Fast Hybrid Search for Red-Zone Play Recommendations

Football Track
Paper 170

1. Introduction

Play calling in the National Football League (NFL) is a recurrent, time-constrained decision problem. After each play, the offense typically has 40 seconds (or 25 seconds in specific administrative situations) to snap the ball before incurring a delay of game penalty. Within this interval, the offensive staff must identify personnel, recognize the defensive structure, select a concept and communicate it to the quarterback in a form that can be executed and optionally adjusted at the line of scrimmage. Thus, in practice, only a fraction of the play clock is available for choosing the play itself. The final decision itself must account for down, distance, field position, score, game context and recent tendencies, and it must be made under substantial uncertainty about the defense's next call.

Coaches already treat this problem as one of recall over a large but structured space of past experience. During film study, coaches examine large numbers of plays and group them by situation, concept and coverage for the purpose of distilling them down into game specific call sheets. These sheets present families of plays keyed to down, distance and field position, often with separate sections for areas such as the red zone, third down and two-minute offense. On game day, the coordinator uses these sheets together with live observations to select a small number of candidate calls that fit the current situation, then chooses one based on preference, game plan and feel. This workflow implicitly relies on the ability to retrieve relevant precedents from a much larger universe of plays than can be printed on a single sheet or remembered in detail.

While current playcalling methods rely on archaic call sheets, the broader ecosystem around the sport now provides the raw data needed to mechanize this kind of retrieval. Detailed play-by-play logs capture for every snap the teams involved, down, distance, field position, play description and outcome, and are available for entire seasons in machine readable form. In parallel, player tracking and event data support increasingly sophisticated predictive models of football, including completion probability, separation and route concepts. In most current applications, however, these data feed offline analysis, scouting reports and broadcast graphics instead of real time decision support systems that operate within the play clock.

In this paper, we formulate in-game offensive play selection as an information retrieval problem over a corpus of historical plays. Rather than attempting to learn a full play-calling policy or to predict a single optimal action in the abstract, we ask whether an automated system can approximate the retrieval component of coaching (i.e given a description of the current situation, can the system rapidly surface a small set of analogous plays that a coach might have recalled from film?) and express play recommendations in a compact "play family" description that fits naturally into existing workflows. We refer to these retrieved precedents as inspirations, and to the synthesized description as a play family, for example "SHOTGUN - PASS SHORT MIDDLE to SLOT RECEIVER."

As a means of instantiating this idea, we use red zone plays as our beachhead, where the field is short, the playbook is tightly constrained and small differences in concept selection can have outsized impact on scoring. Using play-by-play logs for the 2020 NFL regular season, we construct a corpus containing all offensive plays that begin inside the opponent's 20 yard line. This yields 7,381 red zone plays across all teams. Each play is mapped to a coaching shaped textual representation that includes the offensive and defensive teams, down, distance, yard line, quarter, a bounded history of recent plays, formation when available, play type, direction and the original natural language description. The result is a collection of compact "play cards" that resemble the summary text coaches already use in practice but that can be consumed by standard retrieval models.

Our retrieval architecture follows a dual retrieval pipeline. First, we formulate offensive play selection as real time hybrid retrieval over a league-wide red zone corpus constructed from play-by-play data and show that a standard BM25 plus E5 configuration can retrieve analyst designated inspirations at useful rates within tens of milliseconds. Second, we couple this retrieval layer with a constrained LLM agent that yields play family recommendations and concrete historical analogues in ∼1 second. This fits naturally within the time budget imposed by the NFL play clock. We relate our design choices to established practice in information retrieval and retrieval augmented generation, positioning the system as a plausible foundation for decision support tools that function as an always on memory for coaching staff rather than as purely offline analytics.

To evaluate our system, we measure Hits@K and mean reciprocal rank with respect to a single designated inspiration play per query, and we record latency statistics for the hybrid search alone. In our experiments the hybrid retriever recovers the labeled inspiration within the top ten candidates in 41% of cases on the 7,381 play corpus, with a mean reciprocal rank of 0.3950 and median retrieval latency of 38.47 milliseconds. The end-to-end system produces a valid play family for every evaluated situation, covers the target play in the candidate set in 38% of cases and attains a 36% match rate on play type, with median end to end latency of 865.91 milliseconds and retrieval accounting for approximately 8% of that time.

