Multimodal Information Retrieval for Open World with Edit Distance Weak Supervision

KMA Solaiman
Department of Computer Science
Purdue University
West Lafayette, IN 47906, USA
Email: ksolaima@purdue.edu

Bharat Bhargava
Department of Computer Science
Purdue University
West Lafayette, IN 47906, USA
Email: bbshail@purdue.edu

Abstract*—Existing multi-media retrieval models either rely on creating a common subspace with modality-specific representation models or require schema mapping among modalities to measure similarities among multi-media data. Our goal is to avoid the annotation overhead incurred from considering retrieval as a supervised classification task and re-use the pre-trained encoders in large language models and vision tasks. We propose "FemmIR", a framework to retrieve multimodal results relevant to information needs expressed with multimodal queries by example without any similarity label. Such identification is necessary for real-world applications where data annotations are scarce and satisfactory performance is required without fine-tuning with a common framework across applications. We curate a new dataset called MuQNOL for benchmarking progress on this task. Our technique is based on *weak supervision* introduced through *edit distance* between samples: graph edit distance can be modified to consider the cost of replacing a data sample in terms of its properties, and relevance can be measured through the implicit signal from the amount of edit cost among the objects. Unlike metric learning or encoding networks, FemmIR re-uses the high-level properties and maintains the property-value and relationship constraints with a multi-level interaction score between data samples and the query example provided by the user. We also proposed a novel attribute recognition model from unstructured text "HART" that can identify attributes without finetuning or large language models. We empirically evaluate FemmIR and HART on a missing person use-case with MuQNOL. HART successfully identifies human attributes from large unstructured text without additional training, while FemmIR performs comparably to similar retrieval systems in delivering on-demand retrieval results with exact and approximate similarities while using the existing property identifiers in the system.

I. Introduction

With the influx of multimedia data sources, comparing data from different modalities to grasp a more informed decision for any phenomenon has become increasingly difficult. Humans analyze information across modalities using many indirect cues and common hints. But when it transfers to hours of videos or thousands of documents, it is imperative to have a recommender system that filters out the most important information according to the user's preference and recommends with an unseen item that this user is likely to interact in the future. Existing sequential recommenders propose contrastive learning [1] or attention networks to do representation alignments between multiple modalities based on similarity labels across data from different modalities. Without the similarity labels, it is impossible to adapt these retrieval models by fine-tuning them to certain applications, which is the most common scenario for most real-world use cases.

Example 1 (Application-specific Information Need): An organization wants to build an automated system to find video frames containing persons of interest from many hours of video feeds, connect them to previous occurrences from incident reports, and find patterns among these occurrences. Alex is asked to develop a machine learning (ML) pipeline over these datasets to predict the videos where the person mentioned in the text would be found, and, subsequently, the authority would look for them in those videos. Alex decides to use an off-the-shelf retrieval algorithm that is trained over video and text. However, the performance was not satisfactory, as: (1) Alex could not modify the model to focus on specific properties that are most common for a missing person as he did not know which video frames contained the person mentioned in the reports, and (2) he could not perform transfer learning as the annotated data is very difficult to achieve in this case, where one positive case occurs in 8-10 hours of video. Now he wonders: (1) how can he re-train the retrieval model without any training data to focus similarity on the desired properties? (2) if he runs a property identifier in each data modality and performs only explicit matching would that achieve the desired performance? (3) how can he map similar properties that are described differently from each modality?

To address these issues, we propose a **Fe**ature-centric **M**ulti**m**odal **I**nformation **R**etrieval model for open-world environment, **FemmIR**. Our framework designs multiple plug-and-play components with effective representation alignment and matching objectives to enable ranked information retrieval across application domains and modalities. Specifically, we leverage pre-trained text and vision property identifiers as feature extractors. The modality-specific high-level features are fused into multimodal item representation via a graph representation approach and an attention network, which is subsequently processed by a graph similarity approximation model to capture the implicit similarities. Since we assume no similarity label is available across multiple modalities, we needed weak supervision from other sources [2], [3], [4], [5].

We hypothesize that capturing how much change is needed to convert a data sample to another can provide us with a source of weak supervision. Considering a data sample as a collection of objects with certain properties along with the relationships among them, we modeled a novel distance metric based on graph edit distance (GED).

We introduce a new benchmark and dataset called MuQNOL (**Mu**ltimodal **Q**ueries with **NO** similarity **L**abel) to train and evaluate models to retrieve the relevant data from a multimodal corpus given multimodal (vision + language) queries without any similarity labels. To create this dataset, we start with the MARS [6] dataset as a source – MARS is a large-scale dataset of pedestrian image sequences with 16 annotated attributes. We combined it with an unstructured text dataset, InciText, consisting of incident reports, press releases, synthetic reports, and officer narratives from old police cases. We also annotated InciText [7] for three attributes with a wide range of possible values. We proposed a novel **H**uman **A**ttribute **R**ecognition model from unstructured **T**EXT, HART to identify these attributes from InciText. From the 16 attributes in MARS, we select common attributes in MARS and InciText for MuQNOL where the retrieved answer includes both an image, video, and text.

Unlike existing multimodal retrieval models, FemmIR does not require a large amount of training data, and data representations can be aligned through the weak supervision. Among existing works, correlation learning methods [8], [9], [10], [11], [12], [13], [14] linearly or non-linearly projects low-level features from representation models to a common subspace. Metric learning methods [15], [16], [17] learn a distance function over data objects based on a loss function to map them into the common subspace. FemmIR closely relates to metric learning methods. Contrary to them, we do not directly correlate class labels or weak labels to the loss function. The proposed edit distance between property graphs implicitly captures the signal for relevance. In contrast to common representation learning models, *data discovery* models based on relational queries allow more flexibility to consider explicit information needs from users, and use high-level properties in the system. EARS [18] is one such content-based data discovery system that, similar to our approach, takes user examples as queries and delivers relevant multi-media results. However, the prime aspect of EARS is it assumes a schema mapping among all modalities, the number of JOIN queries increases as a product of the number of properties-of-interest and modalities, and to introduce new modalities the common schema needs to be updated. In contrast, FemmIR offers a general solution to include retrieval from novel modalities for a diverse set of systems and does not need an explicit design for each new modality.

Our contribution and findings are listed below.

- We introduce a new dataset MuQNOL to facilitate research on multimodal information retrieval for real-world use cases without similarity labels.
- We propose an end-to-end retrieval model, FemmIR, that delivers ranked results from multimodal data relevant to a given multimodal query using weak supervision from a novel distance metric CED. FemmIR does not need any similarity label and can be pre-trained on any application domain for faster inference time.
- We benchmarked another end-to-end model, EARS on MuQNOL dataset. EARS is an exact inference model for multimodal data retrieval. We observed that FemmIR is a capable multimodal retriever that surpasses existing multimodal knowledge retrieval methods without fine-tuning.
- We proposed HART, to retrieve attribute values from the unstructured text as part of the basic property identifiers for the FemmIR framework.

II. Preliminaries & Problem Definitions

In this section, we first provide formal definitions relevant to our proposed methods. We then proceed to formulate the problem of multimodal information retrieval for unconstrained data for property-specific information needs, and the problem of object-property identification from text.

Definition 2.1 (Attributed relational graph): An attributed relational graph (ARG) is a graph whose nodes and edges have assigned attributes (single value or vectors of values). Although we focus our methodology only on directed and labeled graphs, it is designed to handle any form of graphs. An ARG is defined as: $g = (N, E, l)$ where

1) N is the finite set of nodes,
2) $E \subseteq N \times N$ is the set of edges,
3) $l : N(g) \cup E(g) \rightarrow \Sigma$ is a labeling function that assigns labels to each vertex and edge from Σ.
4) Σ is a set of unconstrained labels. $A \in \Sigma$ represents labels enumerating the node-type.

Definition 2.2 (Wu-Palmer Distance): Wordnet [19] is a lexical knowledge base where words are organized in a hypernym tree based on their origin. Wu-Palmer distance calculates the similarity between word meanings based on the similarity between the word senses and the location of the synsets relative to each other in the hypernym tree. Given the synsets of two strings s_{t_1} and s_{t_2}, and the LCS (Least Common Subsumer) between them, the Wu-Palmer distance is:

$$wpdist(s_{t_1}, s_{t_2}) = 2 * \frac{depth(lcs(s_{t_1}, s_{t_2}))}{depth(s_{t_1}) + depth(s_{t_2})} \quad (1)$$

Definition 2.3 (Natural Language Inference): Given a hypothesis h and a premise p, natural language inference (NLI) is the task of determining the probability Pr of the hypothesis being true (entailment E), false (contradiction C), or undetermined (neutral N). NLI determines the best label l:

$$\arg \max_{l \in \{E,C,N\}} Pr(l \mid h, p) \quad (2)$$

A. Problem Definitions

Assuming a collection of data from $\mathcal{M} \in \mathbb{Z}^+$ modalities, we denote the set containing $n_i \in \mathbb{Z}^+$ samples from the i-th modality as $\mathcal{D}_i = \{\mathbf{d}_1^i, \mathbf{d}_2^i, \ldots, \mathbf{d}_{n_i}^i\}$, where j-th sample of the i-th modality is $\mathbf{d}_j^i$.

