Multi-modal Information Retrieval for Systems with Explicit Information Needs and Object Properties (FemmIR)

KMA Solaiman
ksolaima@purdue.edu
Purdue University
West Lafayette, IN 47906, USA

Bharat Bhargava
bbshail@purdue.edu
Purdue University
West Lafayette, IN 47906, USA

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. The heterogeneity gap between explicitly mentioned properties in the information need and low-level representation features used in the retrieval models makes them unusable in certain systems. Our goal is to avoid the annotation overhead incurred from considering retrieval as a supervised classification task, and re-use the pre-existing properties in the system. We propose FemmIR, a framework to retrieve multi-media results relevant to information needs expressed with data examples from various modalities. Such identification is necessary for real-world applications where computational resources are scarce and rapid turnaround is required. 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 as a property identifier. We empirically evaluate FemmIR and HART on a missing person use-case with a combination of a real and synthetic dataset. HART successfully identifies human attributes from large unstructured text without additional training. FemmIR performs comparably to similar systems in delivering on-demand retrieval results with exact and approximate similarities while using the existing properties in the system.

CCS CONCEPTS

• **Information systems → Multimedia and multimodal retrieval**; *Retrieval effectiveness*; *Content analysis and feature selection.*

KEYWORDS

multi-media datasets, multi-modal information retrieval, text attribute detection, human attribute detection, property identification, data discovery

1 INTRODUCTION

With the influx of media collections, exploratory data analysis requires comparing data from different modalities to grasp a more informed decision for any phenomenon. With the ever-growing size of the multi-media data, multi-modal data analysis becomes difficult for any real-world application-specific information needs, specially when there is heterogeneity among data properties in different modalities and information needs. Current data discovery systems rely on manual lookup, exploration of the relational database structure, or cross-modal information retrieval for the data preparation task. Traditional cross-modal retrieval models create a common representation space to compare the similarities between data sources, whereas relational query models find the user intentions from the query history and deliver the data tuples that match the user preference model. However, having a common subspace to translate all incoming data, or designing new queries for every new modality causes a system with existing properties to fail. So we ask the question, *how can we handle data retrieval in applications with existing object properties and explicit information needs?*

Two common reasons retrieval systems with application-specific information needs fail are: (1) *disconnect* between high-level information needs and low-level object properties, and (2) lack of annotated data compared to the size of the multimedia data. Object properties are often specified to the system as high-level information needs, whereas most retrieval systems use representation models to gather low-level features from different modalities before mapping them into the common subspace to compare them. Multi-modal systems that can process high-level information need described as properties [39, 61] often have to handle retrieval from a large repository, or streaming data. [61] expects to process 60,000 frames per minute from the camera feeds [64], whereas on average 6,000 tweets are generated per second in [61]. Most retrieval systems cannot process data ingestion for these large amounts of data, and annotating them for training to discriminate between relevant and irrelevant, is a near-impossible task.

Therefore, in case of application-specific information needs, one should explore a retrieval system that would use existing object properties in the data. Examples of commonly observed retrieval system failures caused by explicit information need include: (1) decline of model accuracy (due to using the same property identifiers for all systems, and lack of generalization), (2) waste of computational resources (due to re-running identifiers), (3) excessive processing time (due to failure to scale to large data), (4) inability to incorporate new modalities (due to incoherent representation and redundant properties), and (5) system crash (due to invalid modalities beyond design). These examples indicate a common problem: *mismatch* between properties in the information need and in the data, as well as *dependence on annotations*. To motivate our work, we start with an example inspired by a real-world system, where local law enforcement officers asked for assistance to sift through hours of videos [64].

Example 1.1 (Object-property focused Information Need). An agency wants to build an automated system to find persons of interest from many hours of video feeds. Incident reports and text queries were considered to be text modalities. Alex is asked to develop a machine learning (ML) pipeline over this dataset 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 multi-media data. But the performance was not satisfactory. Alex was not able to modify the model to focus on specific properties which are most common for a missing person. On the other hand, he could not run 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 features? (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 from each modality?

Existing tools [19, 63] that use encoder-decoder architecture or metric learning to map low-level features to a common space cannot explicitly consider high-level properties. Also, the system required a soft-match approach rather than an exact match since finding persons-of-interest is a sensitive use case, and although organizations want to ease their workload, they do not want to commit any mistakes. Example 1.1 is one among many incidents in real-world applications where similarities among multi-media sources are required with a focus on specific object properties [citations]. As mentioned in prior work [13], *"21% of the bugs encountered in Microsoft Azure services were due to inconsistent assumptions about data format [32]. Furthermore, 83% of the data-format bugs were due to inconsistencies between data producers and consumers, while 17% were due to mismatches between different consumer interpretations of the same data. Similar incidents happened due to misspelling and incorrect date-time format [48], and issues pertaining to data fusion where schema assumptions break for a new data source [9, 66]."* Hence a retrieval system must handle the inconsistent assumptions about object properties from different data sources. We provide another example where a system fails with the introduction of novel modalities or data sources.

Example 1.2 (Mismatched Properties across Data Sources). As pointed out as an example in prior work [47], "An organization within the Air Force collects data from sensors to support data scientists in producing data-driven reports for decision-makers. This vast and heterogeneous data is organized across hundreds of tables in a data lake, each with a different schema." DICE [47] helps in finding the right data sources by finding join paths across tables and involving human-in-the-loop. But as new tables from different sensors are added, there is no guarantee the previous joins will hold.

The aforementioned examples bring forth three key challenges. First, we need to find a *common representation model* for object properties from all modalities and *map* them into a common embedding space for the downstream similarity matching task. Second, the similarity of the data samples needs to be *measured* in a manner that captures the *approximate matches*. For example, two records can be similar if they have the "same entities", and/or they describe the "same event". Third, for the retrieval model, we need to find a *training* method in absence of annotated data.

Common representation modeling with Graphs. Towards solving the first challenge, our observation is that most real-world data describes relational knowledge among different entities, along with their attributes and metadata such as spatial and temporal data. *Graph representation* allows these structural and characteristic information to be stored and accessed efficiently [3], across multiple modalities. Besides, deep learning-based dynamic graph embedding methods [5, 22] learn low dimensional vector representations for the graph while preserving both the graph properties and the structure. Despite different naming and organization conventions across various data sources, each data-sample still holds the relationship properties among entities. This eliminates the issue of heterogeneous property representations. For example, *social network recommendation engines* (such as, Yelp [57], or Pinterest [16]), or *multi-media recommendation applications* (such as, micro-video recommendation in tiktok, Kwai, or MovieLens [69]) often use graph representations.

Tensor-based similarity comparison. Our second observation is that real-world information need often emphasizes implicit matching (retrieval matching, entity-relation matching, or user-item matching) [46] rather than just explicit matching. Since tensor is a geometric object that describes relations between vectors and prior works [58] have shown that neural tensor network (NTN) can explicitly model multiple interactions of relational data, we choose to use an NTN-based framework to measure the similarity between graph representations. The optimal number of interaction scores is application-specific. For example, *an application looking for person-of-interests wants to soft-match a person with similar race, gender, and clothes, whereas a missing person search would require to match all human attributes for an exact match.* The model would learn that the multi-media samples from different modalities describing the same entity would be in similar parts of the semantic space.