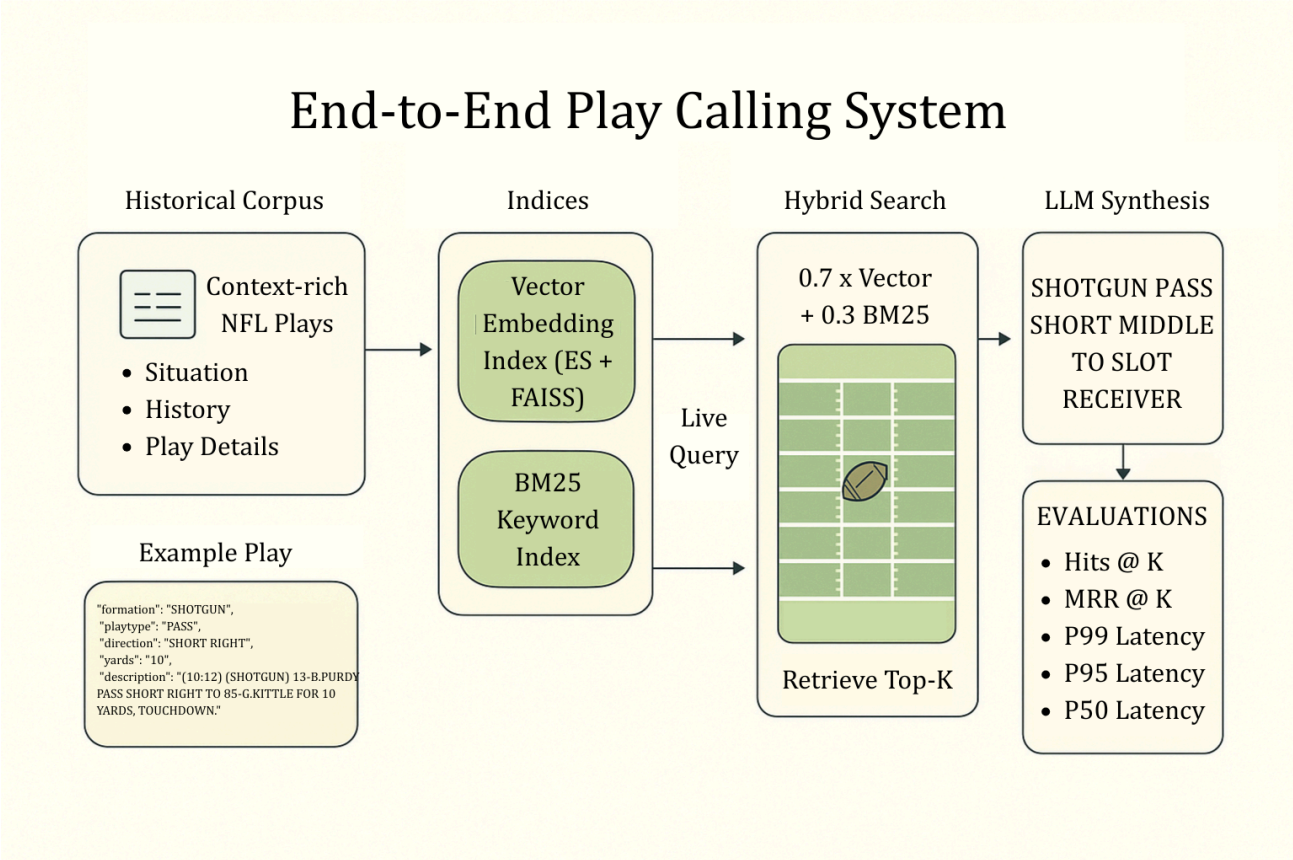

Figure 1. **End-to-End Play Calling System.** Historical play data are encoded into two indices: (1) a vector-embedding index (E5-base-v2 over FAISS) and (2) a BM25 keyword index. A live in-game state (football-field panel) is issued as a query; a hybrid scorer (0.7× vector + 0.3× BM25) retrieves top-K historical analogues. A lightweight LLM then synthesizes a coach-ready call, and outputs are evaluated with Hits@K, MRR, and latency percentiles (p50/p95/p99).

2. Background

2.1 Information Retrieval and Hybrid Lexical-Semantic Search

Most text retrieval systems are built around a standard pipeline: documents are preprocessed into an inverted index and at query time the system computes a relevance score between the query and each candidate document; the top ranked items are returned, often followed by a re ranking step that uses more expensive models on a small candidate set [1]. The BM25 family of probabilistic ranking functions has been the dominant lexical baseline in this pipeline for decades, due to its robustness, simplicity, and strong empirical performance on TREC style benchmarks [2]. BM25 scores documents based on term frequency, inverse document frequency, and document length normalization, and consistently provides a high recall, low latency first stage for ad hoc retrieval [1].

Neural representation learning has introduced dense retrievers that map queries and documents into a shared vector space and retrieve relevant documents by performing a nearest neighbour

search. Dense Passage Retrieval (DPR) and its successors demonstrated strong performance for in domain open domain question answering [3]. The E5 family of text embedding models went further, showing that a single general purpose embedding model trained with weakly supervised contrastive learning can outperform BM25 alone on BEIR in a zero shot setting, while remaining efficient enough for large scale retrieval [4]. However, even with strong dense retrievers, empirical studies on BEIR and related benchmarks show that dense, sparse, and late interaction models have complementary failure modes. No single architecture dominates across all tasks and domains.

These observations have motivated hybrid retrieval architectures that combine a lexical component, typically BM25 or a learned sparse variant, with a dense component based on neural embeddings. Early hybrid work showed that interpolating BM25 scores with neural similarity scores at the retrieval stage improves recall compared with either signal alone, while maintaining acceptable latency by running the two components in parallel [5]. Later work refined the combination step, studying interpolation functions, complementarity objectives, and re ranking strategies, and confirmed that hybrid retrieval can significantly improve nDCG and recall on BEIR and other public benchmarks [6].

Standard industry practice mirrors this research trend. Search platforms such as Vespa [8], Elasticsearch [7], and OpenSearch [9] expose first class support for hybrid ranking profiles that combine BM25 and vector similarity, and vendor documentation explicitly recommends hybrid retrieval as a robust default for question answering and RAG, with BM25 providing strong lexical grounding and dense embeddings providing semantic generalization.

2.2 Offensive Playcalling Practice in the NFL

Offensive play selection in the National Football League is organised around call sheets that encode the game plan for a given opponent. During the week, coordinators and position coaches conduct film study, identify defensive tendencies, and curate families of plays that attack specific structures, coverages, fronts, and personnel groupings. These plays are then organised on the call sheet into sections keyed by down and distance, field zone, hash location, personnel package, and special situations such as red zone, third and long, two minute offence, and backed up scenarios [10]. For example, reporting on Kyle Shanahan's play sheet describes it as a grid of boxed sections that organise plays by situation, including scripted openers, third-down ranges, short-yardage, two-minute and "four-minute" offence, and a dedicated red-zone section in which plays are broken down into five-yard field-position increments [11].