Any data sample $\mathbf{d}_j^i$ is described with a subset of object properties, $\mathcal{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_l\}$ where z_r is the set of values of $\mathbf{o}_r$. Property identifiers implement a relation, *PROP* $(\mathbf{d}_j^i) \subset \mathcal{O}$ that maps a data-sample to a set of object-properties. A query is issued against a corpus with $\mathcal{M}$-modalities, $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_\mathcal{M}\}$.

Problem 2.1 (Multimodal Information Retrieval): Given a query in modality m, $\mathbf{d}_q^m$, the task is to retrieve a ranked list, $R = (\mathbf{d}_1^{x_1}, \mathbf{d}_2^{x_2}, \ldots \mathbf{d}_t^{x_t})$ of $t \in \mathbb{N}_0$ data-samples from all available modalities in the system satisfying *PROP* $(\mathbf{d}_q^m)$. $\mathbf{d}_q^m$ can be expressed in two different ways:

(1) (Query-by-Properties) $\mathbf{d}_q^m$ with p object-properties $\{\mathbf{o}_1 = z_1, \mathbf{o}_2 = z_2, \ldots, \mathbf{o}_p = z_p\}$, or

(2) (Query-by-Example) An example data-sample, $\mathbf{d}_q^m$ with the *PROP* relation.

Relevance is scored based on the degree of common object-properties between a data-object $\mathbf{d}_c^{x_c}$ in the ranked list, and the query data $\mathbf{d}_q^m$, *PROP* $(\mathbf{d}_q^m) \cap$ *PROP* $(\mathbf{d}_c^{x_c})$. A similarity score is used to define the degree of relevance,

$$0 \leq \text{SIM}\ (\mathbf{d}_c^{x_c},\ \mathbf{d}_q^m) \leq 1.$$

Similarity score of 0 indicates non-relevance, whereas a score of 1 indicates complete relevance and a proper subset, *PROP* $(\mathbf{d}_q^m) \subset$ *PROP* $(\mathbf{d}_c^{x_c})$.

1) Property Identification from Unstructured Text: As a relevant sub-task, we explore the problem of identifying properties describing humans from unstructured text. As discussed in SurvQ [20], a finite number of visible and approximate properties such as, GENDER, RACE, BUILD, HEIGHT, CLOTHES, etc. are used in describing a person-of-interest to search for them. We denote these properties for person identification as $\mathcal{O}_H$.

Example 2: The sentence "a †`person with white ethnicity and` **`medium`** `build was seen in Vernon St.,` *`wearing`* **`white jeans`** `and` **`blue shirt`**" describes properties of a person:

1) BUILD = `medium`,
2) *CLOTHES = {`jeans, shirt`},
3) UPPER-WEAR-COLOR= {`white`},
4) BOTTOM-WEAR-COLOR = {`blue`}, and
5) RELATION = {`wearing, †Person, *Clothes`}.

Problem 2.2 (Human Attribute Recognition from Text): Given a large text T with T_s sentences, each with $|w|$ tokens, the problem of human attribute recognition from T is to

1) identify the set of sentences $C_s \subset T_s$ that describes properties of a person,
2) expose the set of object-properties $\mathcal{O}_H$ from C_s and
3) extract the set of values z_p of the identified properties $\mathbf{o}_p$.

Our problem setting assumes that the set of key-phrases (Q_H) often used in sentences describing properties of a person are either known (provided by domain experts), or a small amount of annotated documents are provided to identify Q_H manually. In Example 2, Q_H = {`wearing`}. The first assumption is derived from the literature on pedestrian attribute recognition from visual and textual modalities, and the second assumption is computationally inexpensive. Note that, $(Q_H \cap \mathcal{O}_H) \neq \{\phi\}$.

Definition 2.4 (Candidate Sentences): Given a collection of sentences T_s, key-phrase for describing an object in text $q_H \subset Q_H$, and an empirical threshold θ_H, Candidate sentence is

$$C_s = \{s : s \in T_s, q_H \in Q_H \mid SIM(q_H, s) > \theta_H\} \quad (3)$$

III. MULTIMODAL INFORMATION RETRIVAL

Our proposed similarity matching algorithm considers the data samples from the data repository or the data streams, $\mathbf{d}_c^{x_c}$ and user-provided example, $\mathbf{d}_q^m$ as input and outputs the similarity score between them: SIM $(\mathbf{d}_c^{x_c}, \mathbf{d}_q^m)$. The corresponding object properties are assumed to be extracted by the system-specific property identifiers in the pre-processing stage. We propose a novel distance metric to rank the data samples based on the number of edits needed to convert the properties of one sample to another instead of manually annotating or aggregating the number of matched object properties. To this end, we first process the extracted properties from the input data samples with a graph encoding mechanism which converts the properties into a hierarchical attributed relational graph (HARG) and generates a graph representation for each sample. Then FemmIR adopted the Munkers' algorithm [21] to calculate the proposed edit distance between the data samples and use it as a similarity label. Finally, we used an edit distance approximation algorithm with Neural Tensor Network (NTN) to learn a function to map the graph embedding of the HARGs to a similarity score between the data samples. During inference, the model just takes the extracted properties from the data samples and outputs the similarity score by using the mapping function. We start with a use case to demonstrate the information retrieval task and our observations that led to the proposed system.

A. Graph Representation for Data Sample

Consider the task of finding the location of a person from a large amount of video data using text queries or reports. The system finds the video feeds that have persons similar to the report description by focusing on the object properties of the person in the video and text. The goal is to identify the similarity score between video feeds, text queries, and incident reports which can then be used to deliver a ranked list of relevant data samples to the user. We make the following observations for property-specific multimodal queries:

OIII.1 The number of object properties between two data samples is finite, and the values of the properties are mostly categorical values.

OIII.2 A data sample can describe a large number of objects and object properties, but for system-specific similarity comparison, a user is only interested in a finite number of properties.

OIII.3 Data samples are special objects with different properties such as metadata, topics, and events.

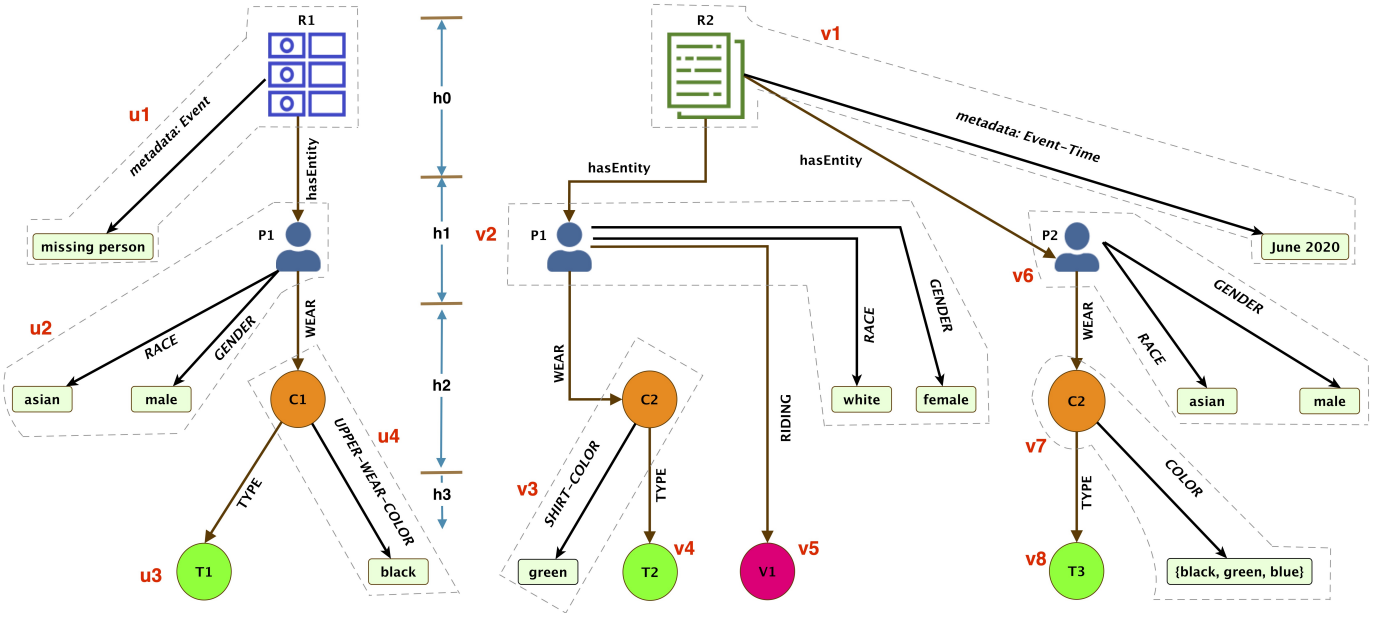


Fig. 1: HARG and Weak Label Generation; Left sided graph refers to g^q, and the right sided graph refers to g^c. Node-type labels are as follows. V: EPL Vertex, R: Root, P: Person, C: Clothes, T: Type, M: Motor-Vehicles. Squared nodes correspond to the non-empty leaf nodes.

