WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

CampusLink: An ML-Based Campus Event Discovery and Recommendation System

Kabilan V[1], Samuel Santhosam S[2], Poojitha V[3], Dr. G. Paavai Anand[4]

[1–3] Undergraduate Students; [4] Faculty Guide

Department of Computer Science and Engineering SRM Institute of Science and Technology, Vadapalani, Chennai, India

[1] RA2411003040070 [2] RA2411003040026 [3] RA2411003040030

[1] Kv7794@srmist.edu.in, [2] ss5732@srmist.edu.in , [3] poojithavv6630@srmist.edu.in

ABSTRACT

Campus event announcements distributed across messaging groups and social-media pages are difficult to search and personalize. This paper presents CampusLink, a campus event discovery and registration system that integrates text classification, semantic retrieval and hybrid recommendation. A React interface communicates with a FastAPI backend. TF-IDF with Logistic Regression assists organizers in assigning event categories; keyword retrieval and a pretrained MiniLM encoder support discovery; reciprocal rank fusion combines their rankings. A recommender integrates student interests, participation similarity, popularity and negative feedback. Evaluation separates authored examples, synthetic events and reviewed campus announcements. Classification achieves 78.6% accuracy and a macro-F1 of 0.793 on 28 reviewed real announcements after training on 72 authored examples. On a 200-event synthetic retrieval benchmark, TF-IDF achieves nDCG@5 of 0.817 on the topical and constrained portion of a 45-query test split. A recommendation experiment with 30 simulated users yields Recall@10 of 0.167 for the hybrid method and 0.100 for popularity-only ranking. Semantic and fusion performance remain unmeasured on the larger and real-event benchmarks. The results establish a functioning applied ML prototype and identify the need for independently labelled data and real-user evaluation.

Keywords Campus events; machine learning; TF-IDF; semantic search; reciprocal rank fusion; hybrid recommendation; event classification.

1. Introduction

Campus clubs and departments publish workshops, contests, seminars and recruitment-related events through several communication channels. Students must interpret posters, locate registration links and check dates and venues separately. A chronological message stream makes it difficult to retrieve an older announcement or discover an event through a description of a learning goal. Organizers also need consistent event metadata and a dependable record of registrations.

CampusLink addresses these tasks through a shared catalogue and ML-assisted discovery. The implementation focuses on CSE clubs and department events at SRMIST Vadapalani. A student can search for an event, inspect its venue and schedule, register, and receive recommendations from interests and participation history. Organizers receive category suggestions while retaining control over the published category.

1.1 Background and Motivation

Event-based social networks connect online activity with offline participation [1]. In a campus setting, event text provides useful information even when interaction histories are sparse. Lexical retrieval offers an interpretable baseline, while sentence embeddings can represent paraphrased requests. Combining these signals is relevant when a student describes an intention such as learning to secure a website rather than naming the exact workshop title.

A central catalogue also creates a common description of each activity. An event title alone may communicate a brand name without explaining the subject, while a long poster may contain venue directions and promotional phrases alongside its actual learning objectives. Separating structured details from descriptive text allows the application to answer two different questions: whether the event is relevant to a student's interests and whether the student can attend it. The first question motivates text-based ranking; the second requires dependable metadata and registration rules.

The motivation for combining methods is practical. Exact technical names, such as a programming language or framework, can be informative lexical features. A natural-language description of an intended skill may use none of the same words as an announcement. Participation history introduces another source of evidence, but a new student cannot supply it immediately. CampusLink therefore brings these signals into one workflow while keeping their purposes distinguishable.

1.2 Problem Statement

The problem is to retrieve and recommend relevant campus events from heterogeneous announcements while supporting organizer categorization and registration. A suitable system must distinguish topical relevance from explicit constraints such as date and price. It should also return an empty result when no event is sufficiently relevant. For personalization, it must remain usable when a new student has little or no participation history.

Three tasks define the ML scope. First, the organizer supplies title and description text and receives a category suggestion. Second, a student supplies a query and receives a ranked list of eligible events or an explicit empty result. Third, a student profile and available interactions produce personalized recommendations. These tasks have different targets and must be evaluated separately: a correct category does not guarantee a useful search ranking, and a relevant search result does not establish that an individual student will register.

1.3 Technical Challenges

Announcements vary in length, vocabulary and completeness. Closely related categories, including programming, web development and design, share terms. New users create sparse interaction matrices, and synthetic examples may overstate model quality. Retrieval introduces a further trade-off: a permissive threshold improves coverage but can return unrelated events. Evaluation must therefore separate the source of labels, the population represented and the benchmark used for each claim.

A further challenge is changing event availability. Workshops expire, registration closes, and organizers revise locations. A semantically similar historical event can be useful in an offline retrieval experiment but unusable in the live discovery screen. Eligibility is therefore a property of the application state, not a learned relevance score. The distinction also affects experiments: an evaluation that ignores availability answers a narrower question about textual matching.

Sparse participation creates uncertainty in peer comparisons. Two users may appear similar after sharing one event, even though their wider interests differ. Popular events can dominate because they supply more interactions. Negative feedback is also ambiguous: a student may dismiss a topic, a date, or an event that they have already attended elsewhere. The prototype makes these signals visible in its scoring design, but their interpretation requires further study with actual participants.