The absolute number of plays encoded in these sheets is large. Bruce Arians has described call sheets that contain roughly 150 plays for a given game, split into clusters such as 25 runs, 35 passes, 15 to 20 third down calls, and 15 to 20 red zone plays, among others [12]. In Arizona, Carson Palmer has noted that his quarterback wristband sometimes listed 171 plays for a single game [13]. Other reporting on play caller practice indicates that call sheets commonly span several laminated pages, with each box or section containing multiple candidate plays that share a common concept but differ in formation, motion, or route detail [21].

On game day, offensive coordinators operate under strict temporal and informational constraints. The play clock is typically 40 seconds from the end of the previous play, but operational overhead, communication with the quarterback, and potential personnel changes reduce the effective decision window to somewhere between 15 and 25 seconds for routine situations and even less in hurry up or two minute contexts. Within that window, the play caller must integrate scoreboard context,

recent play outcomes, defensive substitutions, and live tendencies, then locate an appropriate candidate on the call sheet and communicate it to the huddle or to the quarterback via headset [21].

Qualitative accounts from coaches and quarterbacks highlight several practical limitations of this workflow. First, even with well structured sheets, scanning multiple sections under time pressure can cause coordinators to fall back on a small set of familiar calls rather than systematically evaluating the full space of options keyed by situation [21]. Second, call sheets are static snapshots of the game plan. Adjustments during the game, such as discovering that a particular concept is working against an unexpected coverage variation, are often tracked informally on notepads or through verbal communication rather than being reflected in a dynamically updated index of plays. Third, play sheets primarily encode the offence's perspective on each concept and do not explicitly attach a rich history of similar plays across the league, such as how other teams have attacked a given coverage structure in the same part of the field over multiple seasons.

These constraints mean that coordinators rely on a combination of pre game scripting, shared institutional knowledge, and their own recollection of prior games to select plays that are appropriate for the current situation. The process is highly skilled and heavily studied, but it is not supported by a corpus scale retrieval system that can surface similar plays, from any team, in real time given a structured description of game state. As a result, many potentially relevant precedents remain invisible at call time, especially when they involve uncommon route combinations, motions, or formation adjustments that are difficult to recall accurately under pressure.

2.3 Real Time Analytics Infrastructure in Professional Sports

Over the last decade, professional sports leagues have invested heavily in real time sensing and analytics infrastructure that operates within the temporal constraints of live play. In the NFL, the Next Gen Stats system places RFID tags in player shoulder pads and the ball, and collects position data at approximately ten samples per second from antennas installed around each stadium. This data is streamed to cloud infrastructure hosted on Amazon Web Services, where it is fused with play by play logs to generate derived metrics such as separation, top speed, and completion probability during games [14].

Other sports exhibit similar patterns. Optical tracking systems used in basketball and soccer, including providers such as Second Spectrum and related computer vision systems, extract player and ball trajectories from high frame rate video feeds and deliver positional data and derived statistics in real time for coaching and broadcast applications. Independent validation of electronic performance and tracking systems for soccer reports average system latency on the order of 1 second between the physical event and the availability of tracking data for analysis [15].

Within American football specifically, teams already make use of tablet based video systems that deliver multi angle replays on the sideline within tens of seconds after each play, and analysts in the coaching booth receive live cut ups and basic statistical summaries during the game [16]. However, most of these tools focus on visualisation and descriptive statistics, rather than on corpus scale retrieval of similar historical situations. They provide fast access to what just happened, but not to what has happened across thousands of prior games in similar circumstances.

For a retrieval system to be viable as an in-game decision support tool in this ecosystem, its latency must be small relative to the play clock and comparable to the latency of existing analytics products. Sub-second retrieval for the candidate set and sub 1-2 second end-to-end response times are comfortable thresholds in this context: they are fast enough to be used interactively during the 15-20 seconds when coordinators typically select plays, and they align with the latencies already accepted for tracking based metrics and video replay delivery in professional sport.

3. Methodology

3.1 Dataset

The dataset consists of offensive snaps from the 2020 National Football League (NFL) regular season where the line of scrimmage is inside the opponent's 20 yard line. Play-by-play logs were obtained from BigDataBall [17], a commercial provider of structured NFL statistics that aggregates league data into machine readable tables suitable for analysis.

From the full season logs, all offensive plays with a starting yard line between the opponent's 20 and goal line were extracted. This filtering produced a corpus of 7,381 red zone plays across all teams. For each play, the raw data includes identifiers for game and play, offensive and defensive team abbreviations, down, distance to first down, yard line, quarter, play type, nominal direction, yards gained, and a free text description drawn from the official play by play source.

To support a retrieval formulation, each play is augmented with a structured context and a history window. The context comprises:

- Situation: down, yard to go, yard line, quarter
- Teams: offense and defense identifiers
- History: up to the last five offensive plays by the same team in the same game prior to the current snap, each represented by play type, direction, and yards gained

The history window is truncated if fewer than five prior offensive plays are available in the game. For example, a red zone pass on the opening drive of a game may have only one or two preceding plays.

The dataset is stored in a JSON based format at the play level. Each entry contains a unique play id, a nested play object with play level attributes, and a context object with the situation and history. An example play object from the corpus is provided in Appendix A.1.

To make the corpus compatible with standard text retrieval tools, each play is converted to a coaching shaped textual representation. The transformation function concatenates team labels, situation, recent history, formation, play type, direction, yards gained, and the original natural language description into a single string. For Example:

'MIA offense vs BUF defense. Situation: 2 and 5 at 5 yard line, quarter 4. Recent plays: PASS DEEP RIGHT 0yd, PASS SHORT LEFT 0yd, PASS SHORT MIDDLE 7yd, PASS DEEP RIGHT 0yd, PASS SHORT LEFT 0yd. Play: PASS SHORT RIGHT for 5 yards. Description: (12:22) (Shotgun) 1-T.Tagovailoa pass short right to 10-T.Hill for 5 yards, TOUCHDOWN.'