Weakly supervised learning for retrieval. Our third and final observation is, capturing how much change is needed to convert one data-sample to another can provide us with a source of weak supervision for cross-modal retrieval. We consider a data sample as a collection of objects with certain properties along with the relationships among them. Since graph edit distance (GED) has been shown to be an effective graph distance metric in many applications, such as graph similarity search [30, 79], graph classification [49, 50], image indexing [72], etc., we modeled a data-sample similarity metric based on GED. Since multi-media data is usually large in number, it is expensive to annotate the relevance for each different system. Prior works in retrieval system [1, 29, 36, 60] use inexact weak supervision to tackle this issue of annotation expense.

Solution Sketch. We propose FemmIR, a framework that compares data from different modalities and heterogeneous sources to user-provided information need and calculates a similarity score among them. Our framework involves three main components:

(1) *Data Ingestion:* a graph encoding mechanism that translates properties from incoming data into an attributed graph representation. In general, property names are considered as edge labels, whereas values are used as node labels.
(2) *Weak Label Generation:* For capturing the similarity between a pair of data samples, we define a new distance metric that

indirectly holds the entity and relationship constraints between the samples, Content Edit Distance (*CED*). CED captures the amount of change needed to convert one attributed data graph to another by including the object replacement cost for cost matrix calculation in Munkres algorithm [51]. CED is later used to define **relevance** label based on system requirements.

(3) *Similarity Comparison:* Finally, we train a lean-able embedding function for multiplicative comparison between attributed graphs using the SimGNN architecture [4]. During the training, the objective function minimizes the difference between the predicted score and the ground truth obtained from converting the *CED* into a similarity score. During inference, the learned embedding function calculates the similarity score between the attributed graphs from the data-records.

Given a scenario where the user provides information need as an example and incoming streams have identified properties, FemmIR starts with building the attributed graphs. In case of unseen raw data, FemmIR extracts the object properties either offline, or with priority polling [64] for bulk streams in an additional *property identification* component. Second, the CEDs between the graphs are calculated between the query example and the data samples. Finally, the model is trained to calculate the similarity score between records.

Scope of our work. In this work, we only focus on retrieval cases where either (1) properties from different objects and relationships among them have been identified, or (2) the system has specified its own identifiers for specific data modalities. Note that prior data-matching approaches [24, 61] that employ retrieval model on high-level properties assume there exists a common schema or feature mapping among different modalities. In contrast, FemmIR is agnostic to the design of the source-schema and can support any type of property schema from any data source ranging from raw data in data warehouse (Example 1.1) to a relational database in a data lake (Example 1.2). FemmIR also delivers varying degrees of relevance without the computational overhead. However, FemmIR cannot handle multiple query examples at the same time for single information need as it requires predicting user intent from those examples [13] and that is not the focus of this work.

FemmIR requires knowledge of the application-specific property identifiers to be used throughout the system. The choice of property identifiers depends on the domain knowledge and the properties-of-importance, e.g., *in Example 1.1, Alex would require identifiers for video and text which extracts human properties i.e., gender, race, cloths-worn, cloths-colors, etc.* This assumption holds because: (1) for object and action recognition tasks there exists a well-known set of relevant identifiers for common modalities - video [45, 80], text [8], image [25, 75], and 3D models [42] with reasonable performance. Objects and actions cover the most common properties in retrieval applications. (2) If the information need is expressed through high-level properties [14, 56, 61], we can assume the system already extracted properties from most modalities and domain experts are typically aware of the likely class of properties for the specific task at hand and can easily provide this additional knowledge to the system.

Property Identifiers. While we use the property-identifier outputs to find relevant data sources to the query example, developing identifiers is orthogonal to our work. A number of object and action recognition paradigms from different modalities exist in the literature. FemmIR assumes access to a suite of property-identification techniques and uses them to extract properties from the data. To support a new data source, FemmIR needs to know the corresponding modality and the properties-of-interest. We discuss some common classes of property identifiers as representative ones, which are currently supported in the implementation of FemmIR. We have selected them based on the criteria of having semantically similar properties or similar property definitions. For properties discovery in visual modalities, we rely on prior work on action recognition [25, 80], object detection [75, 78], etc. As part of FemmIR, we proposed a novel property identification method for properties described in textual modalities. Properties from the unstructured text are hard to extract because of its multifaceted and individualistic characteristics of it. Traditional natural language processing techniques for entity and relation extraction fail for such entity-specific properties. As a specific example, we proposed *a method for extracting properties describing human attributes in textual modalities.* While our evaluation covers specific property identifiers, FemmIR is generic and works for any class of identifiers, as long as the corresponding properties are available.

Limitations of previous works. Correlation learning methods [21, 40, 41, 44, 53, 65, 77] linearly or non-linearly projects low-level features from representation models to a common subspace. Metric learning methods [10, 68, 73] learn a distance function over data objects based on a loss function to map them into the common subspace. All these models require a large amount of training data and data representations lack a common encoding mechanism. 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 allows more flexibility to consider explicit information need from users, and use high-level properties in the system. EARS [61] 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, 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.

Contributions. In this paper, we make the following contributions:

- We design and develop a novel multi-modal information retrieval approach to find multi-media data relevant to information need expressed as Query-by-Example, or Query-by-Properties. The approach leverages a neural-network based graph-matching technique to capture the interactions between information need and the data properties, with a weak supervision from a novel distance metric for data samples. (Section 3)
- We propose a novel human attribute recognition model from the unstructured text as part of the property identifiers. The

model leverages pattern-matching techniques and contextualized language models while exploiting the syntactic grammatical properties to extract properties describing a person. (Section 4)

- We evaluate FemmIR on a real-world application for Missing Persons, with an unannotated dataset and a property-specific information need. We demonstrate that FemmIR shows similar performance to other retrieval systems [61] while leveraging pre-identified properties, on a novel multi-media dataset comprised of pedestrian identification and real-world dataset.
  Over a combination of real-world and synthetic datasets, we further show the efficacy of the human attribute recognition model. Moreover, we benchmark property identifiers for the visual modalities to identify the best model for the downstream retrieval task. (Section 5)

2 PRELIMINARIES & PROBLEM DEFINITION

In this section, we first provide formal definition to attributed relational graph, wordnet synsets, and natural language inference. We then proceed to formulate the problem of multi-modal information retrieval for property-specific information need, and the problem of 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 values or vectors of values from Σ). For the sake of simplicity, from now on we denote the node and edge attributes by labels, as labels are specific type of attributes. Although we focus our methodology only on directed and labeled graphs, it is designed to handle any forms of graphs. It 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 labelling function that assigns each vertex and/or edge a label from Σ. Specifically, $l(u)$ and $l(u, u')$ are the label of node u and the label of edge (u, u'), respectively, (4) Σ is a finite or infinite set of unconstrained labels. $A \in \Sigma$ represents labels enumerating the node-type.

