Different Modes of Semantic Representation in Image Retrieval

Rory Bennett

20 March 2018

Abstract

Image retrieval systems are supposed to only retrieve images that are relevant to a given query. Therefore, they need methods by which to represent the meaning of both the query word and the image, so that these meanings can then be compared. Distributional semantic models typically use semantic vectors to represent words' meanings, based on the extent to which they appear near other words in text. By comparing these semantic vectors, we can compare words' meanings, and thus find words that are simlar or relevant to each other. In this study, I extend this idea, to implement an improved image retrieval system: I build semantic vectors for both words in text and captioned images, and compare these vectors to find, for each query term, the image whose meaning is most relevant to the query's meaning. I consider taking information from captioned images and inserting it into text, to build "multi-modal" semantic vectors that contain information across modes of meaning, from both textual and visual data; I also consider filtering which images I consider for vector comparison, based on whether they contain, in their captions, words that are similar to the query term in the text. Results show that overall, inserting perceptual information into the text actually causes the image retrieval system to retrieve less relevant images.

Contents

List of Figures

List of Tables

1 Introduction

Image retrieval systems take as input a query word, and are tasked with returning images that are relevant to the query. To ensure that a relevant image was returned for a query word, text-based image retrieval (TBIR) systems have traditionally relied on the query words as being directly associated with the available images, in the form of captions, tags, etc. But the vast majority of image tags are relatively concrete words. A word's concreteness is determined by the extent to which it refers to something "physical or spatially constrained" [1]; less physical concepts are termed as "abstract." More concrete terms include "dog" and "chocolate"; more abstract words include "respect" and "love." Similarly, imageability refers to the extent to which terms can be associated with images; imageable terms can thus be thought of us as a subset of all concrete terms, as concreteness is determined with respect to all modes of perceivability. For example, the word, "jury" is concrete, in that one can be part of a jury, and probably even imagine what it looks like when a jury convenes; that is, they would look like a group of people sitting around and discussing a case. However, just showing an image of people discussing something is not constrained just to denoting a jury; so there are other concrete words that are more imageable as well. For example, "tractor" has a clear image associated with it, so it is imageable and thus also concrete. When people are asked to assign tags to images, they generally associate images with the most prominent objects or tangible features of an image, which these concrete terms tend to address, so they are unsurprisingly the most common words associated with images (see figure 1). But studies by Barsalou and Wiemer-Hastings [1] suggest that abstract and concrete concepts share situational content, such that abstract words can provide more general, overarching descriptions of extra-linguistic data (*e.g.*, images), and thus better represent the data's meaning. And by definition, abstract concepts "encode higher-level...principles than concrete concepts" [6]. Therefore, an abstract word should be able to describe situations involving different concrete concepts, across various images, so long as these images share a common overarching meaning.

Suppose, for example, a user queries an image retrieval system with the word, "elegant." "Elegant" is a very abstract term, because it does not refer to anything tangible, so it would likely tag zero images in the system's database, and thus return zero images. However, there are various terms, such as "swan" and "tuxedo," that are generally considered to be elegant, and because they are actually concrete, they are likely to also tag images showing swans and tuxedos, respectively. Because these things are considered to be elegant, "elegant" can always be used to enhance descriptions of "swan" and "tuxedo" as they appear in captions, and can thus always be added to a caption, to improve its overall interpretation of an image containing swans or tuxedos. So because "elegant" shares situational content with these various concrete terms, it relates, to some extent, to the images that these concrete terms tag, and vice-versa: were the system

Figure 1: Both images are annotated by humans. Their captions contain mostly just concrete words, which provide only a basic description of the things shown. The left image's caption contains, "grass," "puppy," "canine," "bark," "black," "dog," "ear," "lassie," and "german." The right image's caption contains, "church," "brick," "green," and "door."

to return, for "elegant," images of tuxedos and swans, the user could easily see that the objects in these images are generally considered to be elegant. But the Distributional Hypothesis states that "[t]he degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic context in which A and B can appear" [9]. This suggests that we should expect to see "elegant" appear near "tuxedo" and "swan" in text, at least more often than other words whose images and thus meanings cannot be as easily associated with "elegant." We should thus assume that if we could determine, for any given abstract word, which concrete words co-occurr with it in text and thus describe the same thing, the abstract term would provide a sufficient high-level description of many of the images associated with those concrete terms. Conversely, I hypothesize that if a user were to query a TBIR system with this abstract word, they should be satisfied if the system returned images associated with these concrete words. This would provide strong evidence that words of varying concreteness are grounded in perception; that is, even the most abstract words can effectively be represented with perceptual content. In order to determine which concrete words (and thus their images) relate to an abstract word, I will need to compare their respective semantic representations.