This representation is stored alongside the structured fields as text. It mirrors the kind of notes coaches write on cut ups while remaining amenable to both lexical and embedding based retrieval.

Corpus statistics are computed to characterise the data. The distribution of playtypes is dominated by PASS and RUSH, with smaller counts of SCRAMBLE, SACK, and rare events such as two point attempts with formations being primarily SHOTGUN and UNDER CENTER.

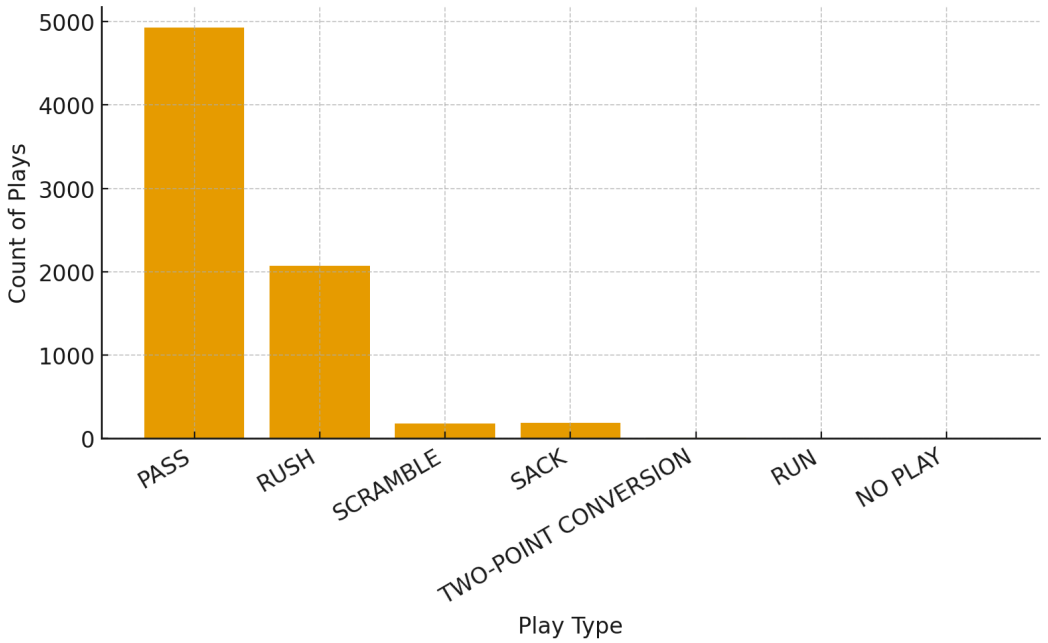

Figure 2. Distribution of Play Types in Red Zone Corpus.

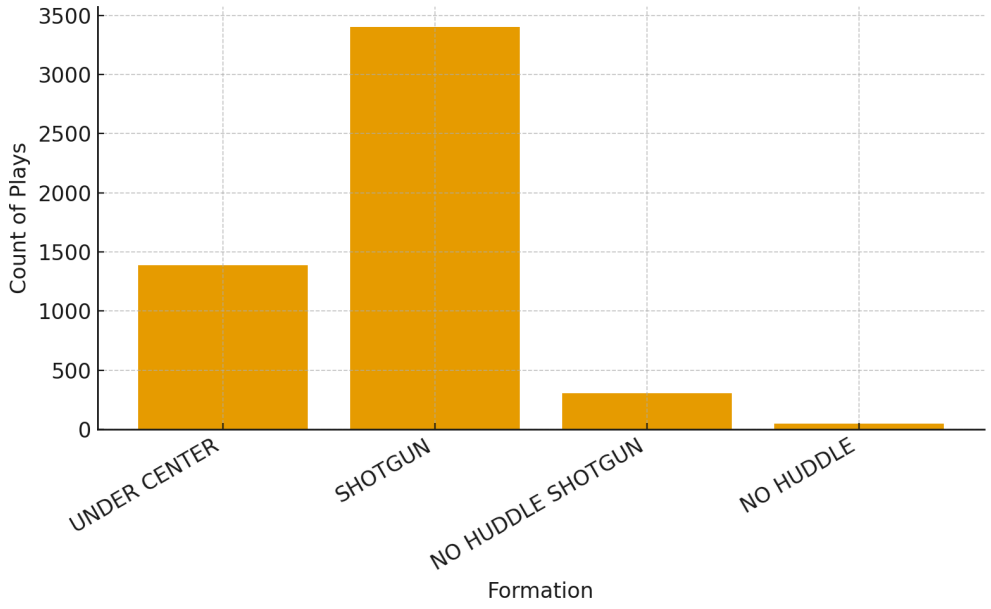

Figure 3. Distribution of Offensive Formations in Red Zone Corpus.

3.2 Retrieval Pipeline

The retrieval system maps each coaching shaped play representation to a dense vector and indexes all vectors with an approximate nearest neighbour structure. At query time, the system encodes the input situation, retrieves a candidate set by vector similarity, combines those scores with lexical BM25 scores, and returns the top ranked plays.

3.2.1 Dense Encoding with E5-base-v2

Dense representations are constructed using the E5-base-v2 sentence embedding model. The E5 family of text embeddings is trained with weakly supervised contrastive pre-training on a large corpus of paired texts, and is designed to provide a single vector embedding suitable for retrieval, clustering, and classification tasks. In particular, the base model is a 12 layer transformer encoder that produces a 768 dimensional embedding for each input string. Inputs are tokenised, passed through the encoder, and pooled into a single vector. The resulting 768 dimensional vectors are converted to NumPy format and L2 normalised so that cosine similarity reduces to inner product. Normalisation ensures that similarity scores lie in a predictable range and works well with inner product based search in FAISS.

