duration_level, which is related to the consistency between the definition of weak labels and the opinions of experts. However, this model can only get 0.662 in the aspect of difficulty, 0.537 in the aspect of visual_type and 0.529 in the aspect of scene_type, which is enough to show that this model can accurately judge the open semantic tasks, especially in the aspects of scene and visual types, the class boundary is not clear and overlaps.

Although the metric has improved as a whole, it should be noted that the existence of semantic bottleneck has not changed. In order to find out the source of semantic error, the normalized confusion matrix of scene_type is drawn and shown in the following Figure 5.

From the chart, it can be seen that the recall rate of the concept_intro is 0.85, but there are 43% of the samples, and 22% of the samples are also

included in the concept_intro. The main reason for this phenomenon is that the two concepts overlap, and the labeling rule is relatively weak, which leads to the default of concept introduction. A more reasonable improvement measure is to change the default class to pending review, and to ask teachers to re-label.

In order to analyze the stability of the model in this study, the data are divided into three types of random data, and the mean value of Macro-F1 is calculated to see which model is the most stable. This analysis can clarify the specific reasons for the performance gain, and it is not related to the data splitting. The specific situation is shown in the Figure 6.

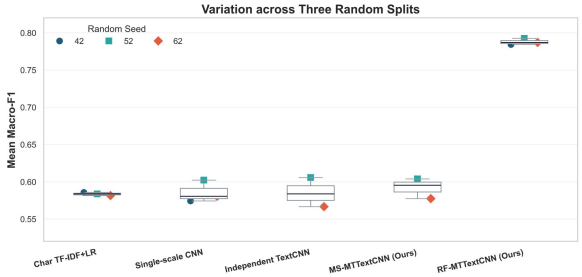

Figure 6. Average Macro-F1 Fluctuation Across Three Random Splits

From Figure 6, we can see that the traditional model will produce a certain degree of fluctuation in the case of different random splitting, and the fluctuation of the independent task is more prominent, with a range of 0.566-0.605.The RF-MTTextCNN model can be controlled within a reasonable range, which is 0.784-0.793, which is enough to show that the model has strong stability and is not affected by training initialization. However, this stability does not mean that the labeling rules will not change. If the rules change, the experts should update their knowledge in time, otherwise they can only reflect the old definition.

4.4 Ablation Study

In order to analyze whether the expert module and shared components play a role in this research, the ablation method is introduced in this research. The Macro-F1 score is taken as the evaluation index, and the expert modules and shared layers are deleted or modified to obtain variant results. The specific results are shown in Figure 7.

The ablation experiment results are shown in Figure 7. If the duration expert is removed, the average macro-f1 will drop from 0.788 to 0.675, with a drop of 0.113. If the code expert is removed, the average macro-f1 will drop from 0.788 to 0.735, with a drop of 0.053. If the formula expert is removed, the average macro-f1 will drop from 0.788 to 0.758, with a drop of 0.030. If all three experts are removed, the model will change into MS-MTTextCNN, which is a purely neural network, and the average macro-f1 will drop from 0.788 to 0.592, with a drop of 0.196. If the shared layer is added to the purely neural network, the average macro-f1 will drop from 0.788 to 0.527, with a drop of 0.261, which shows that the performance problem can not be solved only by adding the shared layer.

The most critical factor is the duration expert, which is related to the imbalance between the two extremes: the pure neural model can accurately predict for a long time, with an accuracy of 0.939, but the Macro-F1 value is only 0.323. There are many shortcomings in the length threshold, and the medium class and the short class can not be restored. The code experts can bring stable gains, but the gain is not large, which is because the regular expression is more accurate than the local character features. It should be noted that there is an important fact that can not be ignored: the RF-MTTextCNN model can not improve the visual_type and scene_type, and can not be semantically innovative, which requires in-depth research.

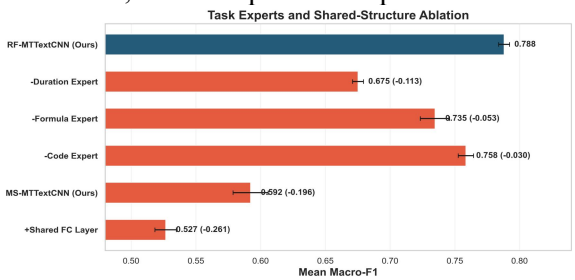

Figure 7. Ablation of Task Experts and Shared Structure

4.5 Qualitative Generation and System Verification

In order to further confirm whether the MP4 file can meet the needs of audio and video synchronization and enhance technical compatibility, this paper selects multiple representative topics for research, focusing on the number of storyboards, video encoding, audio encoding, and the difference between video and audio duration. The specific results are shown in Table 3.