Semantic representations are ways of representing words' meanings. Distributional semantic models (DSMs) represent these meanings by relying on words' "statistical distribution in text" [3], under the hypothesis that similar words have similar surrounding (context) words. In this experiment, I attain these distributions by constructing, for each word, a "continuous vector space where semantically similar words are mapped to nearby points" [13]. Given a word, its corresponding DSM is a vector of numbers, each of which represents the word's distribution with other context words in a textual corpus. Table 1 shows what the vectors might look like for the words involved in the above example, for "elegant." Multimodal semantic models [3], meanwhile, integrate information from different modalities, including linguistic (text) and visual (images) data, to attain DSMs of words that might otherwise be limited, were the information

Word	Vector Space
elegant	$0.443251, 0.124922, -0.137212, \ldots$
majestic	$0.418276, 0.117542, -0.156315, \ldots$
swan	$0.357891, 0.085432, -0.178954, \ldots$
tuxedo	$0.339842, 0.068432, -0.199981, \ldots$
chocolate	$0.194786, 0.014711, -0.309462, \ldots$
fear	$0.153233, 0.003281, -0.347981, \ldots$

Table 1: Vector representations of various words. Note that if a word is more similar to "elegant," that word's vector is more similar to that of "elegant." Although "majestic" is evidently the most similar to "elegant," it is also a relatively abstract word, whereas "swan" and "tuxedo" are concrete and still have fairly similar vectors, so captions containing either of these words can be enhanced by "elegant."

limited to one modality. These models require an information propagation step, which selects information from a dataset of one modality, and adds it to the dataset of another modality. Since this alters the data itself, it alters the vector spaces produced by distributional semantic techniques. Hill and Korhonen [8] show that when constructing a word's DSM from multi-modal data, the appropriate amount of information to propagate between modalities depends on the word's concreteness: multimodal models improve semantic representations for concrete words, but extralinguistic information propagation is actually detrimental to abstract words' DSMs.

For this experiment, I am going to implement a TBIR system that, given query word w, returns images directly associated with the words that have semantic representations most similar to w's. The system will determine these similar words by utilizing the abovementioned distributional semantic techniques. So one focus of this study is to find those methods that return the most relevant images for w, as this helps me determine the best way to find the words most similar to w. This in turn indicates which method yields the most accurate semantic representation for w. Furthermore, I am varying the amount of perceptual information that I propagate to then derive my vectors. Therefore, my more specific focus is to gain insight into the appropriate amount of perceptual information that is required for improving image retrieval, for words of varying concreteness and imageability. This helps in undestanding the extent to which abstract words are in fact perceptually grounded.

The rest of this paper is organized as follows. Section 2 provides introduces distributional semantics in more depth. Section 3 describes multimodal distributional semantics, and corresponding techniques and models from past experiments, including those that Hill and Korhonen implemented, which I will utilize for my own experiment. In section 4, I present and analyze the results of my pre-experiment, in which I try to replicate one of Hill and Korhonen's experiments. In section 5, I describe my current experiment, which I will use to test which techniques best improve TBIR, given query word w, by altering which context words are considered most similar to w; I also here describe the methods with which I will evaluate my results.

2 Distributional Semantics

2.1 Traditional Approaches

Distributional semantics is a research area that focuses on acquiring a word's meaning. DSMs seek to do so by constructing, for a given word, a corresponding semantic representation. DSMs build a word's representation based on the Distributional Hypothesis, so given a word w, a DSM utilizes techniques for identifying other words that most often occur with w in the same or in similar contexts. In traditional distributional semantics, "context" refers to a text corpus, so two words here co-occur if they literally appear closer to each other within the corpus's sentences. One of the most popular ways to represent these distributions, i.e., obtain a word's semantic representation, is with a vector space. Building word vectors is just one of various mathematical techniques that can be used "to turn the informal notion of contextual representation into empirically testable semantic models" [9]. Lenci [9] specifies that if we associate a word's contextual representation with an n-dimensional vector space, we can also "conceive words as points in a 'distributional space,' i.e. a space whose dimensions are provided by the relevant linguistic contexts, and in which the position of a word-vector is determined by its statistical distribution in each context." Assuming, by the Distributional Hypothesis, that this is a semantic representation, each vector maps its corresponding word to a common semantic space, such that the similarity between two words' meanings can then be determined by finding the similarity (e.g. cosine similarity) between their respective vectors. Figure 2 shows what the vectors in table 1 might look like in a corresponding two-dimensional semantic space. Lenci [9] also mentions that "many differences exist depending on the specific mathematical and computational techniques,..., the definition of the linguistic context used to determine the combinatorial spaces of lexical items, etc."