To support fast nearest neighbour search over the 7,381 document embeddings, the system uses the FAISS library, a toolkit for efficient vector similarity search and clustering developed at Meta. FAISS provides a range of index types; this work uses the Hierarchical Navigable Small World (HNSW) graph based index, which offers high recall at low query latency for moderate sized collections.

The 768 dimensional document embeddings are added to an *IndexHNSWFlat* index. The HNSW connectivity parameter is set to *M=32*, which controls the number of neighbours per node in the graph, and the construction parameter is set to *efConstruction=200* which improves recall at build

time. The search parameter is set to *efSearch=64*, which trades off recall against query latency during search. These values are typical for high accuracy HNSW configurations on collections of this size. During index construction, embeddings are stored as 32 bit floating point arrays.

At query time, the pipeline encodes the input situation into a 768 dimensional query embedding, normalises it, and performs an approximate nearest neighbour search over the HNSW index to retrieve the top-k plays by dense similarity. The index returns both the indices of the nearest neighbours and their similarity scores.

3.2.2 BM25 Lexical Index

To complement dense retrieval, the system builds a BM25 index over the same coaching shaped text. BM25 is a probabilistic ranking function that computes relevance based on term frequency, inverse document frequency, and document length normalisation, and has been the dominant lexical baseline in ad hoc retrieval for many years.

Each text field is tokenised using a simple lowercasing and word boundary regular expression. The resulting token lists form the input to a BM25Okapi index that precomputes document level statistics and term weights. Once the index is built, a tokenised query is scored against each document using the standard BM25 formula:

$$BM25(Q, D) = \sum_{q \in Q} IDF(q_i) \cdot \frac{f(q_i, D)(k_1+1)}{f(q_i, D) + k_1(1-b+b\cdot\frac{|D|}{avgdl})}$$

Where $f(q_i, D)$ is the term frequency of q_i in D, $|D|$ is the document length, $avgdl$ is the average document length in the corpus, and k_1, b are hyperparameters.

The BM25 index is used to retrieve the top-k documents for each query, together with their BM25 scores. Lexical matching is particularly useful in this domain for capturing exact occurrences of structured tokens such as “SHOTGUN”, “UNDER CENTER”, “SHORT MIDDLE”, team abbreviations, and down and distance phrases.

3.2.4 Hybrid Fusion

To obtain a final candidate set, the system combines dense and lexical scores using a simple hybrid fusion scheme. For each query:

1. The dense retrieval stage returns the top-k documents and similarity scores.
2. The BM25 stage returns the top-k documents and BM25 scores.
3. The two candidate lists are merged based on document identifier, so each document appears at most once.
4. BM25 scores are min-max normalised to the interval [0,1] across the union of candidates.
5. For each candidate document d, a hybrid score is computed as:
   $hybrid\ score = 0.7 \times vector\ similarity + 0.3 \times bm25\ norm$
6. Candidates are sorted by hybrid score and the top-k plays are returned as the retrieval result.

3.3 LLM Play Call Generation

The second stage of the system uses a compact language model to synthesise a play family description from the retrieved neighbourhood. The model used is GPT-4o-mini, a fast, cost efficient variant of OpenAI's GPT-4o series designed for focused tasks and short responses [18].

The language model is not used as a standalone decision maker that observes raw game state. Instead, it acts only on the structured situation description and the retrieved candidate set. For each query, the system constructs two messages:

1. A system message that defines the role of the model as an NFL play recommendation agent and instructs it to base recommendations on the provided historical candidates. The message specifies that the output must be a single play call in a fixed format.
2. A user message which contains:
    a. The original situation
    b. The ranked list of the top-k retrieved plays

The prompt instructs the model to read these candidate plays, consider situational similarity, and synthesise an optimal play call pattern. The model is explicitly told not to mention specific player names or team names in the output and not to provide explanations or alternatives, only a single formatted call.

The OpenAI chat completions API is invoked with GPT-4o-mini, a low temperature (0.3) to reduce randomness, and a small max_tokens budget, reflecting the short expected output length. The system records the end to end model latency for each request.

The raw text returned by the model is normalised by trimming surrounding quotation marks if present. The resulting string is stored as the recommended play family, for example:

SHOTGUN - PASS SHORT MIDDLE to SLOT RECEIVER

To link this abstract family back to concrete historical plays, a simple fuzzy matching step is applied over the retrieved candidate set and the candidate with the highest score is selected as the most similar historical play. This design keeps the language model tightly constrained such that it must operate within the space of retrieved candidates and a simple fixed schema, which reduces risk of hallucinated structures.

3.4 Evaluation

The system is evaluated in two stages: pure retrieval quality and end-to-end recommendation behaviour. In both cases, experiments are conducted on held out situations derived from the same source as the dataset.

3.4.1 Retrieval Evaluation

Retrieval is evaluated as an ad hoc query-document ranking task. A set of *N* red zone situations is sampled, each with an associated target inspiration play identifier. For each situation:

1. The query description is encoded using the E5 embedding model.
2. The retriever returns a ranked list of play identifiers with hybrid scores.
3. The rank position of the target inspiration is recorded if present in the top-k, otherwise the query is treated as a miss.

Effectiveness metrics are computed as follows:

- Hits@K: The proportion of queries for which the target play appears in the top-k results.
- Mean Reciprocal Rank (MRR): The average of the reciprocal ranks of the first relevant result across a set of queries.

Latency is measured by instrumenting the retrieval pipeline to record start and end timestamps for each query. The total retrieval latency includes encoding of the query with E5, FAISS HNSW nearest neighbour search, BM25 scoring, and hybrid fusion. For each experiment, the median, mean and upper percentiles of the latency distribution are reported.

Overall, these measures quantify how often the hybrid system retrieves the designated inspiration play near the top of the ranked list, and how quickly it can do so over the 7,381 play corpus.