1.4 Contributions

The contribution is an integrated campus application of established ML methods, accompanied by an explicit evaluation of available evidence. The system combines category assistance, three retrieval modes and an interpretable hybrid recommender within event workflows. It exposes recommendation components and classification feature contributions, and applies explicit relevance and eligibility checks. The study reports separate synthetic and real-announcement experiments, including failure cases and missing evaluations, rather than claiming a new learning algorithm.

The research questions are correspondingly limited. How well does an authored training corpus transfer to differently worded campus announcements? Can lexical ranking retrieve relevant catalogue records while rejecting clearly unrelated requests? Does a hybrid recommendation score retrieve more held-out interactions than its individual baselines in the available simulation? These questions organize the experiments in Section 4 and prevent application features from being mistaken for measured educational or administrative outcomes.

2. Related Work

2.1 Lexical Retrieval and Text Classification

Salton and Buckley [2] studied term weighting for document retrieval and established the importance of weighting informative terms. TF-IDF provides a suitable baseline for short event descriptions because its feature contributions can be inspected. CampusLink uses word unigrams and bigrams for retrieval and classification. Logistic Regression, Multinomial Naive Bayes and linear support vector machines are available through scikit-learn [3]; the deployed classifier uses Logistic Regression to return class probabilities and feature contributions.

For CampusLink, a useful property of sparse text features is that the representation can be examined directly. A term that occurs across almost every event contributes less discrimination than one concentrated in a small set of descriptions. Bigrams retain some short phrases that would otherwise be split into unrelated words. This makes the baseline appropriate for diagnosing category overlap and discovering whether a prediction depends on a meaningful technical phrase or on incidental wording. Its main limitation is dependence on observed vocabulary: an unfamiliar paraphrase may share little evidence with an otherwise relevant event.

2.2 Sentence Embeddings for Semantic Retrieval

Reimers and Gurevych [4] introduced Sentence-BERT, which generates sentence representations that can be compared using cosine similarity. This approach supports retrieval without jointly processing every query-document pair through a cross-encoder. CampusLink uses the existing all-MiniLM-L6-v2 encoder, whose model card specifies 384-dimensional embeddings and truncation of long inputs [5]. The project performs local inference with pretrained weights; it does not fine-tune the encoder on campus data.

Dense retrieval changes the representation rather than the event eligibility rules. A pretrained encoder places text in a continuous vector space, making similarity available even when exact words differ. However, general language similarity is not the same as campus-specific relevance. Two activities can discuss the same broad subject while addressing different audiences or levels of experience. For this reason, the semantic path is treated as a separate model component whose threshold and retrieval quality require validation on the intended catalogue.

2.3 Rank Fusion

Cormack, Clarke and Büttcher [6] proposed reciprocal rank fusion to combine retrieval rankings. Rank-based fusion is useful when lexical and embedding scores have different numerical scales. CampusLink adapts this approach by applying relevance thresholds to form a candidate union and then combining lexical and semantic ranks. The present evidence does not establish that this configuration outperforms either constituent method on real campus announcements.

Fusion avoids directly averaging values whose distributions may differ. A high lexical score and a high embedding score do not necessarily have the same interpretation, whereas an item's rank expresses its relative position within each method. Reciprocal rank fusion rewards items appearing near the top of either ranking. Agreement can strengthen a candidate, but fusion cannot create relevant evidence when both inputs are poor. The relevance gate is consequently important: it prevents rank combination from automatically treating every event as an acceptable answer.

2.4 Collaborative and Event Recommendation

GroupLens [7] demonstrated collaborative filtering based on similarities between users. Liu et al. [1] examined online and offline interactions in event-based social networks, including event recommendation. CampusLink uses a smaller neighbourhood model that combines peer participation with event text and stated interests. The current interaction data are simulated, which limits comparison with studies based on observed behaviour. The contribution lies in the application workflow and transparent scoring, not in introducing collaborative filtering.

Content-based recommendation is especially relevant to a small campus deployment because an organizer can publish a new event before any student has interacted with it. The event's description still supports matching against stated interests. Collaborative evidence becomes useful when enough overlapping activity is available, while popularity supplies a simple comparison baseline. These components address different information conditions. Their combination should therefore be judged through cold-start and returning-user experiments, rather than assuming that one fixed mixture is best for every student.

2.5 Research Focus

These methods provide complementary building blocks, but their suitability must be assessed within the target campus catalogue. This study investigates whether a small-data prototype can support understandable classification and retrieval, and whether adding peer signals changes recommendation ranking in simulation. It does not claim that previous systems lack these features. The unresolved empirical questions concern performance on independent announcements, rejection of unrelated queries and recommendation quality for actual students. Graded retrieval measures [8] help distinguish useful ranking from simple keyword overlap.

3. Methodology

3.1 Software Components

The frontend uses React and Vite. FastAPI exposes endpoints for event search, categorization and recommendations, alongside authentication and registration functions. NumPy and scikit-learn implement vector operations and the classical classifier. Sentence Transformers loads the local MiniLM encoder. SQLite supports the packaged demonstration. The semantic model must be cached locally; unavailable semantic inference produces an explicit error rather than silently substituting keyword search.