Turney and Pantel [12] present, in a survey paper, three different classes of vector space models: term-document, word-context, and pair-pattern matrices. Each of these defines "context" differently, such that each of the models has a different application. The term-document model is just a matrix presenting the number of occurrences of each word in each of the given documents, such that the rows of the matrix are the words and the columns are the documents. Whereas term-document counts the number of times a given word appears in a given document, word-context matrices instead look at the number of co-occurrences between words, across all documents included: rows represent words, and so do columns. Given word x, the word-context model defines a word y to co-occur with x if it appears within x's k-window; that is, given some constant k, y must appear either as one of $\frac{k}{2}$ words to x's left, or as one of $\frac{k}{2}$ words to x's right. The word-context model is thus most similar to the standard vector space model that Lenci [9] describes, at least

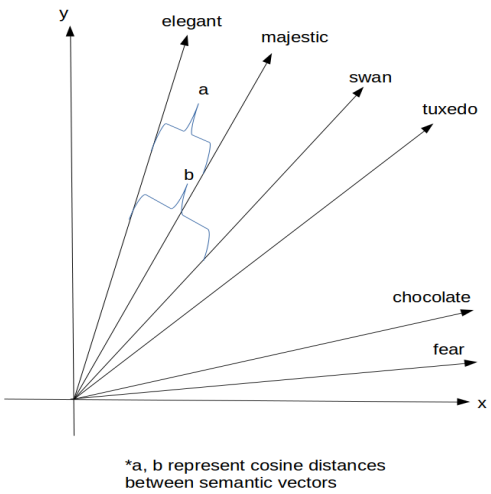

Figure 2: Note that $a < b$, indicating that "majestic" is more similar to "elegant" than "swan" is. However, "swan" and "tuxedo" are relatively close to "elegant," when compared to other, irrelevant terms of varying concreteness, like "chocolate" and "fear."

in terms of its definition of "context." Note that like term-document vectors, each number, or dimension, within a word-context vector is meant to reflect the frequency with which a given word co-occurs with a context word. So if there are N words in a document's vocabulary, the word-context vector would create an $N \times N$ matrix, in which each row would be associated with one vocabulary word. So given a word w's row, at index i in the matrix, the more frequently another word c occurred within w's k-window in the document, the higher the value would be at $[i, j]$ of the matrix, where j would be c's corresponding column index. For example, the following is just a fragment of the Brown corpus's word-context matrix [7], with a k-window of size 7.

$$
\begin{array}{c}
 \\
\textit{apricot} \\
\textit{pineapple} \\
\textit{digital} \\
\textit{information}
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc}
\textit{aardvark} & \dots & \textit{computer} & \textit{data} & \textit{pinch} & \textit{result} & \textit{sugar} & \dots
\end{array} \\
\begin{pmatrix}
0 & \dots & 0 & 0 & 1 & 0 & 1 & \\
0 & \dots & 0 & 0 & 1 & 0 & 1 & \\
0 & \dots & 2 & 1 & 0 & 1 & 0 & \\
0 & \dots & 1 & 6 & 0 & 4 & 0 &
\end{pmatrix}
\end{array}
$$

For term-document models, a similar matrix can be built given N vocabulary words and D total documents. Pair-pattern matrices have rows that represent pairs of words, and the columns represent the context

in which those pairs of words appear: a row might contain "butcher-meat," and one such column that this pair would correspond to is one that says, "X cuts Y," because a butcher cuts meat. Turney and Pantel [12] go on to talk about the specific applications that each of these three models have. Since the term-document model defines the "context" in which a word appears as simply the documents in which it appears, it is often used, unsurprisingly, for returning documents that contain a given query. It also helps with grading essays, by comparing the language in one essay to that of a high-quality reference essay. Meanwhile, the word-context document measures context at a higher granularity, between individual words. Naturally, because its definition of "context" is so similar to Lenci [9]'s, this model is ideal for finding similarity between individual words. Finally, the pair-pattern matrix has the most rigorous definition of "context," as it accounts for the relationship that multiple words have, as determined by additional, third-party words. Therefore, its applications include relational similarity, i.e., comparing contextual relations between different pairs of words; each pair of words can thus be represented in vector space, the same as individual words.

