SimMeme: Semantic Based Meme search

Maya Ekron
Tel Aviv University
mayaekron@mail.tau.ac.il

Tova Milo
Tel Aviv University
milo@post.tau.ac.il

Brit Youngmann
Tel Aviv University
brity@mail.tau.ac.il

ABSTRACT

With the proliferation of social image-sharing applications, image search becomes an increasingly common activity. In this work we focus on a particular class of images that convey semantic meaning beyond the visual appearance, and whose search presents particular challenges. A prominent example are *Memes*, an emerging popular type of captioned pictures, which we will use in this demo to demonstrate our solution. Unlike in conventional image-search, visually similar Memes may reflect different concepts. Intent is sometimes captured by user annotations (tags), but these too are often incomplete and ambiguous. Thus, a deeper analysis of the semantic relations among Memes is required for an accurate search. To address this problem, we present SimMeme, a semantic aware search engine for Memes. SimMeme uses a generic graph-based data model that aligns all the information available about the Memes with a semantic ontology. A novel similarity measure that interweaves common image, textual, structural and semantic similarities into one holistic measure is employed to effectively (and efficiently) answer user queries. We will demonstrate the operation of SimMeme over a large repository of real-life annotated Memes which we have constructed by web crawling and crowd annotations, allowing users to appreciate the quality of the search results as well as the execution efficiency.

1 INTRODUCTION

With the ubiquity of image-sharing platforms like Flicker, WeChat and Instagram, an abundance of social images is available on the Internet and the demand for dedicated image-search engines has become increasingly pertinent. In this demo we focus on a particular class of images that convey semantic meaning beyond the visual appearance. A prominent example are *Memes*, a popular new type of captioned images. We present SimMeme, a dedicated image search engine that is based on a comprehensive novel similarity measure which we harness for this task.

The term Meme, defined as a cultural symbol or idea that spreads and mutates, was coined by evolutionary biologist R. Dawkins. Similar to the way that DNA spreads from one location to another, a Meme travels from mind to mind. The majority of modern Memes are humorous captioned photos, often ridiculing human behavior. The term has become synonymous to a clever or humorous combination of text and image (see Figure 1 for examples).

The key challenge with Meme search stems from the gap between text description and visual presentation. Unlike traditional image search, visually similar Memes may express a radically different concept. Indeed, in many cases, without the text or context, the image alone is meaningless. The associated text is often ironic and thus captions that contain common words may also express different ideas. User textual annotations (or tags), enabled by social websites, provide meaningful semantic information about the Meme intent, however, effectively utilizing it introduces its own challenges. Tags are often incomplete, while user queries in image search engines tend to be short and ambiguous. Moreover, Memes are chiefly characterized by concepts, and the same concept can be redundantly captured by different tags.

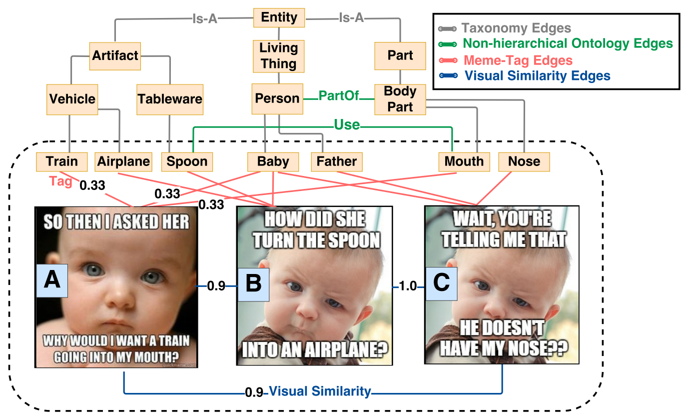


Figure 1: [Best viewed in color] Example HIN, aligning structural relations (tags and Memes) with semantic ontology.