The software separation also defines where failures should be handled. The browser presents search modes, event details and registration outcomes; the API validates requests and connects the workflow to stored records; the ML components transform text and produce scores. A model-loading failure should be distinguishable from a successful request with no relevant events. These states have different remedies: the former requires an available semantic model, while the latter may require a revised query or a broader event catalogue.

The classical and sentence-embedding paths have different operational requirements. Sparse features and a linear classifier support the baseline without loading the sentence encoder. Semantic inference requires the pretrained model files and an initialized runtime. Keeping the requested mode explicit makes demonstrations repeatable and helps users understand which method generated the result. It also prevents an evaluation from silently attributing lexical results to the semantic model.

3.2 Data Sources and Preprocessing

The project separates three principal datasets, summarized in Table 1. The authored classification corpus contains six balanced categories: AI & Data, Web & App, Cybersecurity, Programming, Design & Product, and Career & Community. Each category has 12 training and three test examples. The reviewed-event catalogue contains 49 records derived from campus announcements, of which 28 are marked for classification evaluation. Categories were assigned by the project author and have not been independently adjudicated.

Table 1. Dataset composition and intended use

Dataset	Records	Evaluation role
Authored classification text	90	72 training; 18 test
Fictional event catalogue	200	Retrieval and topic metadata
Reviewed campus announcements	49	Retrieval; 28 classification test rows
Real-event retrieval queries	16	Author-assigned relevance judgments
Synthetic retrieval queries	76	31 validation; 45 test
Simulated peer users	30	Recommendation leave-one-out test

The real catalogue includes 26 verified poster associations. Ambiguous records are retained in the archive but excluded from the selected classification test. No real student interaction dataset is included. Text features use English stop-word removal, unigrams and bigrams; classification and lexical search use sublinear term-frequency scaling. The vectorizer is fitted on the classifier training partition. Retrieval fits its lexical vocabulary on the event corpus and transforms each query against that vocabulary.

The distinction between training text and searchable catalogue text is essential. Category prediction must operate on information available before the category has been assigned. Adding the correct category label to the classifier input would make the task artificially easy. Search, in contrast, can legitimately use an organizer-approved category and tags because they are already part of the published event. The same field may therefore be valid in one pipeline and invalid in another; data preparation must follow the task definition.

The authored corpus gives the classifier an initial vocabulary, but it does not reproduce all the variation of real announcements. Repeated templates, balanced classes and clear topic words can produce favourable results on a small authored test set. Real notices may emphasize a club name, guest speaker or promotional theme instead of the category. Keeping the real-announcement test separate makes this difference observable. The selected real records still provide only a limited sample, especially for categories with very few examples.

3.3 System Architecture

Figure 1 shows the application layers. The API supplies eligible event records and interaction history to the ML components. Predictions and ranked results return as structured responses to the interface. Search operates on published upcoming events in the application; the offline real-announcement experiment deliberately searches the historical catalogue to evaluate relevance independently of current event eligibility.

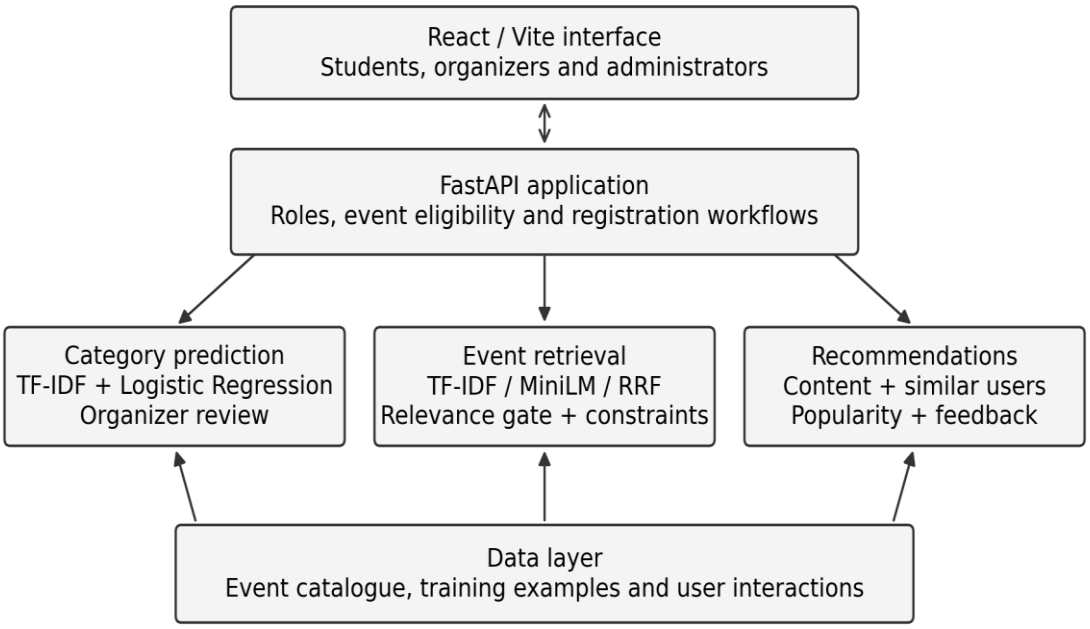


Figure 1. CampusLink architecture and the inputs used by each ML component.