The applications of the above three models, while all different from each other, are also somewhat analogous to each other, and thus reinforce how all DSMs are common in certain ways. As per Lenci [9], they all share certain properties. Put simply, each dimension of a word's semantic vector represents an individual context in which the given word appears. But for a corpus of V distinct vocabulary words, each vocabulary word's corresponding vector would need to have V dimensions, to account for every possible co-occurrence with every possible word in the corpus. Note that above, we show only a small fragment of the Brown corpus's word-context matrix; in reality, each word's full row (semantic vector) would have a column for each of the corpus's vocabulary words. So when we combine the vectors of a corpus's vocabulary words to form an $V \times V$ matrix, the matrix is likely to get more sparse as V becomes arbitrarily large, because given a word w, a larger matrix is more likely to contain other words that are irrelevant to w, and that are thus less likely to co-occur with it. This means that for each w, its vector will contain many zeros, making for much unnecessary computation when comparing vectors to find words similar to w; these computations make word comparison intractable within arbitrarily large corpora. Meanwhile, Mikolov et al. [10] have developed the Skip-gram Model, a vector space model built using unique mathematical techniques, such that it outperforms previous models while efficiently computing representations for very large corpora.

2.2 The Skip-gram Model

Mikolov et al. [10] sought, with their novel DSM, to learn "high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary." Doing so would improve on pre-

vious architectures, which have only been successfully trained on a vector space of a few hundred million words. While maximizing word representation accuracy, the authors wanted to also minimize computational complexity, the amount of parameters that need to be accessed to fully train their two models. I have used this model in my own experiments. Given a corpus with V vocabulary words, the model utilizes a neural network, shown in figure 3 [11], that includes "hidden layer" and "output layer" matrices, each comprising semantic vectors for each of the vocabulary words, for when they appear as "target" and "context" words, respectively. A "target" word is the word in the corpus whose vector representation is currently being compared with surrounding words; a "context" word, conversely, surrounds the current target word. Therefore, when this model compares a target word's representation with its context words' vectors, it takes the former from the hidden layer, and latter from the output layer. Also note that when the neural network finishes running on the entire corpus, we take each word's hidden layer vector as its final semantic representation, with which we can then perform word similarity tasks. The model's efficiency comes from the fact that in the hidden layer, each vector's dimensionality N (set to 300 in my and in Hill and Korhonen [6]'s experiments) is non-linear in V, such that it is much smaller, making for more efficiently computable word vector comparisons, independent of the size of the corpus or its vocabulary. This independence comes from the fact that while the vectors in the hidden layer initially contain values pertaining to word co-occurrence, like in the above DSMs, our neural network updates the current target and context words' vectors in their respective layers, as part of its backpropagation step, according to how well the target word w's vector predicts w's context words at the current point in the corpus, within a k-sized window. We acquire this predictive value by taking the dot product of the vectors for w and each context word c, and applying the Softmax function to convert each dot product to a value between 0 and 1. Then, given a ground truth vector of w's co-occurrence probabilities with other vocabulary words, where these probabilities are also between 0 and 1, we can compare our Softmax output to the corresponding ground truth probability. The smaller the difference in these two values, the better the model has predicted that c should appear in w's current context, so we do not need to update w's weights as much in the hidden layer, or the cs' weights in the output layer.

So we are still assuming, as per the Distributional Hypothesis, that a word's surroundings determine its meaning, but the Skip-gram model's neural network allows us to work with vectors of a smaller dimension, because of the way it accounts for any of the V vocabulary words co-occurring with w. Whereas in more basic DSMs, a vector needs V columns to account for co-occurences with each of the corpus's V vocabulary words, the neural network compares w's N-dimensional vector with the vector of each surrounding word within the k-window, at each point in the corpus in which w appears. Therefore, by the time the neural network has finished, it has accounted for appropriate updates to w's vector, in relation to every word c