Definition 2.2 (Wordnet Synsets). Wordnet[11] is a lexical knowledge base where words are organized in a hypernym tree based on their origin. Words are grouped into Synsets based on their synonyms. Wu-Palmer distance calculates the similarity between word meanings based on how similar the word senses are and where the Synsets occur 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:

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

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:

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

2.1 Problem Definition

Considering a collection of data from $\mathcal{M} \in \mathbb{Z}^{+}$ modalities, we denote the j-th sample of the i-th modality as $\mathbf{d}^{i}_{j}$. The set containing all the $n_i \in \mathbb{Z}^{0+}$ samples of the i-th modality is denoted as $\mathcal{D}_i = \{\mathbf{d}^{i}_{1}, \mathbf{d}^{i}_{2}, \ldots, \mathbf{d}^{i}_{n_i}\}$. Each data sample contains a collection of *object-properties*. For example, a document has a *topic, metadata*, and *entities* with their *relationships*, along with any *event* it describes. Let $O = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_l\}$ be the set of all such *object-properties*, where z_r is the set of values of property $\mathbf{o}_r$. A data sample $\mathbf{d}^{i}_{j}$ is described with a subset of O. $O_E \subseteq O$ denotes the set of object-properties describing an entity E. $z_r = \{\phi\}$ indicates that $\mathbf{o}_r$ is not present in $\mathbf{d}^{i}_{j}$. Property identifiers implement a relation, *PROP* ($\mathbf{d}^{i}_{j}$) $\subset O$ that maps a data-sample to a set of object-properties (*PROP*: $\mathcal{D} \rightarrow O$). 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). $Q_{\mathbf{d}^{m \in \mathcal{M}}} = \{\mathbf{o}_1 = z_1, \mathbf{o}_2 = z_2, \ldots, \mathbf{o}_p = z_p\}$ *is a data query that is expressed in one of the two ways:* (**1**) *(Query-by-Properties) with p object-properties mentioning a target data-sample* $\mathbf{d}^{m}_{q}$ *from modality m with PROP* ($\mathbf{d}^{m}_{q}$) = $Q_{\mathbf{d}^{m \in \mathcal{M}}}$, *or* (**2**) *(Query-by-Example) with an example data-sample* ($\mathbf{d}^{m}_{q}$) *of modality m with PROP* ($\mathbf{d}^{m}_{q}$) = $Q_{\mathbf{d}^{m \in \mathcal{M}}}$. *The task is to retrieve a ranked list,* $R = (\mathbf{d}^{x_1}_{1}, \mathbf{d}^{x_2}_{2}, \ldots \mathbf{d}^{x_t}_{t})$ *of* $t \in \mathbb{N}_0$ *data-samples from all available modalities in the system satisfying PROP* ($\mathbf{d}^{m}_{q}$)*, where* $\mathbf{d}^{x_c}_{c}$ *is c-th data in R from modality* $x_c \in_R \mathcal{M}$.

Relevance is scored based on the degree of common object-properties between the data-object $\mathbf{d}^{x_c}_{c}$ in the ranked list, and the query data $\mathbf{d}^{m}_{q}$, *PROP* ($\mathbf{d}^{m}_{q}$) $\cap$ *PROP* ($\mathbf{d}^{x_c}_{c}$). A similarity score is used to define the degree of relevance, $0 \leq SIM(\mathbf{d}^{x_c}_{c}, \mathbf{d}^{m}_{q}) \leq 1$. Similarity score of 0 indicates non-relevance, whereas a score of 1 indicates complete relevance and a proper subset, *PROP* ($\mathbf{d}^{m}_{q}$) $\subset$ *PROP* ($\mathbf{d}^{x_c}_{c}$).

Our problem setting assumes that the user has knowledge about o_p and their corresponding z_p for Query-by-Properties. This assumption is realistic in real-world scenarios and has been considered in multimodal data query literature where properties are used to express the information need [14, 56, 61].

2.1.1 Property Identification from Unstructured Text. Specifically, we explore the problem of identifying properties describing humans from unstructured text. As discussed in SurvQ [64], a finite number of object properties such as, gender, race, build, height, clothes, etc. are used in profiling a person-of-interest to search for them. We denote object-properties used in person profiling as O_H.

Example 2.4. The sentence "`a` **`white male`** `with` **`medium`** `build was seen in Vernon St., wearing` **`white jeans`** `and` **`blue shirt`**" describes object-properties of a †person: (1) gender = `male`, (2) race = `white`, (3) build = `medium`, (4) *clothes = {`jeans`, `shirt`}, (5) upper-wear-color= {`white`}, (6) bottom-wear-color = {`blue`}, and (7) 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* 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

2.4, Q_H = {wearing}. The first assumption is derived from literature in pedestrian attribute recognition from visual and textual modalities, and the second assumption holds as small amount of curated data is always available for a problem setting. Note that, $(Q_H \cap O_H) \neq \{\phi\}$. Candidate sentences are sentences in the text that mentions phrases similar to the key-phrases within an empirical threshold value.

Definition 2.5 (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

$$C_s = \{s : s \in T_s, q_H \in Q_H \mid SIM(q_H, s) > \theta_H\} \tag{2}$$

3 MULTIMODAL INFORMATION RETRIEVAL

We will now describe the multimodal similarity matching method to find the relevant data to user provided information need (mentioned with an example, or with object properties). The matching algorithm considers the data samples, $\mathbf{d}_c^{x_c}$ (from the data repository, or from data streams) 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 available from each data sample extracted by the system-specific property identifiers before the matching algorithm is applied. We propose a weakly supervised approach to rank the data samples by generating a distance metric between them based on the amount of edits (changes) needed to convert the properties of one sample to another instead of manually annotating the number of matched object-properties. To this end, we first process the input data samples with a graph ingestion mechanism which converts the extracted properties into a hierarchical attributed relational graph (HARG). Our weakly supervised strategy, FemmIR adopted the Munkers' algorithm [51] to calculate the edit distance between the data samples. Finally, we used a neural network based edit distance approximation algorithm 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 functions. We start with an example scenario to demonstrate how data ingestion works and then proceed to describe the weakly supervised approach.

3.1 Data Ingestion with Graphs

Consider the task of finding the location of a person from large amount of video data using the text queries or reports (Example 1.1). The system finds the video feeds that has the persons similar to the report description (using multi-modal similarity matching) by focusing on object-properties of persons in the video and text. The goal is to identify the similarity score between video feeds, text queries, and incident reports which can be used to deliver a ranked list of relevant data samples to the user.

Observations. We make the following observations.

O3.1 The number of object-properties that is used to compare between two data samples are finite, and the value of the properties are mostly categorical values. A data sample can describe a large amount of objects and object-properties, but for system-specific similarity comparison an user is only interested in a finite number of properties.

O3.2 Data samples are objects themselves that have different properties such as, metadata, topics, and events that they describe. *Entities* are specific types of objects described in a data sample which has its own properties.

O3.3 *Relationships between objects* are a specific types of object-properties which 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.*

O3.4 Some properties in z_p have single and fixed value-set i.e., GENDER, RACE, HEIGHT, while other properties have a multiple number of values in their value-set i.e., CLOTHES.