OIII.4 *Relationships between objects* are specific types of object properties that belong to all participating objects. The set of values corresponding to the objects would be complementary to each other. Value for relation-name can be different for the same relationship through different data samples. For example, different text would describe the same action in different forms: *wearing, wear, has*.

OIII.5 Some properties in z_p have single and fixed value-set i.e., GENDER, RACE, HEIGHT, while other properties have multiple values in their value-set i.e., CLOTHES.

OIII.6 Some object properties such as, CLOTHES-COLOR have different values for different data samples. For example, in Figure 1, `UPPER-WEAR-COLOR`, `SHIRT-COLOR`, `COLOR` all refer to the color of clothes.

Our intuition here is that entities, relationships, and object properties in a data sample have an inter-connected structure, and if we can capture the number of changes to convert one structure to another, then we can capture the differences between these samples. Based on this intuition, FemmIR starts by constructing a *hierarchical attributed relational graph*, called (HARG), with a common hierarchy for all data samples. The choice of graph as a representation was influenced by the need for a data structure with representation-invariant encoding mechanisms that can capture the syntactic similarities between different values.

Definition 3.1 (Hierarchical Attributed Relational Graph): HARG is a specific type of ARG in the form of a multi-level tree with $|h|$ levels. It consists of a root node, multiple levels of nodes and edges emanating from it, and specific type of leaf nodes. Nodes at level h are denoted by N^h.

*a) **CONSTRUCT-HARG**:* Each data sample is represented as HARG, following the steps:

1) The graph starts with a single node at level 0 $(h = 0)$ containing a common label across all data samples in the same application domain: $l(N^0) = \{ROOT\}$.
2) Level 1 nodes constitute the object-properties of the data sample itself where the property name is the edge label, and the property value is the node label:

$$l(N^0, N^1) = \mathbf{o}_p, l(N^1) = z_p.$$

With the exception of o_p being an entity, N^1 would be a leaf node. For entities, we define the edge label as $l(N^0, N^1) = \{hasEntity\}$.
3) In case a set of $\mathbf{o}_p$ describes the attributes of an entity, $N^k(k \geq 1)$ will be a pointer to the attribute properties of that entity, whereas $l(N^k) = \{entity\text{-}type\}$.
4) We categorize entities in two groups for each data sample: *primary*, and *secondary*. Level 1 of HARG only contains primary entities.
5) Level 2 and subsequent levels contain the attribute values of the entities in the previous level with

$$l(N^k, N^{k+1}) = \mathbf{o}_p(k \geq 1), and$$

$$l(N^k) = z_p(k \geq 2).$$

From Definition 3.2, for RELATION properties, $\langle R, S, Arg \rangle$ where entity-pointer S is at level-k and entity-pointer Arg is at level-$(k+1)$,

$$l(N^k, N^{k+1}) = R, l(N^k) = S, l(N^{k+1}) = Arg.$$

6) There can be edges between entities in the same level with RELATION properties, R. With nodes N^k and N^r, $l(N^k, N^r) = R$, where $l(N^k) \neq l(N^r)$ but $k = r$.
7) The leaf nodes of HARG always contain a property-value or a NULL value for $z_p = \{\phi\}$.

Figure 1 demonstrates two examples of hierarchical attributed relational graphs from the MuQNOLdataset. $R1$ and $R2$ refer to two different data samples. For the leaf nodes $T2, M1$, and $T3$ in $R2$, $z_p = \{\phi\}$. `Wear`, and `riding` refers to the RELATION property, where `Persons` are subjects, and `Clothes` and `Motor-vehicles` are arguments.

*Definition 3.2 (*RELATION *between Objects):* For a n-ary relationship R, identifiers associate each action with multiple entity arguments, Arg_1, Arg_2, ..., Arg_i, ..., Arg_n with *role* R^i_o. n-ary relationships are broken into multiple binary relationships with

$$l(N^k, N^{k+1}) = \{R : R^i_o\}, l(N^k) = S, l(N^{k+1}) = \{Arg_i\}.$$

We made two assumptions for the generation process: (**I**) we assume prior knowledge of the system-specific properties, (**II**) the entity types for node labels are system-specific and must be consistent through the lifetime of the system. This assumption is valid since the property identifiers from each modality would be system-specific and extracted object types would be consistent across data samples.

b) Graph Embedding: For calculating the graph embedding, first, Graph Convolutional Networks (GCN) [22] are used on the HARG to obtain the node embeddings. GCN is representation-invariant and allows us to account for different kinds of labels for nodes and edges. It is also inductive and allows computing the node embedding for any unseen graph following the GCN operation, which makes it a great choice for variable-sized FemmIR graphs. Then, a global context-aware attention network is used to combine the node embeddings into a graph embedding. This allows FemmIR to learn the importance of each feature in the similarity determination as part of the end-to-end network.

B. Similarity Label Generation with Content Edit Distance

FemmIR further defines a new distance metric, Content Edit Distance (CED) using a variation of the Munkres' algorithm [21] to calculate the amount of edits (changes) for optimal alignment of the query-example HARG to HARG of another data-sample. CED is considered as weak label for the retrieval task for two reasons: 1) Munkres' algorithm is suboptimal as it only calculates approximate edit distance values, 2) the quality of HARG rely on the choice of primary entity selection which can be noisy. Our intuition was graph edit distance (GED) calculation algorithms (A*-search, VJ, or Beam) would be enough to calculate the number of changes after we have build the HARGs, but we made following observations.

OIII.7 Different nodes and edges in HARG have different change cost. User should be allowed to specify individual property replacement cost.

OIII.8 GED calculation algorithms differ in speed based on the number on nodes and HARG contains variable sized graphs.

OIII.9 Object-properties such as, RELATION have dependency between different levels of HARG and should not be considered individually during the change estimation. For example, for *person wearing clothes*, edit cost for `person` and `cloth` should be considered together between different data-samples.

OIII.10 Considering **OIII.5**, we cannot calculate the edit cost of certain properties just by replacing or deleting them since they have multiple number of values in their value-set.

For properties with list values, we consider two types of comparison: **(PRDIST)** ordered comparison with modified *Levenshtein distance*, and **(HCOMP)** unordered comparison with *hash table*. Summing the cost of edits for all the properties between two data-samples ignores the inter-connected structure among the properties. In Figure 1, the graph from $R2$ has two persons, and while comparing with $R1$ we would want to know the minimal edit cost by considering which person in $R2$ is closer to the person described in $R1$. CONTENT EDIT DISTANCE calculates the cost for the minimal cost alignment of one data-sample to another. Since only property values in leaf nodes in a HARG have direct replacement cost, we propose a new kind of vertex in HARG, *Entity-with-Property-in-Leaf (EPL) vertex* (Definition 3.3) for calculating the cost for an individual object assignment. Given EPL(V) is the finite set of EPL Vertices, EPL(E) $\subseteq$ EPL(V) $\times$ EPL(V) is the set of edges, and EPL(L) $\subset l$ is the labeling function, a HARG is now defined as:

$$g_{\text{epl}} = (\text{EPL(V)}, \text{EPL(E)}, \text{EPL(L)})$$

Definition 3.3 (Entity-with-Property-in-Leaf Vertices): A node labeled with object-type (A) with their outgoing edges labeled with object-properties ($\mathbf{o}_p$) and the connected leaf nodes labeled with property-values (z_p) are considered as ENTITY-WITH-PROPERTY-IN-LEAF (**EPL**) Vertex, EPL(V). A node without any leaf nodes is also considered as an EPL vertex. An EPL vertex can be connected to other EPL vertices and have its own cost functions.

*a) **Munkres Algorithm for CED calculation**:* We consider the CED calculation as an assignment problem and adopted the bipartite graph matching method in [21]. Compared to the exponential time-complexity of A*-search, Munkres' [21] algorithm has a polynomial time complexity. Estimating content edit distance instead of a simple property-to-property comparison allows the flexibility to consider the dependency between properties and graph levels. Given the non-empty HAR graph from query-example, $g^q_{\text{epl}} = (\text{EPL(V)}^q, \text{EPL(E)}^q, \text{EPL(L)}^q)$ and the HAR graph from the compared data-sample, $g^c_{\text{epl}} = (\text{EPL(V)}^c, \text{EPL(E)}^c, \text{EPL(L)}^c)$, where $\text{EPL(V)}^q = \{u_1, \ldots, u_n\}, \text{EPL(V)}^c = \{v_1, \ldots, v_m\}$, the Munkres' algorithm would output CED (g^q_{epl}, g^c_{epl}). We made the following adjustments to the Munkres' algorithm in [21].