To illustrate, consider again Figure 1. Some of the data available about the Memes is captured by the information network inside the dotted rectangle. We see here two types of weighted edges: one connecting Memes to their tags, with weights reflecting the importance of each tag as determined by the users, and the second type of edges connect two Memes, with weights reflecting their visual similarity as determined, e.g., by some image similarity software[1] (for the sake of conciseness, some edge labels and weights were omitted). The semantic information, in particular the tags' ontology, is depicted by the edges and nodes outside the rectangle. Assume that the user wishes to find Memes similar to Meme B. The available data indicates that all three Memes have in common only the tag `Baby` and that Meme C's picture is visually identical to that of Meme B. However, a finer analysis of the semantic relation among the tags allows to better estimate the similarity between the Memes' intent. Concretely, the tag-pairs `Airplane-Train` and `Spoon-Mouth` that tag, resp., Memes A and B, are more semantically related than any combination of the tags of Meme C and Meme B, other than perhaps the `Baby-Baby` and `Father-Baby`

CIKM'17, Pan Pacific, Singapore
 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

[1]Textual similarity edges, that, for instance, represent the sentiment of the caption, can be added in an analogous manner, as well as nodes/edges that capture information about the Memes' creator or similar available data. We omit this in the example for simplicity.

pairs. Moreover, `Baby` and `Father` are more common tags than `Train` and `Airplane` (i.e., more Memes are tagged by these tags), and since it follows from standard argumentation of information theory that an estimation of similarity increases more drastically when indicated by a less frequent event, such information needs to be considered as well.

Traditionally, image search engines rely mostly on visual features, and even when considering textual features or user tags (that are typically evaluated separately), make limited use of semantic information [5, 6, 11, 16]. In contrast, our system takes a holistic approach that considers all available data and allows users to indicate the importance of visual appearance and semantic meaning in their search. SimMeme uses a simple generic data model that aligns the information network and the semantic knowledge (the ontology) into a single (weighted) heterogeneous information network (HIN). The advantages of this data model are derived from its schema-less nature, namely, its flexibility allows for a natural incorporation of any other data source, e.g., expressing visual/textual resemblance of the Memes's picture/caption via edges among Memes. Over this HIN we define a new similarity measure that differs from existing measures in its comprehensive nature and allows to effectively and efficiently identify Memes that match the user's request.

Typically, similarity measures in HINs consider solely the structure of the network[2] [4, 12, 14]. Contrarily, semantic measures (e.g. [7, 10]), quantify similarity based on meaning or semantic content, which play no role in structural-based computations. A naive approach would be to simply consider some (non) linear combination of the two distinct notions. However, we argue that a carefully chosen tighter weaving of the concepts, leads to superior similarity analysis. Our dedicated measure (formally defined in Section 2.2) interweaves two well studied measures, SimRank [4], a popular structural similarity measure, and Lin [7], a generic Information Content (IC)-based semantic measure. In SimRank, the structural similarity of two network nodes is determined, recursively, by estimating the similarity of its neighboring nodes. Informally, we refine SimRank by weighting, at each step of the computation, the nodes' structural similarity with their semantic scores, while considering the edge weights as well. Since the network includes, among others, edges that capture visual and textual similarities, these are naturally taken into consideration in the computation. Interestingly, we show that the interplay between the two notions not only yields a more profound similarity measure, it also allows to speed up the performance by pruning irrelevant computation, thereby facilitating an efficient retrieval of the most relevant Memes.

Users can query SimMeme by pointing to an existing Meme or by providing search keywords (tags) and possibly a picture [3]. The query is then interpreted as a new Meme-node connected to its corresponding tags. A prominent feature of SimMeme is allowing users to provide custom weights to the keywords, or tune the importance of visual appearance and the Meme's intent (in case no weights are provided, a uniform distribution is assumed). SimMeme then employs its dedicated similarity measure to (i) efficiently retrieve the top-k most relevant Memes, and (ii) cluster the results into semantically meaningful subsets [11, 16], thereby enabling the users to quickly focus on the groups most relevant to their intent. The system further provides the users with an *explanation* for why a specific Meme is proposed, highlighting the semantic, visual and structural relations of the retrieved result to the query. This last important feature distinguishes our solution from Machine Learning (ML)-based approaches, as it allows users to get a clear understanding of the selection and ranking process.