O3.5 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.

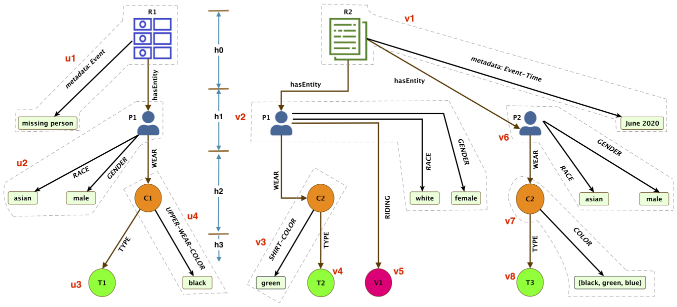


Figure 1: HARG and Weak Label Generation; Left side graph refers to g^q, and the right side graph refers to g^c. Node-type labels are as follows. V: EPL Vertex, R: Root, P: Person, C: Clothes, T: Type, M: Motor-Vehicles. Squared nodes correspond to the non-empty leaf nodes.

Our intuition here is that entities, relationships, and object-properties in a data sample have a inter-connected structure and if we can capture the changes we need to make to this structure to convert it to structure of another data-sample, 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: (1) graph being the best data structure to capture information from connected structures, (2) based on observations **O3.3** and **O3.5**, a data structure with representation-invariant encoding mechanisms that can capture the syntactic similarities between different values was necessary.

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 is denoted by N^h.

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 (CONTENT/ OBJECT/ ROOT) for all data samples in the same application domain: $l(N^0) = \{ROOT\}$.

(2) Level 1 nodes constitute of 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 of that data sample, N^1 would be a leaf node. And 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 object-properties of an entity, $N^k(k \geq 1)$ will be a pointer to the properties of that entity, whereas $l(N^k) = \{entity\text{-}type\}$. We categorize entities in two groups for each data sample: *primary*, and *secondary*. Level 1 of HARG only contains primary entities.

(4) Level 2 and subsequent levels contain the property-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 3.3, for RELATION properties, $\langle R, S, Arg\rangle$ where entity-pointer S is at level-k and entity-pointer Arg is at level-$(k+1)$, $l(N^k, N^{k+1}) = R, l(N^k) = S, l(N^{k+1}) = Arg$.

(5) 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 $k = r$.

(6) The leaf nodes of HARG always contain a property-value or a NULL value for $z_p = \{\phi\}$.

Definition 3.2 (Primary Entities). Primary entities are entities that take the role of a subject in terms of a verb. (1) In visual modalities, entities that control the action or relation properties are considered as primary entities. (2) Entities that satisfy any of the following criteria is considered as *primary entities* in textual modalities: (a) In phase structure grammars, primary entity is an immediate dependent of the root node [52], (b) In dependency grammars, primary entity is an immediate dependent of the finite verb [37]. (3) For database records, we consider the entities from the tables with *no foreign key constraints* as the primary entity. Secondary entities include any entities not satisfying the conditions of primary entities including *objects, verb arguments, and themes*.

Definition 3.3 (RELATION between Objects). Object-properties describing a relationship or action R between two entities S (initiator) and Arg (outcome/ receiver/ modifier) are defined as RELATION properties, and the property-value is defined as a triplet of $\langle R, S, Arg\rangle$. For a n-ary relationship R, identifiers associate each action with multiple entity arguments, $Arg_1, Arg_2, \ldots, Arg_i, \ldots, Arg_n$ with *role* R_o^i. n-ary relationships are broken into multiple binary relationships with $l(N^k, N^{k+1}) = \{R : R_o^i\}, l(N^k) = S, l(N^{k+1}) = \{Arg_i\}$.

Figure 1 demonstrates two example hierarchical attributed relational graphs from the experimental dataset. $R1$ and $R2$ refers 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. We made two assumptions for the generation process: (**I**) We assume prior knowledge of the system-specific properties [ ] and that they have been extracted with appropriate property-identifiers, (**II**) The entity types for node labels are system-specific, and must be consistent through 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.

3.2 Weak Label Generation

FemmIR further defines a new distance metric, Content Edit Distance (CED) using a variation of the Munkres' algorithm [51] 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.

O3.6 Different nodes and edges in HARG have different change cost. User should be allowed to specify individual property replacement cost.

O3.7 GED calculation algorithms differ in speed based on the number on nodes and HARG contains variable sized graphs.

O3.8 Object-properties such as, RELATION has 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.

O3.9 Considering **O3.4**, 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: **(prop-LED)** ordered comparison with modified *Levenshtein distance*, and **(hash-cmp)** 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 3.4) 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:

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

Definition 3.4 (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 their own cost functions.

Munkres Algorithm for CED calculation. We consider the CED calculation as an assignment problem and adopted the bipartite graph matching method in [51]. Compared to the exponential time-complexity of A*-search, Munkres' [51] 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, $g^q_{\text{epl}} = (EPL(V)^q, EPL(E)^q, EPL(l)^q)$ and the HAR graph from the compared data-sample, $g^c_{\text{epl}} = (EPL(V)^c, EPL(E)^c, EPL(l)^c)$, where $EPL(V)^q = \{u_1, \dots, u_n\}, EPL(V)^c = \{v_1, \dots, v_m\}$, the Munkres' algorithm would output CED $(g^q_{\text{epl}}, g^c_{\text{epl}})$. We made the following adjustments to the Munkres' algorithm in [51].

(1) EPL-vertices in the query graph needs to be aligned to the data-samples, hence we will fix the assignment size k to $|EPL(V)^q|$.
(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 **O3.5**, while applying Adjacency-Munkre s, 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.

$$c(e_{u_i} \rightarrow e_{v_j}) = 1/wpdist(s_{l(e_{u_i})},\ s_{l(e_{v_j})}) \tag{3}$$

Cumulative-Munkres. Using Adjacency-Munkres from [51] allows us to find the optimal assignment of each EPL vertex without taking into account the dependency among them **O3.8**. 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.

3.3 Similarity Measurement

Finally, we propose to use an end-to-end neural network model, SimGNN [4] 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 system constraints (such as, different property-values) are applied with appropriate replacement costs while calculating CED. Similarity scores for training the model are derived by transforming the distance scores using the normalization method from [43] and an exponential function on the normalized score. (Line 25 in Algorithm 1). The embedding function outputs a number of interaction scores between a pair of graphs using Neural Tensor Networks (NTN) [59] on the graph embeddings. For calculating the graph embedding, first, Graph Convolutional Networks (GCN) [23] 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, when ground truths are available. It is also inductive and allows to compute the node embedding for any unseen graph following the GCN operation, which makes it a great choice for variable sized FemmIR graphs. Then, an attention network is used to combine the node embeddings into a graph embedding allowing to learn each node's weight in the similarity determination as part of the end-to-end network. In addition, SimGNN augments the graph level interaction score with local information by calculating histogram features from a pairwise node interaction score between the node 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 similarities from the weak-labels or the ground-truths using mean squared error loss.

$$C = \frac{1}{|D|} \sum_{d_c \in D} (\hat{s} - s(d_q, d_c))^2 \tag{4}$$

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.