1) EPL-vertices in the query graph need to be aligned to the data-samples, hence we will fix the assignment size k to $|\text{EPL(V)}^q|$.
2) For data retrieval, the entities and relations in query graph needs to be in comparison-graph, otherwise indicates missing property. So there is no need to add dummy nodes to g^q_{epl}. Formally, if $n > m$, only the costs for $max\{0, m-n\}$ node insertions have to be added to the minimum-cost node assignment.
3) Next, the $n \times m$ cost-matrix C is generated. (1) For different type of objects A in u_i and v_j the replacement cost is set to ∞. (2) The cost for a single object assignment $C_{i,j}$ is calculated by comparing the property values z_p (normal-comparison and list-comparison) in EPL-vertex u_i and v_j.
4) To accommodate for **OIII.6**, while applying Adjacency-Munkres, we set the default cost of an edge replacement $c(e_{u_i} \rightarrow e_{v_j})$ based on the Wu-Palmer distance between Synsets of $l(e_{u_i})$ and $l(e_{v_j})$. e_{u_i} denotes all edges connected to u_i and e_{v_j} denotes all edges connected to v_j. In general, any language embedding can be used instead of Synsets.

$$c(e_{u_i} \rightarrow e_{v_j}) = 1/wpdist(s_{l(e_{u_i})},\ s_{l(e_{v_j})}) \tag{4}$$

*b) **Cumulative-Munkres**:* Using Adjacency-Munkres from [21] allows us to find the optimal assignment of each EPL vertex without taking into account the dependency among them **OIII.9**. We utilize the levels from HARG to include the dependency information into the cost-matrix. So for every $C_{i,j}$ in the cost matrix from adjacency-munkres denoting an assignment of u_i to v_j, we add their parent EPL-vertices assignment cost to $C_{i,j}$, starting from EPL-vertices in level-1. In the remainder of this paper, we will call this method CUMULATIVE-MUNKRES since it uses the cumulative cost of the parent and child nodes to preserve the dependency information.

C. Approximate CED Inference

Finally, we propose to use an end-to-end neural network model, SimGNN [23] to learn an embedding function to map d_q and d_c into a similarity score based on the CED score. User requirements (such as relationships between properties, searching in a time range, or within a specified location, etc.) and different system constraints are considered as function parameters with appropriate replacement costs while calculating CED. Similarity scores for training the model are derived by normalizing the distance scores [24] and applying an exponential function on the normalized score. (Line 26 in Algorithm 1). The embedding function outputs a number of interaction scores between the pair of graphs using Neural Tensor Networks (NTN) [25] on the graph embeddings. Finally, a multi-layer fully connected network is applied to learn a single similarity score from the interaction scores, which is compared against the weak CED labels or the ground-truths using mean squared error loss.

$$\mathcal{L}_{mse} = \frac{1}{|D|} \sum_{d_q, d_c \in D} (\hat{s} - s(d_q, d_c))^2 \tag{5}$$

where D is the set of data samples from the repository or the stream, $\hat{s}$ is the predicted similarity score, and $s(d_q, d_c)$ is the ground-truth similarity between d_q and d_c. This similarity score allows us to rank the data samples against the query example.

D. FemmIR algorithm

Algorithm 1 FemmIR

Input: Query example and a single Data sample, d_q and d_c
Replacement cost for property $\mathbf{o}_p$, RCOST($\mathbf{o}_p$)
Insertion cost for property $\mathbf{o}_p$, ICOST($\mathbf{o}_p$)
Output: Similarity score between d_q and d_c, SIM (d_q, d_c)

1: $\mathcal{O}^q \leftarrow$ PROP(d_q), $\mathcal{O}^c \leftarrow$ PROP(d_c)
2: $g^q \leftarrow$ CONSTRUCT-HARG ($\mathcal{O}^q$)
3: $g^c \leftarrow$ CONSTRUCT-HARG ($\mathcal{O}^c$)
4: **if** training **then**
5: $g^q_{\text{epl}},\ g^c_{\text{epl}} \leftarrow$ DISCOVER-EPLV($g^q,\ g^c$)
6: C $\leftarrow \phi$
7: **foreach** $u_i \in$ EPL(V)^q **do**
8: **foreach** $v_j \in$ EPL(V)^c **do**
9: **if** TYPE(u_i) $\neq$ TYPE(v_j) **then** C$_{i,j}$= ∞
10: **foreach** $\mathbf{o}_p \in u_i$ **do**
11: **if** $\mathbf{o}_p \notin v_j$ **then** C$_{i,j}$ += ICOST($\mathbf{o}_p$)
12: **else if** TYPE(z_p) is not list **then**
▷ $z_p(u_i)$is value of $\mathbf{o}_p$ in vertex u_i
13: **if** $z_p(u_i) \neq z_p(v_j)$ **then**
14: C$_{i,j}$+= RCOST($\mathbf{o}_p$)
15: **else**
16: C$_{i,j}$+= 0
17: **else**

$$\text{C}_{i,j} \mathrel{+}= \{\text{oCOMP} * \text{prDIST}(z_p(u_i), z_p(v_j)) + (1 - \text{oCOMP}) * \text{hCOMP}(z_p(u_i), z_p(v_j))\}$$

18: C$_{i,j}$= C$_{i,j}$+ $min\{\sum c(e_{u_i} \rightarrow e_{v_j})\}$
19: **if** MTYPE **then**
20: **foreach** $u_i \in$ EPL(V)^q **do**
21: **foreach** $v_j \in$ EPL(V)^c **do**
22: $u_{\hat{i}} = parent(u_i), \qquad v_{\hat{j}} = parent(v_j)$
23: $C_{i,j} = C_{i,j} + C_{\hat{i},\hat{j}}$
24: CED(g^q, g^c) = MUNKRES (C)
25: nCED= $\frac{CED(\ g^q, g^c\ )}{(|g^q|+|g^c|)/2}$
26: SIM (d_q, d_c) = e^{-nCED}
27: **else**
28: SIM (d_q, d_c) = SIMGNN(g^q, g^c)

Algorithm 1 presents the pseudocode of our retrieval algorithm FemmIR which takes two data samples as input and returns the similarity score between them as output.
Line 1 Extract the set of properties and their values, $\mathcal{O}^j$ from data-sample d_j using the modality-specific property-identifiers.

Lines 2 - 3 Construct the Hierarchical Attributed Relational Graphs using the identified properties following the steps in Section III-A.
Lines 4 - 26 During training, generate the CED as weak label using the Munkres algorithm. CED is used to calculate the similarity score, and this pair of data-samples and the similarity score is added as training sample for SIMGNN.
Line 5 Calculate the EPL-vertices in the HARGs, g_{epl}.
Line 6 Initialize an empty $n \times m$ cost-matrix C.
Lines 7 - 8 Iterate through all the vertices in EPL(V)^q and EPL(V)^c and compare the properties in each vertex to assign the costs.
Line 9 For different types of object, set the cost to ∞, not allowing different types of object to be aligned.
Line 11 If a property in u_i is absent in v_j, it needs to be inserted in v_j. Increment the cost-matrix value by the insertion-cost.
Lines 12 - 16 If the property is not a list, then just compare the values in u_i and v_j. If they mismatch, add the replacement cost to the cost-matrix, otherwise nothing is added.
Line 17 If the property is a list, we need to compare them either with a Levenshtein distance (ordered comparison) or with a hashmap (unordered comparison) from Section III-B. oCOMP is a control variable to specify what kind of comparison is required. The overall cost is added to cost-matrix.
Line 18 For applying Adjacency-Munkres, the minimum edge replacement cost is added to the cost matrix using Equation 4.
Lines 19 - 23 If Cumulative-Munkres is required (set by MTYPE), cost-matrix entry of the parent vertices are added to each $C_{i,j}$.
Line 24 Apply the Munkres algorithm to calculate the optimal assignment based on C, and the associated cost is the CED.
Line 26 Normalize CED to the graph sizes and apply an exponential function to convert it to a similarity score in the range of (0, 1]. Add it to training sample for SIMGNN.
Lines 27 - 28 During inference, apply the learned mapping function to predict the similarity score from the HARGs and rank based on that.

Generalization:

1) Algorithm 1 assumes that the edge labels for level 0 are fixed to $hasEntity$ and $metadata$ with granularity (such as $time, location$, etc.). These are flexible and can be set to any labels in FemmIR as long as it is consistent throughout the lifetime of the system.
2) Object-types are assumed to be system-specific, and can be variable across different systems and applications. FemmIR can handle any labels for entity-type since the retrieval result does not depend on it. The comparison between properties is affected by it which remains valid as long as the same heuristics are maintained for all modalities in a system.
3) FemmIR is capable of handling different replacement costs and insertion costs for properties in different application domains.
4) For the edge replacement cost, any language embedding will work as long as the objective function places semantically similar tokens closer to each other.