The architecture has two related flows. In the organizer flow, event text is passed to categorization, reviewed, and associated with structured schedule and venue details. In the student flow, a query or profile is evaluated against eligible catalogue records, after which the student opens an event and completes the registration workflow. A ranking is an aid to discovery; it is not permission to bypass capacity, publication or registration checks.

Model inputs should remain traceable across these flows. A category explanation refers to the text features used for classification, a retrieval score refers to the current query, and a recommendation explanation refers to profile or interaction signals. Presenting these as separate outputs avoids confusing one type of evidence with another. This separation is also useful during debugging because an incorrect event detail, a weak text match and a poor recommendation can originate in different components.

3.4 Module 1 Event Categorization

The categorization module concatenates the event title and description, transforms them into TF-IDF features, and applies Logistic Regression with C = 8, a maximum of 1,000 iterations and random seed 42. It predicts one category and returns a probability for every class. The highest-probability class becomes the suggestion. A confidence below 0.45 triggers a review flag. This threshold is heuristic; the probabilities have not been separately calibrated.

The returned explanation lists positive products of TF-IDF feature values and the predicted class coefficients. These quantities describe contributions to the linear class score and are not independent changes in probability. Organizers can revise the proposed category. Additional tags are extracted through predefined keyword rules, so category prediction and tag extraction use different mechanisms.

The review flag gives the organizer a decision point rather than an automatic rejection. A low maximum probability can indicate unfamiliar vocabulary or competing categories, but a high value is not proof of correctness. The organizer can inspect the title, description and influential terms before accepting the label. Rule-based tags provide additional descriptors, although their coverage depends on the maintained keyword rules. The distinction matters when describing the project: learned categorization and deterministic tag extraction should not be reported as a single trained model.

For an illustrative case, consider an announcement about designing a mobile application interface. Terms related to applications may support Web & App, while interface and design language may support Design & Product. This example describes the kind of ambiguity revealed by the recorded errors; it is not an additional measured prediction. A richer training corpus could include examples at these category boundaries, while an eventual multi-label design could represent activities that legitimately belong to more than one area.

3.5 Module 2 Event Retrieval

The searchable representation concatenates title, description, category and tags. Including catalogue metadata is appropriate for retrieval because those fields exist before a search. It is unsuitable as input when evaluating prediction of that same category. The query processor removes selected request words and handles price and relative-date expressions through rules. For example, a request for a free event filters fee metadata before application retrieval.

Lexical search computes cosine similarity between TF-IDF query and event vectors. Semantic search encodes both texts with all-MiniLM-L6-v2 [5]. For normalized embeddings, cosine similarity reduces to their dot product. The code caches the sentence model and event embeddings; it fits the lexical vectorizer during scoring.

$$cos(q, e) = \frac{q \cdot e}{\|q\| \, \|e\|} \quad (1)$$

The configured lexical threshold is 0.02, selected using synthetic validation queries. The semantic threshold is 0.30 and remains an untuned default in the larger saved benchmark. An event enters the hybrid candidate set when it passes either threshold. Each retained event receives a rank in both score lists. With one-based ranks, fusion follows Equation (2), using k = 60 [6].

$$RRF(e) = \frac{1}{60 + rlex(e)} + \frac{1}{60 + rsem(e)} \quad (2)$$

The implementation divides the fused value by 2/61 for display. This rescaling preserves rank order and does not turn the result into a probability. When no event passes the relevance gate, the interface returns a no-relevant-events message. Date and fee constraints remain separate from the learned representation.

The retrieval sequence begins with the query and available event records. Explicit constraints narrow the eligible set, text is prepared for the requested mode, and similarity determines the candidate ordering. The relevance threshold then controls whether sufficiently related evidence exists. Hybrid search considers the candidate union before combining ranks. The final list must preserve event identifiers so that a ranked result opens the correct details and registration page.

For example, a request to learn about natural language processing expresses a subject, while a request for a free workshop this weekend adds price and time constraints. The subject can be scored from text, but the date must be interpreted against a concrete calendar interval and the fee against structured metadata. A cooking-related request against a catalogue containing only computing activities should be able to return no suitable event. These examples explain intended behaviour; the reported benchmarks cover only the particular query sets described in Section 4.

Threshold selection involves a trade-off. A permissive gate can recover loosely worded relevant events while also admitting unrelated material. A restrictive gate can improve apparent precision by hiding useful candidates. Validation should therefore measure both positive-query retrieval and unrelated-query rejection. The current lexical threshold has synthetic validation support, whereas the semantic default has not been established as optimal for the larger benchmark. The numerical values should be interpreted in that context.