3.4.2 End-to-end Recommendation Evaluation

The full pipeline, including the integration with the language model agent, is evaluated on a subset of situations. For each query in this subset:

1. Hybrid retrieval is run to obtain the top-k candidates.
2. GPT-4o-mini is invoked with the system and user prompts described in Section 3.3 to synthesize a play family.
3. The fuzzy matching procedure selects a most similar historical play from the candidate set, or returns no match if the output does not align with any candidate.

The following metrics are computed:

- Top-k coverage: The fraction of queries for which the target inspiration appears within the top K retrieved candidates. This reflects the degree to which the retrieval stage gives the language model access to the ground truth inspiration.
- Play type match: The fraction of queries for which the play type inferred from the generated play matches the play type of the target inspiration.
- Latency: The total time it takes to get a response from the system when given a query.

These metrics focus on whether the system can reliably produce syntactically valid recommendations, whether the retrieval stage supplies the model with the correct inspiration, and whether the model's coarse choice of play type aligns with the labelled inspiration. Direct outcome based metrics such as expected points added cannot be computed in this setting because expected point values are not available in the underlying dataset. Instead, the evaluation concentrates on retrieval fidelity, structural alignment of generated play calls with ground truth labels, and suitability of the system for real time use given the measured latency profile.

4. Results

4.1. Retrieval Performance

Metric	Value	Interpretation
Hits@10	41.0%	Found correct play in top-10 (out of 7,381)
MRR	0.3950	Correct play ranks ~#2.5 on average
Latency p50	38.47 ms	Median retrieval time
Latency p99	59.43 ms	99th percentile retrieval time

Table 1. Performance Metrics for Vector Index + BM25 Retrieval

The table reports retrieval quality and speed for a system that, given a game situation, returns a ranked list of historical plays. Hits@10 measures the fraction of test queries for which the exact ground-truth play appears anywhere in the top ten retrieved items; it reflects coverage of the candidate set that a coach or downstream model would actually inspect. Mean Reciprocal Rank (MRR) summarizes where the correct play tends to appear within that list, weighting higher ranks more heavily (a value of 1.0 would mean "always ranked #1"). The p50 and p99 latency values capture the distribution of query times: the median response time and the near-worst-case (99th percentile) response time, respectively.

Hits@10 = 41.0% indicates that in roughly four of every ten situations, the system includes the exact ground-truth play among the first ten results, meaning it routinely narrows thousands of possibilities to a small, actionable shortlist. MRR = 0.3950 implies that when the correct play is retrieved, it typically appears near the top (around rank 2–3 on average), minimizing review effort and improving downstream selection. On speed, p50 = 38.47 ms and p99 = 59.43 ms show the retriever is both fast and stable; even tail queries complete well under 100 ms, leaving substantial time budget for reranking or LLM reasoning within an in-game decision window.

4.2. LLM Play Call Generation Performance

Metric	Value	Interpretation
Top-K Coverage	38.0%	Ground-truth in retrieved candidates
Play Type Match	36.0%	Same PASS/RUSH as ground-truth
Latency p50	865.91 ms	Median end-to-end time
Latency p95	1310.61 ms	95th percentile (under 1.5s)

Table 2. Performance Metrics for LLM Play Generation

The table summarizes end-to-end behavior of the full pipeline that retrieves candidates and then has the LLM synthesize a play call. Top-K Coverage measures how often the ground-truth play appears somewhere within the retrieved candidate set the LLM reviews; it reflects whether the "right answer" is even available to the generator. Play Type Match checks a simpler behavioral criterion, whether the LLM's recommended call chooses the same high-level action (PASS vs RUSH) as the ground-truth play. The latency metrics (p50 and p95) report total wall-clock time from query to final textual recommendation, capturing both retrieval and LLM generation.

Top-K Coverage = 38% shows that in roughly two out of five scenarios the exact ground-truth play is among the candidates presented to the LLM, making a correct recommendation possible without further retrieval. Play Type Match = 36% indicates that over a third of recommendations align with the ground-truth at the strategic level, even when the exact play may differ. On speed, median end-to-end latency = 865.91 ms keeps typical recommendations comfortably under one second, and p95 = 1.31 s shows that even tail cases remain within an in-game decision window where the coordinator still has time to relay a call.

5. Analysis

We interpret the retrieval results in the context of modern hybrid information retrieval. The BEIR benchmark has become a standard testbed for evaluating dense, sparse, and re-ranking models across heterogeneous tasks, and it repeatedly confirms BM25 as a strong lexical baseline for zero shot retrieval. The E5 family of embedding models is trained with weakly supervised contrastive learning and is the first general purpose dense model reported to outperform BM25 on BEIR in a zero shot setting, while remaining efficient enough for large scale retrieval. Recent hybrid studies that linearly combine BM25 scores with dense similarity show consistent gains in nDCG and recall over either signal alone, both in academic evaluations on BEIR and in industrial analyses [19]. Our retrieval stack uses exactly this configuration: BM25 over the play text, E5 base v2 embeddings indexed in FAISS HNSW, and a simple weighted fusion of the two scores. In other words, our system is not an exotic architecture but a direct instantiation of the now standard hybrid recipe that underpins contemporary QA and retrieval augmented generation systems.

Within that framework, we can situate the reported metrics. The corpus contains 7,381 red zone plays, so random retrieval would place the single labeled inspiration in the top ten only with probability 10 / 7,381 ≈ 0.14%. By contrast, Hits@10 = 41% implies that the exact ground truth play appears in the top ten about 293x more often than chance. MRR = 0.3950 indicates that when the labeled inspiration is retrieved it typically appears around rank two or three. On BEIR style passage retrieval tasks, strong hybrid systems often report nDCG@10 values in the 0.4 to 0.6 range on corpora of similar or larger scale [19]. Given the differences between nDCG and Hits@10, and between our single gold play and multi relevant document setups, a 41% Hits@10 on a 7.3k item corpus is consistent with the performance level of competitive hybrid baselines in standard IR benchmarks.