IV. HUMAN ATTRIBUTE RECOGNITION FROM UNSTRUCTURED TEXT (HART)

We now describe the property identification technique for unstructured texts to extract *attribute-based properties* from large text documents. Our algorithm considers the full document as input and reports a *collection* of object-properties and their set of values, as output. To this end, we first identify the candidate sentences C_s from a collection of sentences T_s by searching for the key-phrases (q_H) using pre-trained language representation models and lexical knowledge bases. Then, we propose individual property-focused models to extract the attributes and their corresponding values using the syntactic characteristics (i.e., parts-of-speech) and lexical meanings of the tokens in the *Candidate Sentences*. Our heuristic search algorithm, POSID iteratively checks the tokens in the candidate sentences and based on the assigned tags in accordance with their syntactic functions identifies the properties in $\mathcal{O}_H$ and their values.

A. *Candidate Sentence Extraction*

A naive approach to this task would be to consider it as a supervised classification problem given enough training data. Since during this work, the primary goal was to define on-demand models that works in absence of training data, we designed this as a similarity search problem using pre-trained and lexical features, where the similarity between sentence and key-phrase needs to reach an empirical threshold. We now proceed to describe the different methods used to identify C_s.

*a) **Pattern Matching**:* As a baseline heuristic model, we implemented the **REGULAR EXPRESSION (RE)** Search on T_s. Since we consider all sentences in the document as input corpus, if it describes multiple persons, this model captures all of the sentences describing a person as C_s. Individual mentions are differentiated in later stages. For RE, $SIM(q_H, s) \in \{0, 1\}$. Given the key-phrase q_H, the RE pattern searches for any sentence mentioning it:

$$[\hat{}\,]*q_H\,[\hat{}\,.\,]+$$

*b) **Similarity using Tokens**:* Similarity between q_H and s is calculated based on the similarities between tokens $w \in s$ and q_H. A single model is used to embed both w and q_H into the same space. We used two different token representation models for token to query phrase similarity.

$$SIM(q_H, s) = \max_{w \in s} SIM(q_H, w) \tag{6}$$

(a) ***Word Embedding.*** Tokens in each sentence and in the key-phrase are represented by **WORD2VEC** [26] embeddings. If there are multiple tokens in a key-phrase, the average of the embeddings are used. We use cosine similarity as the distance

metric. Given u_{q_H} and u_w are the final embedding vectors for q and w,

$$SIM(q_H, w) = cos(u_{q_H}, u_w) = \frac{u_{q_H} \cdot u_w}{\|u_{q_H}\| \cdot \|u_w\|} \quad (7)$$

(b) ***Word Synsets.*** Tokens and key-phrases are represented by **WORDNET** [19] synsets in NOUN form. For similarity/distance metric, we used the Wu-Palmer similarity [27]. Given the synsets of q and w are s_{q_H} and s_w,

$$SIM(q_H, w) = wpdist(s_{q_H}, s_w) \quad (8)$$

*c) **Classification Model**:* The similarity search problem is re-designed as a classification problem where the sentences are considered as input sequences, and the key-phrases are considered as labels. The probability of sequence s belonging to a class q_H is then considered as the similarity between a sentence and a key-phrase. To that end, following Yin et al. [28], we used pre-trained natural language inference (NLI) models as a ready-made zero-shot sequence classifier. The input sequences are considered as the NLI premise and a hypothesis is constructed from each key-phrase. For example, if a key-phrase is `clothes`, we construct a hypothesis *"This text is about clothes"*. The probabilities for *entailment* and *contradiction* are then converted to class label probabilities. Then, both the sequence and the hypothesis containing the class label are encoded using a sentence level encoder Sentence-BERT [29] (**SBERT**). Finally, we use the NLI model to calculate the probability P. Given SBERT embedding of a sequence s is denoted with B_s,

$$SIM(q_H, s) = P(s \text{ is about } q_H \mid B_s, B_{q_H}) \quad (9)$$

*d) **Stacked Models**:* While **RE** search relies on specific patterns and returns exact matches, the other models calculate a soft similarity, $0 \leq SIM(q_H, s) \leq 1$. Hence if initial results from **RE** search returns no result for all the key-phrases we use **WORDNET** or **SBERT** model to identify semantically similar sentences to the key-phrases.

B. *Iterative Search for Properties*

We now formally describe the POSID algorithm, which uses the models described in Section IV-A in the first stage. We start with the observations that led to the POSID algorithm.

a) Observations: We proposed POSID based on the following observations:

OIV.1 Common to **OIII.5**, object-properties have single and multiple value contrasts.

OIV.2 Some properties follow specific patterns such as GENDER = {male, female, man, woman, binary, non-binary, ...}, whereas some properties have variable values, as shown in **OIII.6**.

OIV.3 Adjectives (ADJ) are used for naming or describing characteristics of a property, or used with a NOUN phrase to modify and describe it.

OIV.4 Property values can span multiple tokens, but they tend to be consecutive.

OIV.5 Property values for CLOTHES generally include the color, a range of colors, or a description of the material.

OIV.6 CLOTHES usually is described after consecutive tokens with VERB tags, V_{DG}, such as, gerund or present participle (VBG), past tense (VBD) etc. If proper syntax is followed, an entity is described with a VBD followed by a VBG. In most cases, mentioning `wearing`.

OIV.7 After a token with VBG tag, until any ADJ or NOUN tag is encountered, any tokens describing the set P_{DCP}, {Determiner, Conjunction, Preposition}, or a Participle, or Adverb is part of the property-name. An exception would be any {participle, adverb, or verb}, P_{PAV} preceded by any {pronouns or non-tagged tokens}, $P_{P\epsilon}$, which ends the mention of a property-name.

Algorithm 2 presents the pseudocode of the search technique POSID, which takes the sentences in a document T_s as input and returns the *collection* of object-properties and their set of values, $\langle\langle o_p, z_p\rangle\rangle$ as output. In case of an implicit mention of clothes, we made an assumption that description of CLOTHES are always followed by descriptions of GENDER, RACE, and/or HEIGHT.

Lines 3 - 5 Extract the candidate sentences with the RESEARCH. If results are empty, extract them with semantic or classification models. Set of key-phrases Q_H is provided by the system.

Lines 8 - 10 Iteratively search for all the finite-valued properties {GENDER, RACE, HEIGHT} in each C_s and append them to output.

`REGEXPROPVAL()` is a regular expression matching function that takes sentence s and property-name o_p as input, and outputs 1) property-value z_p, if o_p is a finite-valued property, or 2) partial sentence s_p, if o_p is a variable-valued property. Each o_p is mapped to a search-string pattern, s_R in T.

Lines 11 - 12 For CLOTHES, `REGEXPROPVAL()` returns either a partial sentence s_p starting with `wearing`, or an empty string. In case of an empty string, keep the remaining string from L_z after discarding the extracted values from lines 8 - 10.

Lines 18 - 19 If first and second token is verb, it is the start for the RELATION property. Following **OIV.6**, ignore consecutive verbs until another tag is encountered.

Lines 20 - 23 Following **OIV.7**, capture tokens from a VERB until any pronoun or non-tag as a free-form property value for CLOTHES.

Lines 24 - 26 Capture the adjectives as clothes descriptions, and initialize the next property.

Lines 28-30 For noun descriptors in the value i.e., grey **dress** pants, compare the wordnet-synset meaning for `color` ($COLOR_{syn}$) to the noun-token meaning. Since a description is encountered, name-index is re-initialized for the next property-name.

Lines 31-32 If a noun-phrase is not a color, it is considered as cloth-name with multiple tokens i.e., *dress pants, tank top,*

Algorithm 2 POSID

Input: Collection of Sentences, T_s
Output: Collection of $\langle name, values \rangle$ pairs, $\langle\langle \mathbf{o}_p, z_p \rangle\rangle$, $\mathcal{O}_H$