3.6 Module 3 Personalized Recommendation

The recommender constructs a user-event matrix with weights of 1 for saved, 2 for interested, 3 for registered and 5 for attended events. Repeated signals retain the maximum weight. Cosine similarity selects up to 20 neighbouring users, and their weighted interactions produce a collaborative score. A content profile combines stated interests with text from previously interacted events. TF-IDF similarity supplies the content score, while popularity is the proportion of users who interacted with an event.

Let CF and CT denote the collaborative and content scores, P the popularity score, and D the maximum content similarity to a dismissed event, with D = 0 when there are no dismissals. With positively similar neighbours, the ranking score is:

$$H(u, e) = 0.45\,CF + 0.45\,CT + 0.10\,P - 0.30\,D \quad (3)$$

When no positively similar neighbours exist, the system uses the following content and popularity fallback:

$$Hcold(u, e) = 0.85\,CT + 0.15\,P - 0.30\,D \quad (4)$$

The weights are fixed design choices, not learned parameters. The response exposes score components and a rule-selected explanation based on interests, similar participation or content similarity. These explanations describe the inputs used; they are not causal explanations. In the recommendation experiment, previously seen events are removed after ranking.

The interaction weights express a design assumption about signal strength: attendance is treated as stronger evidence than saving an event. Taking the maximum of repeated signals avoids counting the same progression as several independent confirmations. However, these weights do not measure preference directly. Attendance may depend on timetable constraints or a departmental requirement, while saving may reflect genuine interest without later availability. The prototype exposes the scoring components so that these assumptions can be examined during future evaluation.

Cold-start handling applies when positively similar neighbours are absent. The fallback increases the role of the content profile and retains a smaller popularity contribution. This keeps the ranking defined for a student who has supplied interests but has little interaction history. If the supplied profile itself is uninformative, personalization is correspondingly weak; a deterministic score cannot recover preferences that were never observed. Evaluating this case separately would show whether onboarding interests provide enough information for useful first recommendations.

The dismissal term reduces scores for events resembling dismissed content. It introduces negative feedback without claiming that a dismissal is a permanent statement about a student's interests. Because the penalty is a fixed coefficient, its effect can vary with the other score components. Future sensitivity analysis should compare rankings with and without this term and inspect whether it suppresses relevant opportunities too aggressively. Such an ablation is a proposed experiment, not a result of the current study.

3.7 Registration and Event Management

Discovery connects to event details containing schedule and venue information. The application includes registration checks, capacity management, waiting lists and organizer-controlled attendance and on-duty document workflows. These functions provide the context in which recommendations are consumed. The paper evaluates ML outputs rather than attributing improvements in attendance, administrative effort or student engagement to these application features.

Registration and recommendation should remain conceptually separate. An event may rank highly and still have no remaining places. Conversely, an event with weak personalization may remain important to a student who searches for it directly. Event details must therefore provide the decision information independently of rank: timing, location, organizer information and the applicable registration state. The catalogue is useful only when the student can act on the result that the discovery system presents.

The organizer's workflow also provides an opportunity to improve data quality. Reviewing a suggested category, completing a missing venue and correcting a misleading description improves the input available to subsequent search. However, organizer edits are not automatically ground-truth labels for a research benchmark. An experimental dataset needs a documented labelling procedure and a stable version so that later corrections do not silently alter the test set.

4. Experimental Setup and Results

4.1 Experimental Environment

The project provides Python evaluation scripts and saved JSON reports for classical classification, retrieval and simulated recommendation. The recorded validation configuration identifies Python 3.12 and scikit-learn 1.8.0. Complete processor and memory details are absent from the saved experiment records, so this paper does not present a hardware-controlled latency comparison. All results below identify their dataset and evaluation scope. The 28-record real-announcement classifier result was independently reproduced during manuscript preparation.

Reproducibility requires preserving the data split, preprocessing configuration, model parameters and metric definitions together. The reported classifier configuration includes its regularization setting, iteration limit and seed. Retrieval also depends on the text fields included in each record and the chosen threshold. Recommendation depends on interaction weights, neighbour selection and candidate exclusion. Recording only a final score would leave several plausible explanations for a changed result in a subsequent run.

The environment description does not support conclusions about production throughput. Model initialization, embedding computation, database access and network transfer can each contribute to response time. A fair future latency experiment should identify what is timed, the number and length of event descriptions, and whether model caches are warm. It should report repeated measurements under a documented workload rather than a single value.

4.2 Testing Methodology

Classification uses accuracy, macro-F1 and a confusion matrix. The deployed classifier trains on the 72 authored examples and tests separately on the 18 authored examples and 28 selected real announcements. No real test labels are used for training in this experiment. A separate synthetic topic-split experiment is excluded because its features contained the target category; reporting it as generalization evidence would introduce target leakage [9].

Synthetic retrieval uses 76 author-written queries derived from event topics: keyword requests, natural-language requests, price constraints and unrelated queries. Topics determine the 31-query validation and 45-query test partitions. Relevance grades are generated from topic and category metadata, rather than independently reviewed. Topical metrics average over 40 related or constrained test queries. The remaining five test queries measure unrelated-query false returns.