Latency must be evaluated against the constraints of the football play clock and existing real time analytics. NFL rules specify a 40 second play clock after most plays and a 25 second clock in some administrative situations. Within that window the offense must substitute personnel, huddle, receive the call, and reach the line of scrimmage, so only a fraction of the clock is available for the decision itself. Our retriever responds in a median of 38.47 ms with a 99th percentile (worst case) of 59.43 ms, which is effectively instantaneous at football timescales and leaves almost the entire decision budget available for human deliberation or downstream models. The full pipeline, including LLM play synthesis, has a median latency of 865.91 ms and a 95th percentile of 1,310.61 ms, so even worst typical cases consume roughly one second of a 25 to 40 second clock. For comparison, the league's Next Gen Stats system uses RFID tags and AWS hosted models to compute tracking based metrics in what is described as real time or near real time for both broadcast and internal decision support [20]. Our end to end latency lies well within the same order of magnitude, and retrieval itself is significantly faster than the original sub half second tracking pipeline reported when NGS was introduced. This ensures that the system's speed is sufficient to participate in in-game workflows without distorting normal timing.

The results also highlight how automated retrieval expands the option set beyond what is reachable with conventional call sheets. In practice, a coordinator chooses among perhaps three to five plays that have been preselected for a particular yard line bucket, which is only a tiny fraction of the team's playbook and an even smaller fraction of the league's collective history. Our corpus contains 7,381 league wide red zone plays from a full season, each enriched with situation and recent play history, and the hybrid index can scan that entire space in tens of milliseconds. The system then returns a short list of three to five inspirations that are analogous in length to a human call sheet box but drawn from a much larger and more diverse universe of precedents. The fact that the labeled inspiration appears in the top ten 41% of the time, and in the LLM evaluation lies within the retrieved candidate set 38% of the time, indicates that the same plays a post hoc analyst would identify through careful database work are often surfaced automatically at call time. Taken together, these comparisons suggest that a standard hybrid IR stack, tuned on football specific text, is already capable of delivering recall and latency that match established retrieval baselines and fit comfortably inside the operational constraints of modern NFL play calling.

6. Discussion

6.1 Limitations

This study focuses exclusively on red zone plays from a single NFL regular season, using a 7,381 play corpus as a beachhead for evaluating retrieval and recommendation quality. Restricting to red zone situations simplifies evaluation and aligns with an area of the field where play calling is particularly constrained and consequential, but it does not capture the full diversity of offensive decision making across midfield, backed up, two minute, or four minute scenarios. The retrieval and prompting methodology, however, is not inherently specific to the red zone. The same pipeline of constructing rich situation plus history representations, indexing league wide plays with a hybrid BM25 plus dense retriever, and synthesising play families with a language model could be extended to any down, distance, and field position, as well as to specialised contexts such as third and long, short yardage, or backed up offense, provided comparable play by play data and supervision signals are available.

6.2 Future Work

A natural next step is to replace or augment the FAISS-based index with a generative retrieval model that directly decodes identifiers of inspirational plays. Recent work on generative retrieval and document specific identifiers has shown that language models can learn to map queries into sequences of semantic IDs that index a corpus, allowing retrieval to be performed by autoregressive decoding rather than vector search. In the football setting, one could train a model to emit structured play identifiers that encode personnel, formation, motion, and route family, using our existing hybrid system as a teacher. Such a model would aim to capture latent criteria that coaches use implicitly and could, in principle, provide near constant time retrieval as corpus size grows, since decoding a small number of IDs has fixed cost. Combining this with domain specific embeddings and smaller fine-tuned language models tailored to play description and tagging would form a complete generative retrieval stack for in-game play selection that goes beyond the current two stage retrieve then generate design.

7. Conclusion

This paper formulates NFL red zone play selection as an information retrieval problem and shows that a standard hybrid stack, instantiated with BM25 and E5-base-v2 over a FAISS HNSW index, can act as a practical memory for coaches. Using 7,381 red zone snaps from the 2020 regular season, we construct coaching shaped textual representations that combine situation, recent offensive history, and play description, and demonstrate that hybrid retrieval recovers an analyst designated inspiration in the top ten results 41% of the time, with mean reciprocal rank 0.3950 and sub 100 ms retrieval latency. Coupled with a constrained GPT-4o-mini agent, the system synthesizes structured play family calls and links them to concrete historical analogues with median end to end latency of about 0.87 seconds, comfortably within the effective decision window imposed by the play clock. Although this study focuses on red zone plays from a single season, the methodology of contextual encoding, hybrid retrieval, and LLM based play family synthesis can be extended to other game contexts and provides a foundation for future generative retrieval systems that directly decode semantic play identifiers for in-game decision support.