While our exposition on the features of SimMeme focuses on Meme-search, we emphasize that SimMeme is a general-purpose system applicable to the general tagged-image retrieval problem.

Demonstration. We will demonstrate the capabilities of SimMeme by inviting the audience to view the results of example queries, as well as to compose their own ad-hoc queries trough the system's intuitive UI. Subsequently, we will review the results and the explanations for their selection and ranking. We will use the system's UI to examine the segments of HIN and the ontology that influenced the selection and ranking of the results, thereby gaining an intuition on the underlying similarity measure and the retrieval process. Furthermore, to better highlight the effectiveness of our novel measure, we will compare the results to those obtained by several other baseline approaches. (See Section 4)

Related Work. Tagged-image search has received much attention in previous works (e.g., [11, 16]). Image search engines often use (non) linear combination of the textual and visual features to return relevant images [5, 6], yet make limited use of semantic information. Particularly, IC-based measures are not incorporated. Lin goes beyond TF/IDF based techniques, which consider the prevalence of tags, such that more frequent tags contribute less to the search rank, and also takes into account the IC of their common information.

Several similarity measures for information networks have been proposed in the literature [1, 4, 5, 12, 14]. We have adopted SimRank because of its generality, simplicity and wide range of optimizations [3, 13]. Our approach, detailed in the following sections, allows previous optimizations to be immediately carried over to our setting, and be further enhanced with semantic based pruning.

Given examples for tagged Memes and a search query, one may attempt to use ML methods to train a model that implicitly incorporates semantic relations between tags. As often with ML, accounting for every possible query may require wide-ranging training data that is not always available, and, as mentioned, the results are harder to explain. However, we consider such methods a complementary effort to enrich the HIN with more data, e.g., learning sentiment or visual similarity (e.g., [2, 15]) or automatic extraction of tags [17].

2 TECHNICAL BACKGROUND

We next provide a brief overview of the data model and the similarity measure we uses, then explain how similarity scores and top-k results are computed efficiently. Full details can be found in [8].

2.1 Knowledge Repository

Our data is modeled using a weighted (multi) graph, representing an Heterogenous Information Network (HIN). The HIN obtained by gluing together two graphs: the information graph and the ontology. The former captures all the available data about the images (Memes), e.g., the attached tags, the image creator, etc. Weights on edges in

[2]Since the network includes edges that capture visual/textual similarity, these are considered as part of the network.
[3]Additional Meme properties such as creator Id or alike may also be added in the "Advanced Search" screen.

our setting provide an essential information, such as the importance of each tag to a given Meme. Additional knowledge sources, such as visual/textual similarity between Memes is naturally incorporated in the model via edges between Memes, with weights reflecting their similarity scores. (To avoid explosion of edges, we include only edges with similarity scores above a minimal threshold). To capture the tags' semantics, we align the information network and the semantic ontology into a single graph (i.e., each tag is aligned to its corresponding entity in the ontology), using standard entity alignment tools such as [9]. Within the ontology, we pay below a special attention to the hierarchical *taxonomy* of concept, which refers to a parent-child (i.e., is-a) relations among concepts.

2.2 Similarity Notions

Our proposed similarity measure, interweaves two well studied measures, SimRank [4], and Lin [7].

Lin is a common IC-based semantic measure, that is defined over concept taxonomies. Intuitively, the key to the similarity of two concepts is the extent to which they share information in common, indicated by the most specific concept that subsumes them both. The IC of a concept is quantified as negative the log likelihood, that is, as probability increases, informativeness decreases. Intuitively, the similarity between concepts here measures the ratio between the amount of information needed to state their commonality and the information needed to describe them (we omit the formal definition). Note that Lin measure is traditionally defined only for nodes in the taxonomy, i.e., only tags in our case. We extend the definition assignment for all other pair of nodes, e.g., between Meme-nodes, to the constant value of 1 (indicate no semantics is available).