3.4 FemmIR algorithm

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, 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 3.1.
Lines 4 - 25 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 Discover the EPL-vertices in the HARGs, and define 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 - 15 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.
Lines 16 - 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 3.2. cmp is a control variable to specify what kind of comparison is required. The overall cost is added to cost-matrix.

Algorithm 1: FemmIR

```
Input: Query example and a single Data sample, $d_q$ and $d_c$
Output: Similarity score between $d_q$ and $d_c$, SIM $(d_q, d_c)$
Given: (1) Replacement cost for property $\mathbf{o}_p$, rcost($\mathbf{o}_p$)
       (2) Insertion cost for property $\mathbf{o}_p$, icost($\mathbf{o}_p$)
1  $O^q \leftarrow PROP(d_q)$,        $O^c \leftarrow PROP(d_c)$
2  $g^q \leftarrow CONSTRUCT-HARG\ (O^q)$
3  $g^c \leftarrow CONSTRUCT-HARG\ (O^c)$
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 $C_{i,j} = \infty$
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 $C_{i,j}$ += 0
16                 else
17                     $C_{i,j}$ += {cmp *prop-LED($z_p(u_i), z_p(v_j)$)
                        + (1 - cmp)*hash-cmp($z_p(u_i), z_p(v_j)$) }
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                 $u_{\hat{i}} = parent(u_i)$,     $v_{\hat{j}} = parent(v_j)$
23                 $C_{i,j} = C_{i,j} + C_{\hat{i},\hat{j}}$
24     CED ( $g^q, g^c$ ) = Munkres-Algorithm (C)
25     nCED = $\frac{\text{CED}(\ g^q, g^c\ )}{(|g^q|+|g^c|)/2}$ ,     SIM $(d_q, d_c) = e^{-n\text{CED}}$
26 else
27     SIM $(d_q, d_c)$ = SIMGNN ( $g^q, g^c$ )
```

Line 18 For applying Adjacency-Munkres, the minimum edge replacement cost is added to the cost matrix using Equation 3.
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 25 Normalize CED to the graph sizes, and apply an exponential function to convert it to a similarity score in the range of (0, 1].
Lines 26 - 27 If in testing phase, apply the learned mapping function, SIMGNN to predict the similarity score from the HARGs.

Generalization. (1) Algorithm 1 assumes that the edge labels for level 0 is 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 are affected by it which remains valid as long as same heuristics is 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.

4 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 O_H and their values.

4.1 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.

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:

```
[^]*$q_H$[^.]+
```

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.

$$SIM(q_H, s) = \max_{w \in s} SIM(q_H, w) \qquad (5)$$

(a) ***Word Embedding.*** Tokens in each sentence and in the key-phrase are represented by **Word2Vec** [35] 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,

$$SIM(q_H, w) = cos(u_{q_H}, u_w) = \frac{u_{q_H} \cdot u_w}{\|u_{q_H}\| \cdot \|u_w\|} \tag{6}$$

(b) ***Word Synsets.*** Tokens and key-phrases are represented by **Wordnet** [11] synsets in `NOUN` form. For similarity/distance metric, we used the Wu-Palmer similarity [71]. Given the synsets of q and w are s_{q_H} and s_w,

$$SIM(q_H, w) = wpdist(s_{q_H}, s_w) \tag{7}$$

Classification Model. The similarity search problem is re-designed as a classification problem where the sentences are considered as input sequence, and the key-phrases are considered as labels. 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. [74], 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 [54] (**SBert**). Finally, we use the NLI model to calculate the probability P. Given SBERT embedding of a sequence s is denoted with B_s,

$$SIM(q_H, s) = P(s \text{ is about } q_H \mid B_s, B_{q_H}) \tag{8}$$

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.

4.2 Iterative Search for Properties

We now formally describe the POSID algorithm, which uses the models described in Section 4.1. We start with the observations that led to the POSID algorithm.

Observations. We make the following observations: **(O1)** Common to **O3.4**, object-properties have the single and multiple value contrasts. **(O2)** Some properties follows specific patterns such as, gender = `{male, female, man, woman}`, whereas some properties have variable values, as shown in **O3.5**. **(O3)** Adjectives (ADJ) are used for naming or describing characteristics of a property, or used with a NOUN phrase to modify and describe it. **(O4)** Property values can span multiple tokens, but they tend to be consecutive. **(O5)** Property values for clothes generally include the color, a range of colors, or a description of material. **(O6)** 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`. **(O7)** After a token with VBG tag, until any ADJ or NOUN tag is encountered, any tokens describing a P_{DCP} {Determiner, Conjunction, Preposition}, or a Participle, or Adverb is part of the property-name. An exception would be any P_{PAV} {participle, adverb, or verb} preceded by any $P_{P\epsilon}$ {pronouns or non-tagged tokens}, which ends the mention of a property-name.

Algorithm 2: POSID

```
Input: Collection of Sentences, $T_s$
Output: Collection of $\langle name, values \rangle$ pairs, $\langle\langle \mathbf{o}_p, z_p \rangle\rangle$
$fo_p \leftarrow$ {GENDER, RACE, HEIGHT}
$COLOR_{syn} \leftarrow$ SYNSETS("color", NOUN)[0]
$C_s \leftarrow$ EXTRACT-CANDIDATE-SENT-RE($T_s$ , $Q_H$)
if $C_s$ is $\phi$ then
    $C_s \leftarrow$ EXTRACT-CANDIDATE-SENT-MODEL($T_s$ , $Q_H$)
$O_H \longleftarrow \emptyset$   /* Collection of $\langle o_p, z_p \rangle \equiv \langle name, values \rangle$ pairs */
foreach $s$ in $C_s$ do
    foreach $o$ in $fo_p$ do
        /* $L_z$ is last token in $s$ which is a property-value */
        $L_z$ = RE-PROP-VALUES($s, \mathbf{o}_p$)
        $O_H$.APPEND ($\mathbf{o}_p, L_z$)
    $s_p$ = RE-PROP-VALUES($s$, CLOTHES)
    if $s_p$ is $\phi$ then  $s_p \leftarrow s \setminus L_z$
    $N_{idx} \longleftarrow \emptyset$                /* Index-List for property-name */
    $D \leftarrow \emptyset$                       /* List for property-values */
    $T_o \leftarrow$ TOKENIZE-WORD($s_p$)   /* List of tokens from $s_p$ */
    $T_a \leftarrow$ 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 $T_a$      */
    for $(w, t)$ in $T_a$ do
        if $t_1$ is VBD then  continue
        if $t_2$ is VBG and $t_1$ is VBD then  continue
        if $t_i \in P_{DCP} \cup P_{PAV}$ then
            if $t_i \in P_{PAV}$ and $t_{i-1} \in P_{P\epsilon}$ then break
            $N_{idx}$.APPEND (i)
        else if $t_i$ is ADJ then
            $N_{idx} \longleftarrow \emptyset$     /* re-initialize name index-list */
            $D$.APPEND ($w_i$)
        else if $t_i$ is NOUN then
            $S_w \leftarrow$ SYNSETS($w_i$, NOUN)
            $N_{idx}, D, d_{color}$ = MATCH-W-COLOR($S_w, N_{idx}, D$)
            if $d_{color}$ then continue
            $N \leftarrow w_i$
            $N \leftarrow$ POPULATE-PROPERTY-NAMES($N_{idx}, N, T_a$)
              /* finalize property-name & assign the values */
            if $t_{i-1}$ is NOUN and $O_H[-1].name == w_{i-1}$
              then  CONCAT ($O_H[-1].name, w_i$, " ")
            else $O_H$.APPEND ([$N, D$])
            $N_{idx} \longleftarrow \emptyset, D \leftarrow \emptyset$ // re-initialize Lists
        else  break
return $O_H$
```

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 gender, race, and/or height.