Retrieval reports Precision@5, Recall@5, reciprocal rank and nDCG@5 [8]. The synthetic script treats exact-topic grade 2 as relevant for precision and recall, while nDCG uses linear grades. The real-event script treats any positive grade as relevant and uses exponential gain for nDCG. Its reciprocal rank is truncated at five; the synthetic script considers the complete returned ranking. Consequently, scores across these corpora are descriptive results, not a direct comparison of model superiority.

Each metric answers a different question. Accuracy measures the fraction of correct category assignments, while macro-F1 gives equal weight to the category-level F1 scores. Precision at five describes how many of the first five retrieval positions are relevant. Recall at five measures how much of the known relevant set appears in those positions. Reciprocal rank emphasizes the first relevant answer; nDCG additionally distinguishes the order and graded usefulness of several answers. A method can improve one measure and worsen another.

For the recommendation experiment, one withheld interaction per simulated user makes Recall@10 interpretable as the fraction of users whose withheld event is recovered in the top ten. nDCG@10 also rewards placing that event nearer the top. Both measures depend on how candidate events are chosen and whether previously observed events are removed. Reporting these details is necessary to distinguish a useful ranking comparison from an evaluation made easier by duplicate or already-known interactions.

4.3 Classification Results

Table 2. Classification results with training restricted to 72 authored examples

Test data / classifier	Test n	Accuracy	Macro-F1
Authored / majority baseline	18	0.167	0.048
Authored / Naive Bayes	18	1.000	1.000
Authored / Logistic Regression	18	1.000	1.000
Authored / linear SVM	18	1.000	1.000
Real / Logistic Regression	28	0.786	0.793

All three text classifiers correctly classify the 18 authored test examples. This is a small and relatively easy vocabulary-based check, not evidence of universal accuracy. On reviewed real announcements, Logistic Regression correctly classifies 22 of 28 records. Figure 2 shows the six errors: two Web & App events become Design & Product, two Programming events become Cybersecurity, one Programming event becomes AI & Data, and one AI & Data event becomes Web & App.

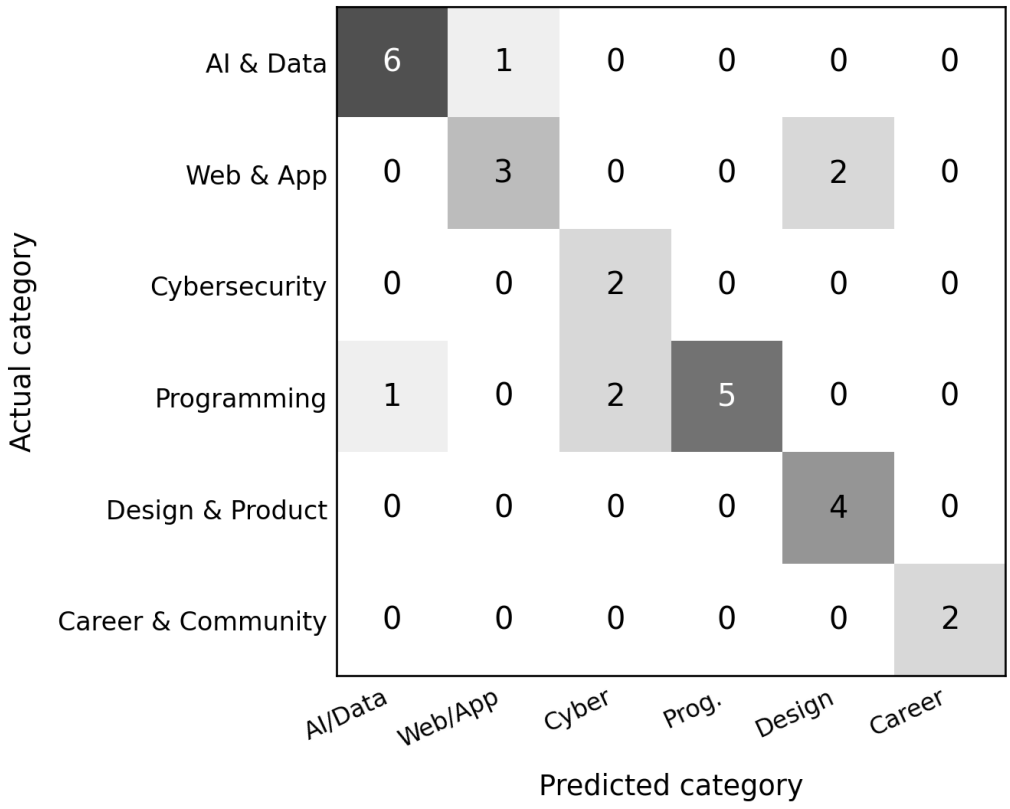


Figure 2. Real-announcement confusion matrix for Logistic Regression (n = 28).

The real test therefore exposes ambiguity absent from the authored split. Mobile-app sessions with design-oriented language and programming sessions with security vocabulary illustrate the overlap. Since each real category has only two to eight evaluated records, class-level conclusions remain tentative.

The difference between the authored and real tests is more informative than the perfect authored score alone. The authored examples demonstrate that the selected features can distinguish the initial templates. The real test introduces vocabulary and framing not controlled by those templates, leading to a lower score. This is evidence of limited transfer within the observed sample, not evidence that the model has failed for every real event. Error review can guide additional data collection without turning those same test examples into an unacknowledged training set.