According to the above table, the three cases generated by the above method all have eight storyboards, and the H.264 encoding technology is used to encode the video, and the audio is

encoded, which can meet the playback requirements of MP4.The time difference between audio track and video track is small, the shortest is 0.011 s and the longest is 0.016 s, which shows that the technical stability is relatively strong, and the audio and video are synchronized.

Table 3. Audio-Video Track Verification of Three Generated Cases

Topic	StoryboardShots	VideoTrack	Audio.Track	Video/Audio.Duration/s	DurationDifference/s
GradientDescent	8	H.264	AAC	75.333/75.349	0.016
RandomForest	8	H.264	AAC	60.000/60.011	0.011
ForwardPropagation	8	H.264	AAC	60.000/60.011	0.011

From the perspective of system pipeline, correct prediction can not guarantee the effectiveness of video, although the correct prediction of visual_type can provide too many coordinates, the labels are more and more, there are more and more data points, and the text is not matched with the keyframes of animation. The current implementation of the system can prevent audio and video truncation through back writing and ensure the integrity of the field through Schema validation, but it can not automatically evaluate the explanation rhythm, visual load and factual correctness. Therefore, only qualitative examples can be given to ensure the smooth running of the whole pipeline, but they can not replace the blind review and learning effect experiments carried out by teachers.

5. Conclusion

In this research, a large number of teaching videos are generated based on the educational text, and six task storyboards are constructed to achieve weak supervision, so as to build a generation pipeline.In the experiment, the unified reproducibility of the generated results is ensured, and it is found that the rule-fused structure can achieve an average accuracy of 0.837 and an average macro-F1 of 0.788.The two indicators are 0.061 and 0.196 higher than the latter, and there is no need to increase the training parameters.

The current method is inevitably limited in the following four aspects: First, the weak label is defined by three rule experts, and the score obtained by aggregation can only reflect the consistency of internal engineering, but it can not be regarded as the gold standard for human beings. Second, the macro-F1 values of the two dimensions are not ideal, the semantic boundary has not been resolved, and the default-class deviation has not been eliminated. Third, the data distribution is not balanced, and the test set of teacher annotation is not used in the current research. The fourth is that the LLM fallback script can be repeated, but the learning effect is not compared with the blind review. Next, we should manually label the test set, introduce the concept of uncertain labels, set the learning gate, compare the Chinese pre-training model, and carry out the research on the system level, including fact checking, diversity detection and so on, so as to promote the system from the initial prototype to the production tool that can be used for a long time.

Acknowledgments

This research was supported by the Undergraduate Innovation Training Program (No. 202610463006).

References

[1] Guo Jianwei. Adding a Beautiful Cover to Flash Animation. Computer Knowledge and Technology, 2016(3):96-97.

[2] Wang Bin. Batch Output for Unattended Multi-Video Rendering with Edius. Computer Knowledge and Technology, 2016(6):29-30.

[3] Singer U, Polyak A, Hayes T, et al. Make-A-Video: Text-to-Video Generation without Text-Video Data. arXiv:2209.14792, 2022.

[4] Kondratyuk D, Yu L, Gu X, et al. VideoPoet: A Large Language Model for Zero-Shot Video Generation. Proceedings of the 41st International Conference on Machine Learning, 2024:25105-25124.

[5] Yang Y, Fei Z, Xu D, et al. CogVideoX: Text-to-Video Diffusion Models with an Expert Transformer. arXiv:2408.06072, 2024.

[6] Ho J, Chan W, Saharia C, et al. Video Diffusion Models. arXiv:2204.03458, 2022.

[7] Caruana R. Multitask Learning. Machine Learning, 1997, 28:41-75.

[8] Kim Y. Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014:1746-1751.

[9] Vaswani A, Shazeer N, Parmar N, et al.

Attention Is All You Need. Advances in Neural Information Processing Systems, 2017, 30.

[10] Ratner A, Bach S H, Ehrenberg H, et al. Snorkel: Rapid Training Data Creation with Weak Supervision. Proceedings of the VLDB Endowment, 2017, 11(3):269-282.

[11] Kim J, Kong J, Son J. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. Proceedings of the 38th International Conference on Machine Learning, 2021:5530-5540.

[12] Paszke A, Gross S, Massa F, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems, 2019, 32