Lines 3 - 5 Extract the candidate sentences with the RE-SEARCH. 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. `RE-PROP-VALUES` 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, `RE-PROP-VALUES` returns either a partial sentence s_p starting with `wearing`, or an empty string. In case of an empty string, extract the remaining string from L_z after discarding the extracted values in lines 8 - 10.
Lines 18 - 19 If first and second token is verb, it is the start for the RELATION property. Following **(O6)**, ignore consecutive verbs until another tag is encountered.
Lines 20 - 22 Following **(O7)**, capture tokens from a VERB until any pronoun or non-tag as free-form property value for CLOTHES.
Lines 23 - 25 Capture the adjectives as clothes descriptions, and initialize the next property.
Lines 27-29 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 30-31 If a noun-phrase is not a color, it is considered as cloth-name with multiple tokens i.e., *dress pants, tank top, dark clothing*. Populate the property-name by backtracking the name-index list.
Line 33 If previous token is NOUN and does not match last token of the previous property-name, description for next property has started. Finalize the current property name and value by appending it to result. Otherwise, in line 32, amend the last inserted property-name by appending the current token to it.

Generalization. Algorithm 2 assumes that the property identifier is intended for human-properties. POSID can be generalized to any object-properties in text as long as the property-names and type of values are known. Search-string for fixed-valued properties have 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 are highly non-restrictive phrases and can be constructed from entity types or entity names.

5 EXPERIMENTS

Dataset Construction. We adopt the **MARS** person re-ident ification dataset from [78] to benchmark the property identifiers in 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 from the local 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 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 the **(InciText + MARS)** dataset to evaluate the retrieval performance of FemmIR. 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 the ground-truth, we ranked the data samples in ascending order of the penalties for the mismatched properties. The properties were chosen depending on the user requirement, and the mismatches were assigned different penalties. In Example 1.1, 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 on top color, there is a chance of it being the same person given 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, hence the highest importance. Exact matches are the top most in the ranking with a zero penalty.

Settings. For Word2Vec, we used the 300 dimensional pretrained model from NLTK [33] 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 [70] 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, suspect, shirts, pants` }. We follow the original train/test partition of MARS [78] dataset for benchmarking. For models in [7, 20, 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 API from clapper[4]. We did not use the local node-node interaction information during the training phase for FemmIR.

[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

	Attr-Only			Attr-Value			θ_H	q_H
Models	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

Figure 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

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

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

Figure 4: Human Attribute Extraction Results

Property Identification in InciText dataset. We compared the baseline RE-model with the other approaches in Section 4.1 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 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 a subset of properties in O_H for InciText. Figure 4 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.

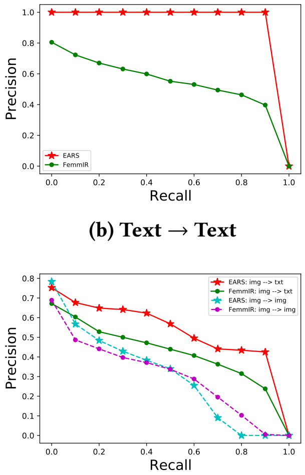
Benchmarking for Property Identifiers in Visual Modality. Since MARS has a large amount of ground truths for person attributes, we compared existing models from *Person Re-Identification* task. From the CNN models, we used the image-based Resnet50 [17] as baseline. Due to the temporal nature of videos, we also compared the 3D-CNN [20], CNN-RNN [34], Temporal Pooling and Temoporal Attention [7] models. As a heuristic based model, we chose the color-sampling model from [64]. Figure 3 describes the bench-marking 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 on pixel values, hence the presence of sunlight or clouds may have adversely affected the performance.

Retrieval Performance of FemmIR. We compared FemmIR with the EARS method [61]. Since EARS does not require any training, we only used the test set in (InciText + MARS) . We formulated the JOIN queries in EARS method on properties from Example 1.1. The results were a union among an exact match, and partial matches for the individual properties.

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

Query	Target	EARS	FemmIR
Image	Text	0.54	0.40
	Image	0.27	0.27
	Video	0.33	0.29
	All	0.30	0.28
Text	Text	1.0	0.52
	Image	0.37	0.29
	Video	0.46	0.33
	All	0.43	0.31
Video	Text	0.62	0.43
	Image	0.30	0.29
	Video	0.37	0.30
	All	0.34	0.30
	Avg	**0.44**	**0.33**

(a) Performance of EARS and FemmIR in mAP(%)



(b) Text → Text

(c) Image → Text, Image

Figure 5: Performance of FemmIR on (InciText + MARS)

(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 the dependence on property identifier performance. Precision-recall (PR) curves in Figure 5b and 5c show that at lower degrees FemmIR perform comparably with EARS, but with higher degrees of recall, the performance degrades. We will perform ablation studies (using local node-node interaction or eliminating imbalance of modalities in training data) in the future for FemmIR performance improvement.

6 RELATED WORK

Metric Learning. [12, 31] uses hinge rank loss to minimize intraclass variation while maximizing interclass variation. [10] minimized the loss function using hard negatives with a variant triplet sampling, but needs fine-tuning and augmented data. [73] uses an additional regularization in the loss function with adversarial learning. [68] 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. [36, 60] use weak signals from entity and relationship similarities retrieved from video captions and text. [61] assumes knowledge of the translation module which makes it less adaptable to novel modalities. [1] 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. [27, 28, 55, 62] learns semantically enriched representations of multi-modal instances by using global and local attention networks. Similarly, FemmIR uses graph convolutional network [23] to align the most important nodes contributing to the overall similarity, denoting the most similar properties between samples.

Content-based Data Discovery. [14, 26, 38, 47, 56, 61] 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 [61] 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.

Cross-modal Correlation Learning. [44, 53, 76, 77] use canonical correlation learning to linearly project the low-level features into a common subspace. For non-linear projections, [2, 21, 40, 41, 65] extended the linear methods [65] or used shallow [40, 41] or deep networks [21] to learn the correlations. SDML [18] removes the dependency of jointly learning from all modalities by predefining a common subspace and using a deep supervised auto-encoder for each modality. DSRL [67] directly learns the pairwise similarities by integrating relation learning, capturing the implicit nonlinear distance metric which FemmIR also focuses on. Most of these works assume the presence of class labels [76, 77], choice of appropriate feature extraction, and translation models for specific modalities. This limits the capability to integrate new sources or use pre-existing features/properties. FemmIR separates the feature extraction modules from the retrieval module and integrates pre-identifier property from any modality using graph encoding networks.