References

[1] Xu, T., Li, C., Wang, X., Wu, J., Jiang, J., & Wen, J. R. (2025). Beyond negative sampling: Dynamic listwise training for ranking. *arXiv preprint arXiv:2502.14822*.
https://arxiv.org/abs/2502.14822

[2] Li, T., Zhang, X., Sun, Y., Zhu, C., Zhang, J., & He, X. (2024). Is all you need dense retrieval? A comprehensive study of traditional sparse retrieval and dense retrieval models. *arXiv preprint arXiv:2408.06643*.
https://arxiv.org/abs/2408.06643

[3] Rayo, C. L., Valcarce, P., Zamora, J. R., & Parapar, J. (2025). From retrieval to generation: Leveraging dense and sparse retrieval for end-to-end document question answering. *arXiv preprint arXiv:2502.16767*.
https://arxiv.org/abs/2502.16767

[4] Wang, K., Gao, J., Wei, F., & Zhou, M. (2022). Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
https://arxiv.org/abs/2212.03533

[5] Wang, X., Xiong, C., Tay, Y., & Bennett, P. (2021). GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*.
https://doi.org/10.1145/3459637.3481911

[6] Abdallah, M., et al. (2025). Hybrid dense and sparse retrieval: A survey. *arXiv preprint arXiv:2502.20245*.
https://arxiv.org/abs/2502.20245

[7] Elastic. (n.d.). *What is hybrid search?* Retrieved May 2025, from Elastic website.
https://www.elastic.co/what-is/hybrid-search

[8] Vespa. (2024, August 21). *Redefining hybrid search possibilities with Vespa*. Vespa AI Blog.
https://blog.vespa.ai/redefining-hybrid-search-possibilities-with-vespa

[9] OpenSearch. (n.d.). *Hybrid search*. In *OpenSearch documentation*. Retrieved May 2025.
https://docs.opensearch.org/latest/vector-search/ai-search/hybrid-search/index

[10] Kirwan, P., & Seigerman, B. (2010, January 27). Producing a game plan takes into account months of work. *NFL.com*.
https://www.nfl.com/news/producing-a-game-plan-takes-into-account-months-of-work-09000d5d8192b29f

[11] Svrluga, B. (2011, November 13). NFL play-calling is an arduous, collective process. *The Washington Post*.
https://www.washingtonpost.com/sports/redskins/nfl-play-calling-is-an-arduous-collective-process/2011/11/11/gIQAchJEGN_story.html

[12] Laine, J. (2020, September 10). How Buccaneers coach Bruce Arians can reignite the fire in Tom Brady. *ESPN*.
https://www.espn.com/blog/tampa-bay-buccaneers/post/_/id/23838/how-buccaneers-bruce-arians-can-reignite-the-fire-in-tom-brady

[13] King, P. (2015, November 19). Inside the game plan: How Carson Palmer and the Cardinals prepare for the Browns. *Sports Illustrated*.
https://www.si.com/nfl/2015/11/19/nfl-carson-palmer-arizona-cardinals-inside-game-plan-part-ii-cleveland-browns

[14] Adamson, T. (2024, January 29). How NFL Next Gen Stats tech works, explained. *SlashGear*.
https://www.slashgear.com/1482632/how-nfl-next-gen-stats-tech-works-explained

[15] Lemire, A. (2022, November 17). FIFA grants quality mark to Second Spectrum tracking technology. *Sports Business Journal*.
https://www.sportsbusinessjournal.com/Daily/Issues/2022/11/17/Technology/fifa-second-spectrum-tracking-technology-quality-mark-electronic-performance-tracking-systems

[16] Nguyen, T., et al. (2025). [Article on real-time analysis tools for American football]. *Scientific Reports*.
https://www.nature.com/articles/s41598-025-85993-1

[17] BigDataBall. (n.d.). *BigDataBall: Downloadable sports data for analytics*. Retrieved May 2025.
https://www.bigdataball.com

[18] OpenAI. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
https://arxiv.org/abs/2303.08774

[19] Herreros, M., & Veasey, J. (2023, June 21). Improving information retrieval in the Elastic Stack with hybrid search. *Elastic Search Labs Blog*.
https://www.elastic.co/search-labs/blog/improving-information-retrieval-elastic-stack-hybrid

[20] Amazon Web Services. (n.d.). *NFL uses machine learning and analytics on AWS to transform player health and safety and the fan experience*. Retrieved May 2025.
https://aws.amazon.com/machine-learning/customers/innovators/nfl

[21] Mays, R. (2017, August 24). *The misunderstood art of play-calling*. The Ringer.
https://www.theringer.com/2017/8/24/nfl/play-calling-strategy-bruce-arians-sean-mcvay

Appendix


```
{
   "play_id": "ee0a64b9-ba92-4dbf-b451-bf04f11fc450",
   "text": "MIA offense vs BUF defense. Situation: 2 and 5 at 5 yard line, quarter 4. Recent plays: PASS DEEP RIGHT 0yd, PASS SHORT LEFT 0yd, PASS SHORT MIDDLE 7yd, PASS DEEP RIGHT 0yd, PASS SHORT LEFT 0yd. Play: PASS SHORT RIGHT for 5 yards. Description: (12:22) (Shotgun) 1-T.Tagovailoa pass short right to 10-T.Hill for 5 yards, TOUCHDOWN.",
   "play_data": {
     "play_id": "ee0a64b9-ba92-4dbf-b451-bf04f11fc450",
     "play": {
       "formation": null,
       "playtype": "PASS",
       "direction": "SHORT RIGHT",
       "yards": "5",
       "description": "(12:22) (Shotgun) 1-T.Tagovailoa pass short right to 10-T.Hill for 5 yards, TOUCHDOWN."
     },
     "context": {
       "situation": {
         "yardline": "5",
         "down": "2",
         "togo": "5",
         "quarter": "4",
         "offense": "MIA",
         "defense": "BUF"
       },
       "history": [
         {
           "playtype": "PASS",
           "direction": "DEEP RIGHT",
           "yards": "0"
         },
         {
           "playtype": "PASS",
           "direction": "SHORT LEFT",
           "yards": "0"
         },
         {
           "playtype": "PASS",
```

```
          "direction": "SHORT MIDDLE",
          "yards": "7"
        },
        {
          "playtype": "PASS",
          "direction": "DEEP RIGHT",
          "yards": "0"
        },
        {
          "playtype": "PASS",
          "direction": "SHORT LEFT",
          "yards": "0"
        }
      ]
    }
  },
  "reference_count": 8
}
```


Appendix A.1. Example play object from the corpus dataset