The six observed errors are concentrated in related subject areas. This suggests that category definitions and boundary examples deserve attention alongside the classifier choice. A future annotation guide should explain how to label mixed-topic activities and allow reviewers to mark genuinely ambiguous records. After any retraining based on the current errors, a new untouched test set would be needed to estimate improvement fairly.

4.4 Retrieval Results

Table 3. TF-IDF retrieval results on separate evaluation corpora

Evaluation	P@5	R@5	nDCG@5	Reciprocal rank
Synthetic: 200 events; 45 test queries	0.740	0.651	0.817	MRR 0.810
Real: 49 events; 16 queries	0.300	0.771	0.853	MRR@5 0.938

On the synthetic test split, none of five unrelated queries returns an event. This is an observation on five cases, not a guarantee of rejecting all irrelevant requests. Validation has an unrelated false-return rate of 0.200. On the real catalogue, the higher reciprocal-rank value indicates that a relevant item often appears early, while Precision@5 is constrained by the small number of relevant events per query. The author-written relevance judgments remain a source of potential bias.

An older archived experiment on 24 demonstration documents and 16 queries records nDCG@5 of 0.5802 for TF-IDF and 0.9530 for MiniLM. That result belongs to an earlier demonstration corpus and was not independently rerun here. It is retained as historical evidence of execution, not as a result on the current catalogue. The larger benchmark records semantic and hybrid search as not run; the real benchmark also lacks semantic results and does not implement a hybrid comparison. No current real-data advantage is claimed for fusion.

The real retrieval results show why a single metric can be misleading. A query with only one relevant event cannot obtain high Precision@5 if five positions are used as the denominator, even when that one event appears first. A high reciprocal rank can therefore coexist with lower precision. Conversely, returning many broadly related events can raise recall without placing the most useful answer first. Inspection of query-level outcomes would complement aggregate metrics and reveal which kinds of requests remain difficult.

The historical MiniLM experiment should not be combined numerically with the larger evaluation. Corpus size, query wording, relevance judgments and implementation versions can all influence retrieval metrics. A valid head-to-head comparison would run the lexical, semantic and fusion methods against the same fixed events and judgments. Until that comparison is available, the semantic component is an implemented capability with incomplete current evaluation, rather than a demonstrated winner on real announcements.

4.5 Recommendation Results

The recommendation experiment holds out one confirmed interaction per simulated peer, builds profiles and neighbours from the remaining interactions, and excludes previously seen events from candidates. This is a random leave-one-out evaluation, not a chronological deployment test. Table 4 compares the resulting rankings at ten positions.

Table 4. Recommendation results on 30 simulated users

Method	Recall@10	nDCG@10
Popularity only	0.100	0.066
Content only	0.100	0.083
Hybrid recommendation	0.167	0.081

The hybrid method recovers the held-out event for approximately five users, compared with three for each baseline. However, its nDCG@10 is slightly below the content-only result. The experiment therefore suggests a change in retrieval coverage, not uniform improvement in ranking quality. With only 30 synthetic users and no uncertainty analysis, these differences do not establish a benefit for actual students.

The recommendation comparison is based on very few recovered interactions. Five recoveries rather than three is a useful observation for diagnosing this simulation, but changing only one user's outcome would noticeably alter the reported rate. The lower hybrid nDCG relative to content-only ranking

further indicates that finding an item within ten positions and placing it near the top are separate objectives. The results should motivate a larger evaluation, not a claim of a reliable engagement improvement.

The simulation also cannot test whether explanations improve trust or whether students prefer the recommended events. Those are user-facing outcomes that require observed responses under a defined study protocol. A prospective evaluation could compare recommendation acceptance, perceived relevance and reasons for rejection while retaining the offline metrics as diagnostic measures. These measures would complement rather than replace one another.

5. Discussion

5.1 Key Findings

The clearest finding is the contrast between authored and real-announcement classification. Perfect scores on 18 authored examples coexist with six errors on 28 real examples. A realistic assessment must include wording that differs from training templates. The inspected real-event result supports organizer assistance with human review, rather than unattended categorization. Returning feature contributions can help organizers understand why overlapping vocabulary caused a suggestion.

TF-IDF provides a measurable discovery baseline and supports explicit rejection of unrelated queries. MiniLM and reciprocal rank fusion offer additional implemented retrieval paths, but the current evidence is insufficient to establish their advantage on the reviewed campus catalogue. The recommender remains useful as a transparent prototype: its components and fallback behaviour are inspectable, while the simulated experiment exposes the trade-off between finding more held-out items and placing them earlier.

The application-level implication is that reliable event information and conservative ML interpretation are both necessary. A good ranking cannot compensate for an incorrect venue or an expired registration link, and accurate metadata alone cannot resolve every natural-language request. CampusLink provides a place to combine these concerns while preserving a distinction between deterministic workflow checks and model-based suggestions. Its present value is demonstrated integration and inspectable behaviour within a limited evaluation setting.