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. [6, 15] used sentence encoders and dense neural networks to combine lexical and semantic features for finding similar sentences in electronic medical records and academic writing.

7 SUMMARY AND FUTURE DIRECTIONS

We introduced the problem of mismatch between the information need and model 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. We demonstrated the performance of FemmIR in identifying the relevant data to the user example without supervised training and additional computational 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.

8 ACKNOWLEDGMENTS

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

REFERENCES

[1] Yazeed Alaudah, Motaz Alfarraj, and Ghassan AlRegib. 2019. Structure label prediction using similarity-based retrieval and weakly supervised label mappingStructure label prediction. *Geophysics* 84, 1 (2019), V67–V79.

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *International conference on machine learning*. PMLR, 1247–1255.

[3] Renzo Angles and Claudio Gutierrez. 2008. Survey of graph database models. *ACM Computing Surveys (CSUR)* 40, 1 (2008), 1–39.

[4] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. 2018. Graph edit distance computation via graph neural networks. *arXiv preprint arXiv:1808.05689* (2018).

[5] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep neural networks for learning graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[6] Qingyu Chen, Jingcheng Du, Sun Kim, W John Wilbur, and Zhiyong Lu. 2018. Combining rich features and deep learning for finding similar sentences in electronic medical records. *Proceedings of the BioCreative/OHNLP Challenge* (2018), 5–8.

[7] Zhiyuan Chen, Annan Li, and Yunhong Wang. 2019. Video-Based Pedestrian Attribute Recognition. *CoRR* abs/1901.05742 (2019). arXiv:1901.05742 http://arxiv.org/abs/1901.05742

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[9] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Kevin Murphy, Shaohua Sun, and Wei Zhang. 2014. From Data Fusion to Knowledge Fusion. *Proc. VLDB Endow.* 7, 10 (jun 2014), 881–892. https://doi.org/10.14778/2732951.2732962

[10] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612* (2017).

[11] Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database.* Bradford Books.

[12] A Frome, GS Corrado, J Shlens, et al. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems*, Vol. 26.

[13] Sainyam Galhotra, Anna Fariha, Raoni Lourenço, Juliana Freire, Alexandra Meliou, and Divesh Srivastava. 2022. DataPrism: Exposing Disconnect between Data and Systems. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 217–231. https://doi.org/10.1145/3514221.3517864

[14] Ralph Gasser, Luca Rossetto, and Heiko Schuldt. 2019. Multimodal Multimedia Retrieval with Vitrivr. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval* (Ottawa ON, Canada) *(ICMR '19)*. Association for Computing Machinery, New York, NY, USA, 391–394. https://doi.org/10.1145/3323873.3326921

[15] Chooi Ling GOH and Yves LEPAGE. 2020. Finding Similar Examples for Aiding Academic Writing using Sentence Embeddings. (2020).

[16] Saket Gurukar, Nikil Pancha, Andrew Zhai, Eric Kim, Samson Hu, Srinivasan Parthasarathy, Charles Rosenberg, and Jure Leskovec. 2022. MultiBiSage: A Web-Scale Recommendation System Using Multiple Bipartite Graphs at Pinterest. https://doi.org/10.48550/ARXIV.2205.10666

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[18] Peng Hu, Liangli Zhen, Dezhong Peng, and Pei Liu. 2019. Scalable Deep Multimodal Learning for Cross-Modal Retrieval. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. ACM, New York, NY, USA, 635–644. https://doi.org/10.1145/3331184.3331213

[19] Melanie Imhof and Martin Braschler. 2018. A study of untrained models for multimodal information retrieval. *Information Retrieval Journal* 21, 1 (2018), 81–106.

[20] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2012), 221–231.

[21] Meina Kan, Shiguang Shan, and Xilin Chen. 2016. Multi-view deep network for cross-view classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 4847–4855.

[22] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[23] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* abs / 1609.02907 (2016). arXiv:1609.02907 http://arxiv.org/abs/1609.02907

[24] Pradap Venkatramanan Konda. 2018. *Magellan: Toward building entity matching management systems.* The University of Wisconsin-Madison.

[25] Yu Kong and Yun Fu. 2022. Human action recognition and prediction: A survey. *International Journal of Computer Vision* 130, 5 (2022), 1366–1401.

[26] Michalis Lazaridis, Apostolos Axenopoulos, Dimitrios Rafailidis, and Petros Daras. 2013. Multimedia search and retrieval using multimodal annotation propagation and indexing techniques. *Signal Processing: Image Communication* 28, 4 (2013), 351 – 367. https://doi.org/10.1016/j.image.2012.04.001

[27] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV).* 201–216.

[28] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 4654–4662.

[29] Zechao Li, Jinhui Tang, Liyan Zhang, and Jian Yang. 2020. Weakly-supervised semantic guided hashing for social image retrieval. *International Journal of Computer Vision* 128, 8 (2020), 2265–2278.

[30] Yongjiang Liang and Peixiang Zhao. 2017. Similarity search in graph databases: A multi-layered indexing approach. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 783–794.

[31] Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. 2016. Deep coupled metric learning for cross-modal matching. *IEEE Transactions on Multimedia* 19, 6 (2016), 1234–1244.

[32] Haopeng Liu, Shan Lu, Madan Musuvathi, and Suman Nath. 2019. What Bugs Cause Production Cloud Incidents?. In *Proceedings of the Workshop on Hot Topics in Operating Systems* (Bertinoro, Italy) *(HotOS '19)*. Association for Computing Machinery, New York, NY, USA, 155–162. https://doi.org/10.1145/3317550.3321438

[33] Edward Loper and Steven Bird. 2002. 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* (Philadelphia, Pennsylvania) *(ETMTNLP '02)*. Association for Computational Linguistics, USA, 63–70. https://doi.org/10.3115/1118108.1118117

[34] Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. 2016. Recurrent convolutional network for video-based person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1325–1334.

[35] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs.CL]

[36] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. 2019. Weakly supervised video moment retrieval from text queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 11592–11601.

[37] Timothy Osborne. 2019. *A dependency grammar of English: An introduction and beyond.* John Benjamins Publishing Company.

[38] Servio Palacios, K.M.A Solaiman, Pelin Angin, Alina Nesen, Bharat Bhargava, Zachary Collins, Aaron Sipser, Michael Stonebraker, and James Macdonald. 2019. WIP - SKOD: A Framework for Situational Knowledge on Demand. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, Vijay Gadepally, Timothy Mattson, Michael Stonebraker, Fusheng Wang, Gang Luo, Yanhui Laing, and Alevtina Dubovitskaya (Eds.). Springer International Publishing, Cham, 154–166.

[39] Servio Palacios, K. M. A. Solaiman, Pelin Angin, Alina Nesen, Bharat Bhargava, Zachary Collins, Aaron Sipser, Michael Stonebraker, and James Macdonald. 2019. WIP - SKOD: A Framework for Situational Knowledge on Demand. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, Vijay Gadepally, Timothy Mattson, Michael Stonebraker, Fusheng Wang, Gang Luo, Yanhui Laing, and Alevtina Dubovitskaya (Eds.). Springer International Publishing, Cham, 154–166.