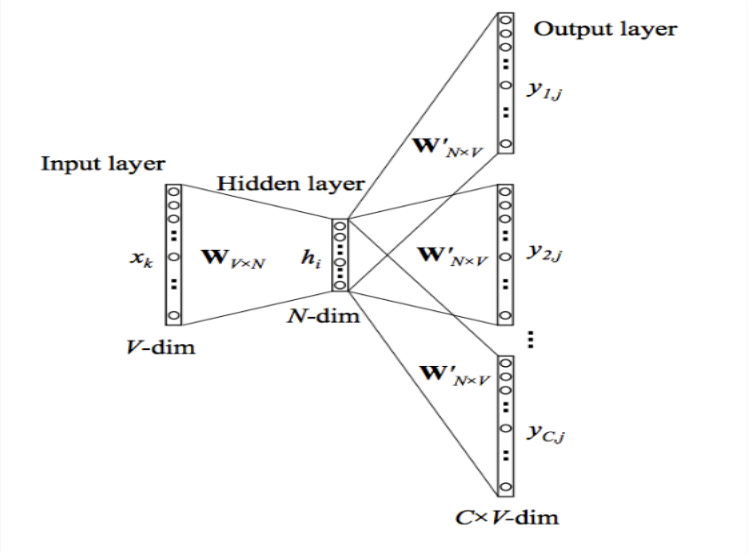

Figure 3: Schematic for the hidden-layer neural network upon which the Skip-gram model is built.

that actually surrounds w in the corpus; comparing two words' N-dimensional vectors after this fact thus still indicates the similarity between the words' overall contexts, and thus still indicates their similarity in meaning, by the Distributional Hypothesis. Also, each time the neural network's backpropagation step is applied to update w's vector, it modifies every dimension in the vector. Therefore, while relative to the word-context model, the significance of each individual vector dimension is unclear, each vector is dense, and its application to similarity testing between words remains.

Given a corpus, the model first applies some "simple" DSM, more reminiscent of the basic word-context model, to compute initial probabilities of vocabulary words co-occurring with each other. This produces, for each word w of the corpus's V vocabulary words, a vector r_w of some arbitrariliy fixed length N, containing the initial weights to be stored in the hidden layer matrix. Figure 3 shows how, after the initial V vectors are produced, they are combined to form an $V \times N$ matrix, such that each column comprises the current vector weights for an individual word; each column in this hidden layer matrix gives a vector representation for each target word w. These initial vectors are also each transposed, and then combined to form the $N \times V$ matrix in the output layer, with which we multiply the current target word w's column vector from the hidden layer, to produce a $1 \times V$ vector of w's dot products with each vocaulary word. So for each sentence S in the corpus, for each target word w, the model feeds as input a $1 \times V$ vector, whose indices correspond to each of the V vocabulary words. Only the index corresponding to the current w is here set to one, while the rest are set to zero, so that when this vector is multiplied with the hidden layer's matrix, it produces a $1 \times N$ vector, the exact vector of weights corresponding to w. In this way, the hidden layer's matrix acts as

a lookup table for the current weights of w. The model then multiplies the target word's $1 \times N$ vector with the output layer's matrix, and given the k-sized window of w's context words in S, the model then takes, from the resulting $1 \times V$ vector, the dot products of r_v with each context word's vector $\hat{r_c}$ in the output layer. By then applying Softmax to each of these dot products, and comparing the result to the abovementioned ground truth probability, the model then determines the "probability of seeing context word c given... w," [6], given by

$$p(c|w) = \frac{e^{\hat{r_c} \cdot r_w}}{\sum_{v \in V} e^{\hat{r_v} \cdot r_w}}.$$

The model finally takes the negative log likelihood of the sum of all such $p(c|w)$, to update r_w in the hidden layer, as well as each $\hat{r_c}$ in the output layer, as per the neural network's backpropagation step. As $p(c|w)$ gets smaller, *i.e.*, as the prediction of seeing c given w is worse, the negative log likelihood of that $p(c|w)$ increases, which means the corresponding vectors in the hidden and output layers will be updated more. By updating r_w and each $\hat{r_c}$ accordingly, for each co-occurrence of w and c, this model learns r_w and $\hat{r_c}$ such that future calculations of $p(c|w)$ get as close as possible to the ground truth probability of seeing c, given w.

I have utilized this model throughout my own study, because it considers, for w and a context word c, their contextual similarity at a specific point in the corpus; it thus accounts for every possible co-occurrence between a given target word and its context words, while keeping vector dimensionality low and independent of the vocabulary size. This allowed me to construct word vectors from the vocabularies of such large corpora as Text8, which I used for my study, and which contains the first 100,000,000 characters on Wikipedia.