SimRank is a generic, commonly used, structural similarity measure. It follows a simple and intuitive assumption: "two objects are similar if they are related to similar objects". Similar to [1], we augment SimRank by taking into account link weights, but consider semantic similarity as well. Formally, given two vertices a and b, their SimRank score, and correspondingly their similarity score (denoted as $sim(a, b)$), is defined as follows. If $a = b$ then both $simrank(a, b) := 1$ and $sim(a, b) := 1$, else: $simrank(a, b)$ is given by the following formula *without the red colored parts*.

$$\frac{Lin(a,b) \cdot c}{N} \sum_{i}^{|I(a)|} \sum_{j}^{|I(b)|} sim(I_i(a), I_j(b)) \cdot W(I_i(a), a) \cdot W(I_j(b), b)$$

where c is a decay factor $\in [0, 1]$, $I(v)$ is the set of neighbors of v, an individual neighbor is denoted as $I_i(v)$, and $N := |I(a)| \cdot |I(b)|$.

The highlighted red parts indicate our extensions to the SimRank's standard formula: (i) an additional semantic factor is added ($Lin(a, b)$) to account for the semantic similarity for two given node; (ii) the edge weights are taken into consideration when weighing neighbors similarity. Correspondingly, in our case, the normalization factor N is set to $N := \sum_i^{|I(a)|} \sum_j^{|I(b)|} W(I_i(a), a) \cdot W(I_j(b), b)$. For both measures, if $I(a)$ or $I(b)$ are the $\emptyset$, then similarity scores are defined as 0.

2.3 Optimizations and Query Evaluation

Given a query, the similarity scores must be computed *online* w.r.t the user query. We next described two optimizations used in our system to enable an efficient and scalable computation. Then, we shortly describe the retrieval process.

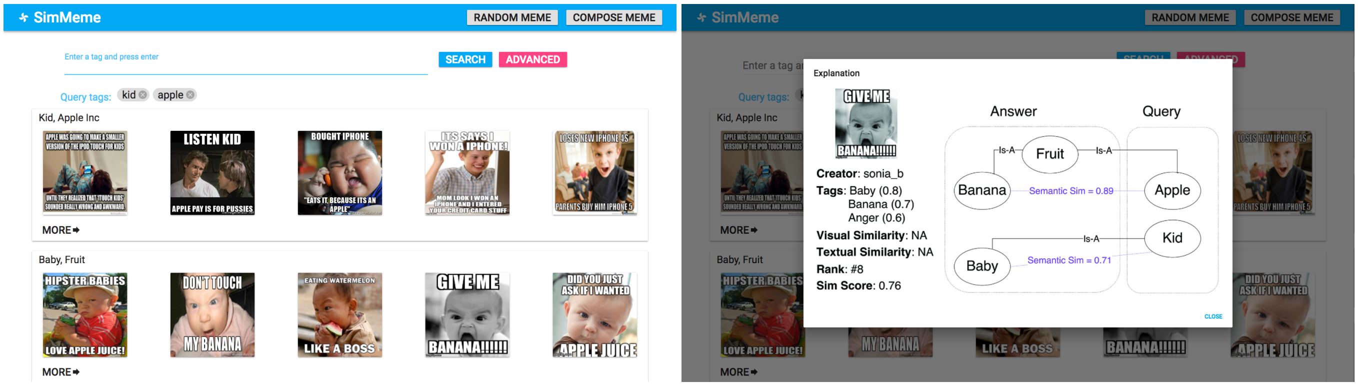
Approximation. Jeh and Widom [4] have established a connection to a "random surfer" model that allows to compute SimRank using random walks. Intuitively, SimRank measures how soon two random surfers are expected to meet, if they randomly walk on the graph. An extensive body of optimizations regarding SimRank's approximated computation are based on the connection to this interpretation. A prominent example is the *Monte-Carlo framework* [3, 13]. A main challenge addressed in our work, was to carefully modify the underlying model to take into account both semantics and weights, while preserving the necessary properties that enable such optimizations. The Monte-Carlo framework of SimRank utilizes the concept of random walks and the fact that SimRank can simply be approximated by using the average meeting index of samples walks. In contrast, in our case, rather than uniformly choosing the next step, the random surfer must be aware of both the semantics and weights. We next shortly explain our novel definition of *semantic-aware random walks* that we developed to address this, then describe our approximated computation using such walks.