1: $fo_p \leftarrow$ {GENDER, RACE, HEIGHT}
2: $COLOR_{syn} \leftarrow$ SYNSETS("COLOR", *NOUN*)[0]
3: $C_s \leftarrow$ EXTRACT-$Cs_{RE}(T_s, Q_H)$
4: **if** C_s is ϕ **then**
5: $\quad C_s \leftarrow$ EXTRACT-$Cs_{model}(T_s, Q_H)$
6: $\mathcal{O}_H \longleftarrow \emptyset$ ▷ Collection of $\langle o_p, z_p \rangle \equiv \langle name, values \rangle$ pairs
7: **foreach** s in C_s **do**
8: $\quad$ **foreach** o in fo_p **do**
9: $\quad\quad L_z =$ REGEXPROPVAL$(s, \mathbf{o}_p)$
10: $\quad\quad \mathcal{O}_H$.APPEND($\mathbf{o}_p$, L_z)
11: $\quad s_p =$ REGEXPROPVAL(s, CLOTHES)
12: $\quad$ **if** s_p is ϕ **then** $s_p \leftarrow s \setminus L_z$
13: $\quad N_{idx} \longleftarrow \emptyset$ ▷ Index-List for property-name
14: $\quad D \leftarrow \emptyset$ ▷ List for property-values
15: $\quad T_o \leftarrow$ TOKENIZE-WORD(s_p) ▷ List of tokens from s_p
16: $\quad T_a \leftarrow$ POS(T_o) ▷ List of ⟨token, POS-tag⟩ from tokens
▷ w_i and t_i is token and POS-tag at i^{th} index in T_a
17: $\quad$ **for** (w, t) in T_a **do**
18: $\quad\quad$ **if** t_1 is *VBD* **then continue**
19: $\quad\quad$ **if** t_2 is *VBG* and t_1 is *VBD* **then continue**
20: $\quad\quad$ **if** $t_i \in P_{DCP} \cup P_{PAV}$ **then**
21: $\quad\quad\quad$ **if** $t_i \in P_{PAV}$ and $t_{i-1} \in P_{P\epsilon}$ **then**
22: $\quad\quad\quad\quad$ **break**
23: $\quad\quad\quad N_{idx}$.APPEND(i)
24: $\quad\quad$ **else if** t_i is *ADJ* **then**
25: $\quad\quad\quad N_{idx} \longleftarrow \emptyset$ ▷ re-initialize name index-list
26: $\quad\quad\quad D$.APPEND(w_i)
27: $\quad\quad$ **else if** t_i is *NOUN* **then**
28: $\quad\quad\quad S_w \leftarrow$ SYNSETS(w_i, *NOUN*)
29: $\quad\quad\quad N_{idx}, D, d_{color} =$ MATCHCOLOR(S_w, N_{idx}, D)
30: $\quad\quad\quad$ **if** d_{color} **then continue**
31: $\quad\quad\quad N \leftarrow w_i$
32: $\quad\quad\quad$ N$\leftarrow$ PROPNAME(N_{idx}, N, T_a)
▷ finalize property-name & assign the values
33: $\quad\quad\quad$ **if** t_{i-1} is *NOUN* and $\mathcal{O}_H$[-1].name == w_{i-1} **then**
34: $\quad\quad\quad\quad$ CONCAT($\mathcal{O}_H$[-1].name, w_i, " ")
35: $\quad\quad\quad$ **else**
36: $\quad\quad\quad\quad \mathcal{O}_H$.APPEND($[N, D]$)
37: $\quad\quad\quad N_{idx} \longleftarrow \emptyset$, $D \leftarrow \emptyset$ ▷ re-initialize Lists
38: $\quad\quad$ **else**
39: $\quad\quad\quad$ **break**

dark clothing. Populate the property-name by backtracking the name-index list.

Line 36 If the previous token is NOUN and does not match the last token of the previous property-name, we consider the end of the current property description. Finalize the current property name and value by appending it to the result. Otherwise, in line 34, amend the last inserted property-name by appending the current token to it.

b) Generalization: Algorithm 2 assumes that the property identifier is intended for human-properties. POSID can be generalized to any object-properties in the text as long as the property names and type of values are known. The search string for fixed-valued properties has to be re-designed. Variable-valued properties following some degree of grammatical structure would be covered by the iterative search pattern in POSID. COLOR will be replaced by the phrase that describes the properties in the corresponding system. Q_H's are highly non-restrictive phrases and can be constructed from entity types or entity names.

V. EXPERIMENTS AND RESULTS

*a) **Dataset Construction**:* For our problem domain, we needed a dataset that did not have similarity or relevance labels but was compatible with existing property identifiers in the literature. During our collaboration with the local police department for the missing person search, we were provided with incident reports and pedestrian videos from the traffic cam. For benchmarking our proposed retrieval methods, we searched for a similar dataset with gold property annotations. We adopted the **MARS** (Motion Analysis and Re-identification Set) person re-identification dataset from [6] for the visual modalities. MARS consists of 20,478 tracklets from 1,261 people captured by six cameras. There are 16 properties that are labeled for each tracklet, among which we used - GENDER (MALE, FEMALE), 9 BOTTOM-WEAR COLORS, and 10 TOP-WEAR COLORS. For property identifiers in textual modalities, we build a collection of text data, named **InciText** dataset from newspaper articles, incident reports, press releases, and officer narratives collected from the police department. We scraped local university newspaper articles to search for articles with keywords i.e., *investigation, suspect, 'person of interest'* and *'tip line phone number'*. InciText provides ground-truth annotations for 12 properties describing human attributes with the most common being – GENDER, RACE, HEIGHT, CLOTHES and CLOTH DESCRIPTIONS (COLORS). Each report, narrative, and press release describes zero, one, or more persons.

Using the above-mentioned datasets, we built a novel retrieval dataset, **MuQNOL**. MuQNOL does not rely on explicit similarity labels and based on the gold property annotations, we can identify the relevant data objects for the user. User can specify which properties are important to them and we can tweak the 'relevance' label based on that. The composition statistics for each modality are:

1) Image (3270/1100/1144),
2) Text (296/178/145), and
3) Video (1454/499/539)

where (*/*/*) stands for the sizes of training/validation/test subsets.

For developing the ground truth, we ranked the data samples in ascending order of the mismatched properties. The properties were chosen depending on the user requirement, and the

Models	Attr-Only			Attr-Value			θ_H	q_H
	Precision	Recall	F1-Score	Precision	Recall	F1-Score		
Word2Vec + POSID	0.83	0.38	0.52	0.85	0.35	0.49	0.5	clothes
RE + POSID	0.86	0.82	0.84	0.92	0.82	0.87	X	wear
WordNet + POSID	**0.93**	0.33	0.49	0.89	0.30	0.45	0.9	*clothes* as noun
SBERT + POSID	0.83	0.49	0.62	0.86	0.45	0.59	0.85	clothes
RE + WordNet + POSID	**0.93**	0.65	0.77	**0.92**	**0.87**	**0.90**	0.9	*clothes* as noun
RE + SBERT + POSID	0.87	**0.87**	**0.87**	**0.92**	**0.87**	**0.90**	0.85	clothes

Fig. 2: Performance of Different Candidate Sentence Extraction Models based on Clothes Property Identification

Properties	CNN (Resnet50)		3D-CNN		CNN-RNN		Temporal Pooling		Temporal Attention		Color Sampling	
	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1
top-color	75.22	73.98	67.91	65.19	70.54	67.33	74.98	73.13	76.05	74.64	44.65	38.31
bottom-color	73.55	54.09	59.77	36.56	67.71	44.44	71.69	47.84	70.15	46.89	45.26	15.88
gender	90.01	89.71	86.49	76.22	90.07	89.62	91.04	90.63	91.82	91.48	-	-
average	**79.59**	**72.59**	67.97	59.18	76.11	67.13	79.24	70.53	79.34	71.01	44.96	27.10

Fig. 3: Comparison of Property Identifiers for Videos with Accuracy (acc) and F1 measure on MARS dataset (%)

mismatches were assigned different penalties. In Example 1, an officer searches in the following order: (1) same gender and race, (2) same bottom clothing, and (3) same top clothing. The intuition behind this is if there is a gender mismatch, they are definitely not the same person. It is possible for a person to change the top clothing in a short span of time, but it is harder to change the bottom clothing. So even if there is a mismatch in top color, there is a chance of it being the same person given a similar time span and vicinity. Therefore, we set the penalty for each mismatched property as follows: $rcost$(TOP-COLOR) = 1, $rcost$(BOTTOM-COLOR) = 2, and $rcost$(GENDER) = 3, with gender having the highest penalty and hence, the highest importance. Exact matches are the top most in the ranking with a zero penalty.

*b) **Settings**:* For Word2Vec, we used the 300 dimensional pretrained model from NLTK [30] trained on Google News Dataset[1]. We pruned the model to include the most common words (44K words). From NLTK, we used the built-in tokenizers and the Wordnet package for retrieving the synsets and wu-palmer similarity score. For SBERT implementation, we used the zero-shot classification pipeline[2] from transformers package using the SBERT model fine-tuned on Multi-NLI [31] task. For part-of-speech tagging, we used the averaged perceptron[3] tagger model. *The manual narratives in the InciText dataset were excluded for property identification task.* Query phrases used for C_s identification are:
q_H = { `clothes, wear, shirts, pants` }.
We follow the original train/test partition of the MARS [6] dataset for benchmarking the property identifiers in visual modalities. For models in [32], [33], [34], we formed a training batch by randomly selecting 32 tracklets, and then by randomly sampling 6 frames from each tracklet. During testing, F frames of each tracklet are randomly split into $\lfloor \frac{F}{n} \rfloor$ groups, and the final result is the average prediction result among these groups. We used a validation set of mutually exclusive 125 people selected from the training set. For color sampling, we used the result from the first frame from each tracklet. We compared three properties across all models - GENDER, TOP COLOR and BOTTOM COLOR. For the retrieval model, we only considered the synthetically generated part of InciText. For Munkres, we used the clapper[4] API. We did not use the local node-node interaction information from simgnn during the training phase for FemmIR.