[40] Yuxin Peng, Xin Huang, and Jinwei Qi. 2016. Cross-Media Shared Representation by Hierarchical Learning with Multiple Deep Networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York,

New York, USA) *(IJCAI'16)*. AAAI Press, 3846–3853.

[41] Yuxin Peng, Jinwei Qi, Xin Huang, and Yuxin Yuan. 2017. CCL: Cross-modal correlation learning with multigrained fusion by hierarchical network. *IEEE Transactions on Multimedia* 20, 2 (2017), 405–420.

[42] Yu-Chi Pu, Wei-Chang Du, Chien-Hsiang Huang, and Chen-Kuo Lai. 2012. Invariant feature extraction for 3D model retrieval: An adaptive approach using Euclidean and topological metrics. *Computers & Mathematics with Applications* 64, 5 (2012), 1217–1225.

[43] Rashid Jalal Qureshi, Jean-Yves Ramel, and Hubert Cardot. 2007. Graph based shapes representation and recognition. In *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer, 49–60.

[44] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM international conference on Multimedia*. 251–260.

[45] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *CoRR* abs/1804.02767 (2018). arXiv:1804.02767 http://arxiv.org/abs/1804.02767

[46] Jing Ren, Feng Xia, Xiangtai Chen, Jiaying Liu, Mingliang Hou, Ahsan Shehzad, Nargiz Sultanova, and Xiangjie Kong. 2021. Matching algorithms: Fundamentals, applications and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence* 5, 3 (2021), 332–350.

[47] El Kindi Rezig, Anshul Bhandari, Anna Fariha, Benjamin Price, Allan Vanterpool, Vijay Gadepally, and Michael Stonebraker. 2021. DICE: Data Discovery by Example. *Proc. VLDB Endow.* 14, 12 (jul 2021), 2819–2822. https://doi.org/10.14778/3476311.3476353

[48] El Kindi Rezig, Lei Cao, Giovanni Simonini, Maxime Schoemans, Samuel Madden, Nan Tang, Mourad Ouzzani, and Michael Stonebraker. 2020. Dagger: A Data (not code) Debugger. In *CIDR*.

[49] Kaspar Riesen and Horst Bunke. 2008. IAM graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 287–297.

[50] Kaspar Riesen and Horst Bunke. 2009. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision computing* 27, 7 (2009), 950–959.

[51] Kaspar Riesen, Michel Neuhaus, and Horst Bunke. 2007. Bipartite graph matching for computing the edit distance of graphs. In *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer, 1–12.

[52] Martin Rohrmeier. 2011. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music* 5, 1 (2011), 35–53.

[53] Jan Rupnik and John Shawe-Taylor. 2010. Multi-view canonical correlation analysis. In *Conference on Data Mining and Data Warehouses (SiKDD 2010)*. 1–4.

[54] Nil s Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL]

[55] S. Sah, Sabarish Gopalakrishnan, and Raymond Ptucha. 2020. Aligned attention for common multimodal embeddings. *Journal of Electronic Imaging* 29 (2020), 023013 – 023013.

[56] Sheikh Muhammad Sarwar and James Allan. 2020. Query by Example for Cross-Lingual Event Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1601–1604. https://doi.org/10.1145/3397271.3401283

[57] Sumedh Sawant. 2013. Collaborative filtering using weighted bipartite graph projection: a recommendation system for yelp. In *Proceedings of the CS224W: Social and information network analysis conference*, Vol. 33.

[58] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. *Advances in neural information processing systems* 26 (2013).

[59] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. 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., 926–934. http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion.pdf

[60] KMA Solaiman and Bharat Bhargava. 2022. Open-Learning Framework for Multi-modal Information Retrieval with Weakly Supervised Joint Embedding. (2022).

[61] K. Solaiman, T. Sun, A. Nesen, B. Bhargava, and M. Stonebraker. 2022. Applying Machine Learning and Data Fusion to the "Missing Person" Problem. *Computer* 55, 06 (jun 2022), 40–55. https://doi.org/10.1109/MC.2022.3145507

[62] Yale Song and Mohammad Soleymani. 2019. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1979–1988.

[63] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2443–2449.

[64] 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. 2020. Surveillance Video Querying With A Human-in-the-Loop. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics with SIGMOD*.

[65] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. On deep multi-view representation learning. In *International conference on machine learning*. PMLR, 1083–1092.

[66] Xiaolan Wang, Xin Luna Dong, and Alexandra Meliou. 2015. Data X-Ray: A Diagnostic Tool for Data Errors. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (Melbourne, Victoria, Australia) *(SIGMOD '15)*. Association for Computing Machinery, New York, NY, USA, 1231–1245. https://doi.org/10.1145/2723372.2750549

[67] Xu Wang, Peng Hu, Liangli Zhen, and Dezhong Peng. 2021. DRSL: Deep Relational Similarity Learning for Cross-modal Retrieval. *Inf. Sci.* 546 (2021), 298–311.

[68] Jiwei Wei, Xing Xu, Yang Yang, Yanli Ji, Zheng Wang, and Heng Tao Shen. 2020. Universal weighting metric learning for cross-modal matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13005–13014.

[69] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-Modal Graph Convolution Network for Personalized Recommendation of Micro-Video. In *Proceedings of the 27th ACM International Conference on Multimedia* (Nice, France) *(MM '19)*. Association for Computing Machinery, New York, NY, USA, 1437–1445. https://doi.org/10.1145/3343031.3351034

[70] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. https://doi.org/10.18653/v1/N18-1101

[71] Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033* (1994).

[72] Bing Xiao, Xinbo Gao, Dacheng Tao, and Xuelong Li. 2008. HMM-based graph edit distance for image indexing. *International Journal of Imaging Systems and Technology* 18, 2-3 (2008), 209–218.

[73] Xing Xu, Li He, Huimin Lu, Lianli Gao, and Yanli Ji. 2019. Deep adversarial metric learning for cross-modal retrieval. *World Wide Web* 22, 2 (2019), 657–672.

[74] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161* (2019).

[75] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. 2022. A survey of modern deep learning based object detection models. *Digital Signal Processing* (2022), 103514.

[76] Xiaohua Zhai, Yuxin Peng, and Jianguo Xiao. 2013. Learning cross-media joint representation with sparse and semisupervised regularization. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 6 (2013), 965–978.

[77] Liang Zhang, Bingpeng Ma, Guorong Li, Qingming Huang, and Qi Tian. 2017. Generalized semi-supervised and structured subspace learning for cross-modal retrieval. *IEEE Transactions on Multimedia* 20, 1 (2017), 128–141.

[78] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. 2016. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision*. Springer, 868–884.

[79] Weiguo Zheng, Lei Zou, Xiang Lian, Dong Wang, and Dongyan Zhao. 2013. Graph similarity search with edit distance constraint in large graph databases. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 1595–1600.

[80] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. 2020. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567* (2020).