Given two random walks $t_1 = \langle u_1, .., u_k \rangle$ and $t_2 = \langle v_1, .., v_k \rangle$, we denote t the coupled random walk of t_1 and t_2, where $t = \langle (u_1, v_1), .., (u_k, v_k) \rangle$ (following [3]). Denote $\tau(t)$ the prefix of t until the first meeting point (inclusive). The weight of a coupled walk t, $W[t]$ is defined as the product of weights in each step, multiply by the semantic similarity. Denote: $t : (u, v) \rightsquigarrow (x, y)$ a coupled random walk from the nodes u, v to the nodes x, y, resp., and $l(\tau(t))$ is its length. Furthermore, denote: $T = \{\tau(t) : (u, v) \rightsquigarrow (x, x)\}$ the set of all distinct prefixes that first met in some node (x, x). We can prove (see [8]) that: $\sum_{t \in T} W[t] \cdot c^{l(t)} = sim(u, v)$.

Our proposed approximation uses a different aggregation over (coupled) random walks than that of SimRank. While for SimRank, an approximate similarity is computed as an average over the coupled random walks, we use a summation, that is, we sum the weight of all (distinct prefixes of) coupled walks. From lack of space, we omit the error bound analysis and only note that our experiments over real data demonstrate the efficiently and high accuracy of our adjusted framework [8].

Pruning. The interplay between semantic and structural similarity further allows to speed up the computation by pruning Memes that posses tags that are all semantically unrelated to the query keywords. For that we utilize the fact (which follows from the definition of our measure) that the semantic similarity between tags provides an upper bound on the overall similarity score. Given the query keywords, the candidate set of potential relevant Memes are only the Meme connected to a tag that is sufficiently related to one of the keywords, i.e., their Lin score is above a certain minimal threshold. Moreover, we can avoid estimating the relevance of unpromising tags by leveraging the taxonomical relations. That is, exploring the tags taxonomy from top to bottom can suggest which categories are irrelevant to the query keywords, thus avoiding unnecessary computations of unrelated to the search query tags.

To complete the picture, recall that a search query is modeled as a (new) Meme-node connected to the corresponding tags. To properly match between the query keywords and the HIN (tag) nodes, we use standard tools for string matching (e.g., Levenshtein



(a) Results generated for the keyword query "Apple and Kid"

(b) An explanation popup to an obtained result.

Figure 2: [Best viewed in color] SimMeme UI.

distance). Then, using our pruning technique, we construct the initial set of all potentially relevant Memes. As this set may still be very large, we employ the above mentioned approximation method to efficiently compute their similarity scores, and return the top-k most relevant Memes.

3 SYSTEM OVERVIEW

SimMemeis implemented in Python 2.7 using NetworkX library https://networkx.github.io/. The user inserts her search query via the *User Interface*, which is then submitted for evaluation. The extracted keywords are parsed and the relevant tags are passed to the *Pruning Module*. The pruning module identify all semantic-related tags, then extracts the initial candidate Memes set. The obtained set is then passed to the *Similarity Estimator* over which similarity operators are evaluated. Finally, the results are summarized into semantic-based clusters, using our novel similarity measure.