Attributes	**Gender**	**Race**	**Height**	**Clothes** Attr-only	**Clothes** Attr-value
Precision	0.94	0.94	0.72	0.87	0.92
Recall	0.73	0.73	0.57	0.87	0.87
F1-Score	0.82	0.82	0.63	0.87	0.90

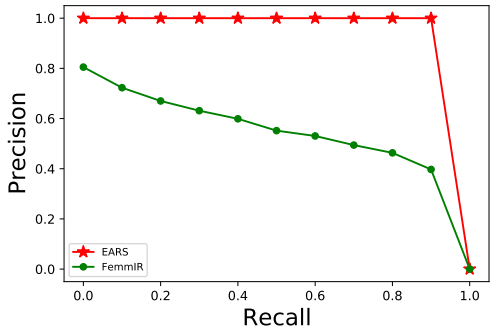
TABLE I: Human Attribute Extraction Results

*c) **Property Identification in InciText dataset**:* We compared the baseline RE-model with the other approaches in Section IV-A for finding C_s. Two different set of metrics were used for the evaluation of CLOTHES identification. (**Attr-only**) evaluates how efficiently the model identified all clothes, and (**Attr-value**) calculates the performance of the model in identifying both the attribute and its descriptive values. For Attr-value, a true positive occurs only when a valid *clothes* name and a correct description of that cloth is discovered. Figure 2 describes the performance of different candidate sentence extraction models based on the performance of CLOTHES

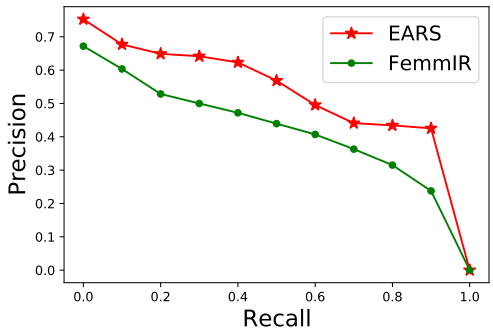
[1] GoogleNews-vectors-negative300
[2] zero-shot-classification
[3] https://www.nltk.org/_modules/nltk/tag/perceptron.html
[4] https://software.clapper.org/munkres/api/index.html

Query	Target	EARS	FemmIR	FGCross-Net
Image	Text	0.54	0.40	0.12
	Image	0.27	0.27	0.11
	Video	0.33	0.29	0.12
	All	0.30	0.28	0.11
Text	Text	1.0	0.52	0.23
	Image	0.37	0.29	0.08
	Video	0.46	0.33	0.07
	All	0.43	0.31	0.10
Video	Text	0.62	0.43	0.09
	Image	0.30	0.29	0.11
	Video	0.37	0.30	0.31
	All	0.34	0.30	0.15
	Avg	*0.44*	**0.33**	0.13

TABLE II: MAP Performance of FemmIR on MuQNOL



(a) Text → Text



(b) Image → Text, Image

Fig. 4: (a) Precision-recall curves for the text as query and data sample modality, (c) Precision-recall curves for the image as query modality with text and image as data sample modality.

identification. For the baseline, the group of tokens around `wear` returned three times better F1-score than any other q_H. With the other models, q_H = {`clothes`} produced the best score. (RE + SBERT) stacked model performs best with 87% and 90% F1-Scores, for both metrics. Although (RE + Wordnet) has a higher precision score of 93% for Attr-only, it has a low recall score of only 65%, indicating over-fitting. Based on a property-frequency analysis, we showed the identification results for the most frequent subset of properties in $\mathcal{O}_H$ for InciText. Table I shows the performance of POSID with (RE+SBERT) for stacked model (lines 3 - 5). For gender and race, the model showed the efficacy of the chosen search-pattern with 94% precision score. A recall score of 73% shows that most people follow similar style for describing gender and race. For *height* with only 57% recall score, a rule based model is not sufficient due to varied styling.

d) ***Property Identifiers in Visual Modalities****:* Since MARS has a large amount of ground truths for person attributes, we compared existing models from the *Person Re-Identification* task. From the CNN models, we used the image-based Resnet50 [35] as the baseline. Due to the temporal nature of videos, we also compared the 3D-CNN [34], CNN-RNN [33], Temporal Pooling and Temporal Attention [32] models. As a heuristic-based model, we chose the color-sampling model from [20]. Table 3 describes the benchmarking results for the compared models. Resnet50 performed significantly better than other models for bottom-color, while temporal attention worked best for top-color. Considering the average performance on all attributes, we choose the *image-based CNN model* for the retrieval task. Since the properties in our task are all motion-irrelevant, the video-based extraction models do not have a large impact on the performance. In terms of training data and time, color sampling surely has an advantage. Resnet50 needed 513 minutes and the temporal attention model needed 1073 minutes for training, whereas color sampling has zero training time. Color sampling works by isolating body regions and evaluating pixel values, hence the presence of sunlight or clouds may have adversely affected the performance.

e) ***Retrieval Performance of FemmIR****:* We compared FemmIR with EARS [18] and FGCross-Net [36]. SDML [37] has shown superior performance to other state-of-the-art cross-modal retrieval models such as ml-CCA [38], ACMR [39], GSS-SL [10], CMPM+CMPC [40]. Since all of them rely on class labels and our problem setting does not allow fine-tuning, we could only compare the models that provided pre-trained models. SDML does not have any publicly available pre-trained model. EARS is an exact inference model and serves best as a baseline model. Since EARS does not require any training, we only used the test set in MuQNOL. We formulated the JOIN queries in the EARS method on the aforementioned properties. The results were a union among the exact match and partial matches for the individual properties. We benchmarked EARS with the extracted property values whereas the gold property annotations were used to create the annotations for multimodal retrieval results.

For evaluation, we considered cross-modal retrieval tasks as retrieving one modality by querying from another modality, such as retrieving text by video query (Video → Text) or, retrieving image by text query (Text → Image). We also show the comparison for multi-modality retrieval. By submitting a query example of any media type, the results of all media types will be retrieved such as (Image → All, Text → All). We adopt mean average precision (mAP) as the evaluation metric, which is calculated on all returned results for a comprehensive evaluation. We consider data samples with $CED < 3$ in

comparison to the query object, as ***relevant*** for that query. This would return contents where persons only with color mismatches are found.

With an average F1 score of 79.59% for video and image property identifiers, the mAP scores of image and video queries are 27%-37%. Text modalities with their high-performance identifiers get the highest mAP across modalities. This indicates FemmIR's correlation with the property identifier performance. If user have a capable property identifier, FemmIR performance will increase. Precision-recall (PR) curves in Figure 4a and 4b show that at lower degrees FemmIR perform comparably with EARS, but with higher degrees of recall, the performance degrades. FemmIR performs significantly better than FGCross-Net without fine-tuning and shows the difficulty of this task.

VI. Related Works

Metric Learning. [41], [42] uses hinge rank loss to minimize intraclass variation while maximizing interclass variation. [15] minimized the loss function using hard negatives with a variant triplet sampling, but needs fine-tuning and augmented data. [16] uses an additional regularization in the loss function with adversarial learning. [17] enables different weighting on positive and negative pairs with a polynomial loss function. FemmIR has similarities to metric learning with the objective of minimizing the edit distance between two graphs. In contrast, FemmIR re-uses pre-extracted properties and does not require data samples to create positive-negative pairs.

Weakly Supervised Learning. [5], [3] use weak signals from entity and relationship similarities retrieved from video captions and text. [18] assumes knowledge of the translation module which makes it less adaptable to novel modalities. [4] uses a similarity-based retrieval technique to extract images with similar subsurface structures. FemmIR also uses a weak signal approach for ranking relevant samples from multiple modalities, but the weak labels are constrained to use the pre-extracted properties and must implicitly maintain the structure between the entities and relationships.

Semantic Understanding with Encoding Networks. [43], [44], [45], [46] learns semantically enriched representations of multi-modal instances by using global and local attention networks. Similarly, FemmIR uses graph convolutional network [22] to align the most important nodes contributing to the overall similarity, denoting the most similar properties between samples.

Content-based Data Discovery. [47], [18], [48], [49], [50], [51] implement content-based data retrieval by taking user-provided example records as input and returning relevant records that satisfy the user intent. Our work shares similarities to DICE [47], which finds relevant results by finding join paths across tables within the data source. However, it focuses on discovering relevant SQL queries from user examples, whereas FemmIR focuses on finding the relevant content directly by finding similar object properties. EARS [18] finds relevant data by applying JOIN queries on the user-required properties from different modalities. Similar to EARS, we also assume the knowledge of pre-identified properties. EARS can scale to petabytes of data, but it needs additional queries to retrieve soft similarities. The number of SQL queries increases proportionally to the number of properties in the user query. Contrary to EARS, we do not assume a common schema for all modalities and do not require re-training from scratch to accommodate new modalities.

Although human attribute recognition from videos and images has been well studied, we believe this is the first work that focuses on finding them from the text. [52], [53] used sentence encoders and dense neural networks to combine lexical and semantic features for finding similar sentences in electronic medical records and academic writing.