5.2 Limitations and Validity

The datasets are small, predominantly authored or synthetic, and restricted to one campus domain. Real-event labels and query judgments come from one author, and selection of the 28 classification records excludes ambiguous rows. These choices limit generalization. The six-category taxonomy does not evaluate coverage of all departments, sports or cultural events. Single split results lack confidence intervals, and changing the simulation can change recommendation outcomes.

Implementation and evaluation also differ in scope. The application searches eligible upcoming records, whereas the real retrieval experiment searches historical announcements. The synthetic benchmark filters a pre-scored corpus, while the application applies constraints before scoring. Relative-date filtering is implemented but is not covered by the reported retrieval metrics. Synthetic ranking timing excludes expensive scoring steps, so it cannot be interpreted as end-to-end response time. No measured user study, attendance improvement or production deployment outcome is available.

The source material includes campus announcements and poster references that may contain personal contact details. This manuscript reports aggregate counts and anonymized error categories and reproduces no private chat messages, student participation records or original posters. Public release of the underlying material requires a separate review of source permissions and removal of personal identifiers. Student feedback collection should use informed consent and anonymized identifiers.

Potential validity threats arise at several stages. Author-created queries may resemble the language used in the event corpus; a manually selected subset can underrepresent difficult examples; and simulated peers may express cleaner interests than real students. None of these issues makes the prototype unusable, but each restricts how far the reported numbers can be generalized. Explicitly separating the datasets and preserving missing-result statements makes the evidence easier to interpret.

The campus context also creates a temporal limitation. An event catalogue changes as activities are announced and completed, while a benchmark is fixed. Model performance measured on one snapshot may change with a new department, club or academic term. Periodic evaluation on newly collected announcements would be more informative than repeatedly reporting the original test score. Any such monitoring should keep evaluation records separate from the data used for adaptation.

5.3 Future Work

The next evaluation should use a larger deduplicated real-event corpus, independent category and relevance reviewers, and a frozen test partition. Classification inputs must match the organizer interface and exclude target metadata. TF-IDF, semantic and fusion retrieval should run on the same corpus with identical relevance definitions and validation-only threshold selection. An ablation study should isolate relevance gating, fusion and metadata contributions. Full query latency should distinguish cold loading from warmed inference.

Recommendation research requires consented student interactions with time stamps, chronological splits and candidate sets restricted to events available at recommendation time. Further comparisons should examine cold-start users, catalogue coverage and popularity bias. Calibration of category

probabilities and validation of recommendation weights may improve reliability. Extension to other departments requires a broader taxonomy and an evaluation dataset that represents those activities.

A practical evaluation sequence would first stabilize the dataset and annotation guide, then compare the three retrieval modes under identical conditions, and finally evaluate recommendation using time-stamped interactions. Thresholds and mixture weights should be selected on validation data before the held-out test is examined. Error analysis should report both successful and unsuccessful query types. This sequence would strengthen the evidence while keeping the current prototype's transparent baseline available for comparison.

Additional work could investigate multi-label categorization, broader department coverage and support for multilingual event descriptions. These are extensions of scope rather than capabilities established by the present experiments. Their implementation would require representative training examples, suitable relevance judgments and checks that explanations remain understandable to organizers and students. Improvements should be accepted on the basis of these evaluations rather than model complexity alone.

6. Conclusion

CampusLink integrates event categorization, lexical and semantic retrieval, hybrid rank fusion and personalized recommendation into a campus event workflow. The implementation combines established models with relevance filtering and inspectable score components. A reproduced real-announcement experiment achieves 78.6% classification accuracy and macro-F1 of 0.793 on 28 records. TF-IDF retrieval provides a baseline on separate synthetic and real corpora, while the recommendation experiment shows higher Recall@10 for the hybrid method without a corresponding improvement over content-only nDCG@10. These results support the feasibility of the prototype within the evaluated setting. Stronger claims about semantic retrieval, personalization and student outcomes require consistent real-data experiments and a prospective user study.

7. References

[1] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han, “Event-based social networks: Linking the online and offline social worlds,” Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1032–1040, 2012. doi: 10.1145/2339530.2339693.

[2] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988. doi: 10.1016/0306-4573(88)90021-0.

[3] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. https://jmlr.org/papers/v12/pedregosa11a.html

[4] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” Proceedings of EMNLP-IJCNLP, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410.

[5] Sentence Transformers, “all-MiniLM-L6-v2,” model card. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (accessed October 30, 2026).

[6] G. V. Cormack, C. L. A. Clarke, and S. Büttcher, “Reciprocal rank fusion outperforms condorcet and individual rank learning methods,” Proceedings of the 32nd International ACM SIGIR Conference, pp. 758–759, 2009. doi: 10.1145/1571941.1572114.

[7] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp. 175–186, 1994. doi: 10.1145/192844.192905.

[8] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” ACM Transactions on Information Systems, vol. 20, no. 4, pp. 422–446, 2002. doi: 10.1145/582415.582418.

[9] Scikit-learn developers, “Common pitfalls and recommended practices,” scikit-learn documentation. https://scikit-learn.org/stable/common_pitfalls.html (accessed October 30, 2026).