Figure 2 depicts the UI using the keywords query "Apple" and "Kid" (in this example, the user provided only keywords and not an image or a Meme). In the upper part, a user issues a search query by typing keywords in the search box. Clicking on the "Advanced" button will invoke the configuration dialog to set various parameters (see Section 4). SimMeme displays the clustered results as horizontal blocks of Memes, where each block contains a sub-set of semantically related Memes (e.g., in Figure 2 (a), the upper block contains "Apple Inc." related Memes, while the lower block contains "Fruit" related ones). A user may click on a Meme to obtain an explanation popup highlighting the reasons for the Meme's selection and the similarity score derivation.

4 DEMONSTRATION

To demonstrate the capabilities of SimMeme we have constructed a repository of 10K annotated Memes, via web crawling and crowd annotations (using CrowdFlower https://www.crowdflower.com/). The domain ontology we use for the demo is WordNet (https://wordnet.princeton.edu/). CIKM conference participants will be challenged to retrieve Memes in a variety of topics, and query the system by providing keywords and possibly a picture, or by composing their own Memes. Our system will then select the top-k most similar Memes, out of Memes composed by other participants or the system's repository. Additionally, users can tune their search queries via the advanced-UI, where they can provide custom weights to the keywords, add an image, or tune the importance of visual appearance and semantics. Going a step further, the user may click on a result Meme to view an explanation about the selection process and ranking (see Figure 2 (b)). The explanation would provide an intuition about the relevant parts of the HIN that led to the selection and obtained similarity scores.

One of the key objectives of the demonstration is to enable the audience to gauge our proposed similarity measure's effectiveness. Towards that end, we introduced in the advanced-search dialog an option to choose an alternative similarity measure, and compare the results to those obtained by our novel dedicated measure. In particular, we will show the search results obtained by several previous approaches (e.g., [1, 5, 11, 12]) as well as (non) linear combination of SimRank and Lin, demonstrating the advantages of our solution.

REFERENCES

[1] I. Antonellis, H. G. Molina, and C. C. Chang. Simrank++: query rewriting through link analysis of the click graph. *PVLDB*, 2008.
[2] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, 2010.
[3] D. Fogaras and B. Rácz. Scaling link-based similarity search. In *WWW*. ACM, 2005.
[4] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *SIGKDD*, 2002.
[5] X. Jin, J. Luo, J. Yu, G. Wang, D. Joshi, and J. Han. irin: image retrieval in image-rich information networks. In *WWW*. ACM, 2010.
[6] Y. Kiyoki, T. Kitagawa, and T. Hayama. A metadatabase system for semantic image search by a mathematical model of meaning. *SIGMOD*, 1994.
[7] D. Lin. An information-theoretic definition of similarity. In *ICML*, 1998.
[8] T. Milo, A. Somech, and B. Youngmann. Semsim: Combining structual and semantic similarity (full version). http://bit.ly/2qc3ktO.
[9] N. F. Noy, M. A. Musen, et al. Algorithm and tool for automated ontology merging and alignment. In *AAAI*, 2000.
[10] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.
[11] B. S. Seah, S. S. Bhowmick, and A. Sun. Prism: concept-preserving summarization of top-k social image search results. *PVLDB*, 2015.
[12] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 2011.
[13] B. Tian and X. Xiao. Sling: A near-optimal index structure for simrank. In *SIGMOD*. ACM, 2016.
[14] C. Wang, Y. Sun, Y. Song, J. Han, Y. Song, L. Wang, and M. Zhang. Relsim: Relation similarity search in schema-rich heterogeneous information networks. In *SDM*, 2016.
[15] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.
[16] S. Wang, F. Jing, J. He, Q. Du, and L. Zhang. Igroup: presenting web image search results in semantic clusters. In *SIGCHI*. ACM, 2007.
[17] L. Wu, R. Jin, and A. K. Jain. Tag completion for image retrieval. *TPAMI*, 2013.