VII. Conclusion and Future Directions

We study information retrieval with multimodal (vision and language) queries in real-world applications, which, compared with existing retrieval tasks is more challenging and under-explored. We introduced the problem of mismatch between the information need and encoder features, along with the lack of annotated data for multi-modal relevance. To this end, we presented FemmIR, a framework that uses weak supervision from a novel distance metric for data objects and uses explicitly mentioned information needs with existing system-identified properties. Extensive evaluations on MuQNOL dataset demonstrate that FemmIR exhibits strong performance amongst the retrieval models that require fine-tuning and identifies the relevant data to the user example without supervised training and additional resources. As a byproduct, we also demonstrated the efficacy of HART, a human attribute recognition model from unstructured text, outperforming the baseline language models. FemmIR has successfully implemented a *missing person* use case and is being updated to provide further assistance to local agencies in social causes. In the future, we plan to extend FemmIR to include multi-objective and evolving information needs to support more real-world use cases.

Acknowledgment

This research is supported by the Northup Grumman Mission Systems' Research in Applications for Learning Machines (REALM) Program.

References

[1] M. Luo, Z. Fang, T. Gokhale, Y. Yang, and C. Baral, "End-to-end knowledge retrieval with multi-modal queries," *arXiv preprint arXiv:2306.00424*, 2023.

[2] Z. Li, J. Tang, L. Zhang, and J. Yang, "Weakly-supervised semantic guided hashing for social image retrieval," *International Journal of Computer Vision*, vol. 128, no. 8, pp. 2265–2278, 2020.

[3] N. C. Mithun, S. Paul, and A. K. Roy-Chowdhury, "Weakly supervised video moment retrieval from text queries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 592–11 601.

[4] Y. Alaudah, M. Alfarraj, and G. AlRegib, "Structure label prediction using similarity-based retrieval and weakly supervised label mappingstructure label prediction," *Geophysics*, vol. 84, no. 1, pp. V67–V79, 2019.

[5] K. Solaiman and B. Bhargava, "Open-learning framework for multi-modal information retrieval with weakly supervised joint embedding," 2022.

[6] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *European Conference on Computer Vision*. Springer, 2016, pp. 868–884.

[7] K. Solaiman and B. Bhargava, "Feature centric multi-modal information retrieval in open world environment (femmir)," 2023.

[8] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 251–260.

[9] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," in *Conference on Data Mining and Data Warehouses (SiKDD 2010)*, 2010, pp. 1–4.

[10] L. Zhang, B. Ma, G. Li, Q. Huang, and Q. Tian, "Generalized semi-supervised and structured subspace learning for cross-modal retrieval," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 128–141, 2017.

[11] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *International conference on machine learning*. PMLR, 2015, pp. 1083–1092.

[12] M. Kan, S. Shan, and X. Chen, "Multi-view deep network for cross-view classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4847–4855.

[13] Y. Peng, J. Qi, X. Huang, and Y. Yuan, "Ccl: Cross-modal correlation learning with multigrained fusion by hierarchical network," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 405–420, 2017.

[14] Y. Peng, X. Huang, and J. Qi, "Cross-media shared representation by hierarchical learning with multiple deep networks," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, p. 3846–3853.

[15] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," *arXiv preprint arXiv:1707.05612*, 2017.

[16] X. Xu, L. He, H. Lu, L. Gao, and Y. Ji, "Deep adversarial metric learning for cross-modal retrieval," *World Wide Web*, vol. 22, no. 2, pp. 657–672, 2019.

[17] J. Wei, X. Xu, Y. Yang, Y. Ji, Z. Wang, and H. T. Shen, "Universal weighting metric learning for cross-modal matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 005–13 014.

[18] K. Solaiman, T. Sun, A. Nesen, B. Bhargava, and M. Stonebraker, "Applying machine learning and data fusion to the "missing person" problem," *Computer*, vol. 55, no. 06, pp. 40–55, jun 2022.

[19] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[20] M. Stonebraker, B. Bhargava, M. Cafarella, Z. Collins, J. McClellan, A. Sipser, T. Sun, A. Nesen, K. Solaiman, G. Mani, K. Kochpatcharin, P. Angin, and J. MacDonald, "Surveillance video querying with a human-in-the-loop," in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics with SIGMOD*, 2020.

[21] K. Riesen, M. Neuhaus, and H. Bunke, "Bipartite graph matching for computing the edit distance of graphs," in *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer, 2007, pp. 1–12.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs / 1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

[23] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, and W. Wang, "Graph edit distance computation via graph neural networks," *arXiv preprint arXiv:1808.05689*, 2018.

[24] R. J. Qureshi, J.-Y. Ramel, and H. Cardot, "Graph based shapes representation and recognition," in *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer, 2007, pp. 49–60.

[25] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 926–934. [Online]. Available: http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion.pdf

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[27] Z. Wu and M. Palmer, "Verb semantics and lexical selection," *arXiv preprint cmp-lg/9406033*, 1994.

[28] W. Yin, J. Hay, and D. Roth, "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach," *arXiv preprint arXiv:1909.00161*, 2019.

[29] N. s Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019.

[30] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ser. ETMTNLP '02. USA: Association for Computational Linguistics, 2002, p. 63–70. [Online]. Available: https://doi.org/10.3115/1118108.1118117

[31] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1112–1122. [Online]. Available: https://www.aclweb.org/anthology/N18-1101

[32] Z. Chen, A. Li, and Y. Wang, "Video-based pedestrian attribute recognition," *CoRR*, vol. abs/1901.05742, 2019. [Online]. Available: http://arxiv.org/abs/1901.05742

[33] N. McLaughlin, J. M. Del Rincon, and P. Miller, "Recurrent convolutional network for video-based person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1325–1334.

[34] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] X. He, Y. Peng, and L. Xie, "A new benchmark and approach for fine-grained cross-media retrieval," in *Proceedings of the 27th ACM international conference on multimedia*, 2019, pp. 1740–1748.

[37] P. Hu, L. Zhen, D. Peng, and P. Liu, "Scalable deep multimodal learning for cross-modal retrieval," in *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR'19. New York, NY, USA: ACM, 2019, pp. 635–644. [Online]. Available: http://doi.acm.org/10.1145/3331184.3331213

[38] V. Ranjan, N. Rasiwasia, and C. Jawahar, "Multi-label cross-modal retrieval," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4094–4102.

[39] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 154–162.

[40] Y. Zhang and H. Lu, "Deep cross-modal projection learning for image-text matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 686–701.

[41] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, "Deep coupled metric learning for cross-modal matching," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1234–1244, 2016.

[42] A. Frome, G. Corrado, J. Shlens *et al.*, "Devise: A deep visual-semantic embedding model," in *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[43] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 201–216.

[44] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4654–4662.

[45] Y. Song and M. Soleymani, "Polysemous visual-semantic embedding for cross-modal retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1979–1988.

[46] S. Sah, S. Gopalakrishnan, and R. Ptucha, "Aligned attention for common multimodal embeddings," *Journal of Electronic Imaging*, vol. 29, pp. 023 013 – 023 013, 2020.

[47] E. K. Rezig, A. Bhandari, A. Fariha, B. Price, A. Vanterpool, V. Gadepally, and M. Stonebraker, "Dice: Data discovery by example," *Proc. VLDB Endow.*, vol. 14, no. 12, p. 2819–2822, jul 2021. [Online]. Available: https://doi.org/10.14778/3476311.3476353

[48] R. Gasser, L. Rossetto, and H. Schuldt, "Multimodal multimedia retrieval with vitrivr," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ser. ICMR '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 391–394. [Online]. Available: https://doi.org/10.1145/3323873.3326921

[49] S. M. Sarwar and J. Allan, "Query by example for cross-lingual event retrieval," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1601–1604. [Online]. Available: https://doi.org/10.1145/3397271.3401283

[50] S. Palacios, K. Solaiman, P. Angin, A. Nesen, B. Bhargava, Z. Collins, A. Sipser, M. Stonebraker, and J. Macdonald, "Wip - skod: A framework for situational knowledge on demand," in *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, V. Gadepally, T. Mattson, M. Stonebraker, F. Wang, G. Luo, Y. Laing, and A. Dubovitskaya, Eds. Cham: Springer International Publishing, 2019, pp. 154–166.

[51] M. Lazaridis, A. Axenopoulos, D. Rafailidis, and P. Daras, "Multimedia search and retrieval using multimodal annotation propagation and indexing techniques," *Signal Processing: Image Communication*, vol. 28, no. 4, pp. 351 – 367, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596512000756

[52] Q. Chen, J. Du, S. Kim, W. J. Wilbur, and Z. Lu, "Combining rich features and deep learning for finding similar sentences in electronic medical records," *Proceedings of the BioCreative/OHNLP Challenge*, pp. 5–8, 2018.

[53] C. L. GOH and Y. LEPAGE, "Finding similar examples for aiding academic writing using sentence embeddings," 2020.