2.3 Query-Image Comparison Techniques

After constructing, for my experiment, word vectors using Word2Vec's implementation of the Skip-gram Model, I still needed a method by which to retrieve the right images, given the vector representation of a query word; below are two such methods that I used in my study. The main idea is that given a captioned image, I take the average of the vectors of the caption words to represent the image's overall meaning, such that I can compare it with the query term's vector. Then, those images whose average vectors have the highest cosine similarity with the query term's vector are taken to be the most relevant to the query.

Figure 4 illustrates the first of these techniques which, from here on out, we refer to as, "Technique A." Given a textual corpus, we derive a semantic vector for each of the text's vocabulary words, here using the Skip-gram Model. Then, given a dataset of captioned images, we look to each image, to derive a vector averaged over each of its caption words' individual vectors, to form a semantic representation of the overall

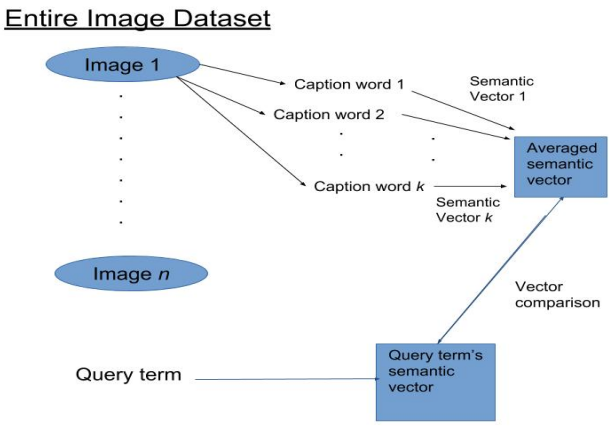

Figure 4: For each captioned image in our perceptual dataset, we average the vectors of the image's caption words, and compare the average vector with the query term's vector.

image. That is, for each of the image's caption words, if the Skip-gram Model derived a semantic vector for that word as it appeared in the textual corpus, then we include it when forming the image's average vector. After finding, for each image, the cosine similarity between the query term's vector and the image's average vector, we sort the images by their cosine similarity and return the top n images.

Figure 5 illustrates the second of these techniques which, from here on out, we refer to as, "Technique B." Here, we essentially do the same thing as in Technique A, but with an additional filter step. That is, rather than look at the average vector of each image in our perceptual dataset, we only consider those images that are directly tagged by the top 10 words in our textual corpus whose vectors have the highest cosine similarity to the query term's vector. So even if other images, not tagged by these top 10 words, were to have average vectors that were most similar to the query term's, this technique would would ignore them for images actually tagged by one of the top 10 words, even if their average vectors were less similar. Here, by giving priority to images on the basis of their tags' similarity with a query term in the context of text, we are accounting for the possibility that semantic similarity in the textual context is more informative than similarity in our perceptual context.

3 Multimodal Distributional Semantics

Lenci [9] explains that due to advanced computational techniques, we are afforded the ability to at least approximate words' semantics from textual corpora. However, it has been argued that if a DSM constructs

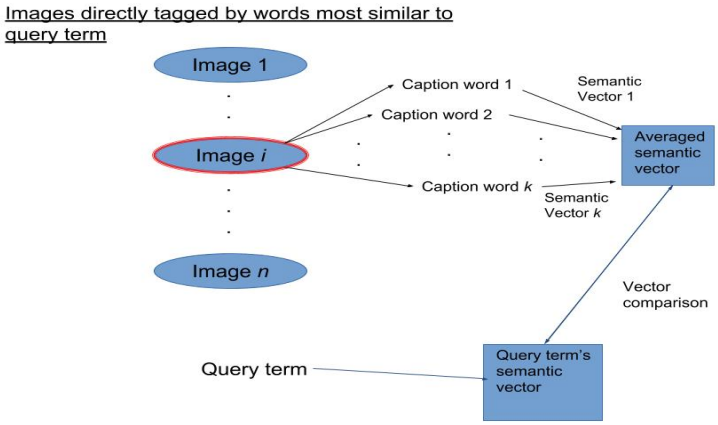

Figure 5: We only compare the query term's vector against the average vectors of those images that are directly tagged by words that are relatively similar to the query term in the textual corpus.

a word's semantic representation solely from other words that occur in the same text, without links to the outside world (e.g., via perception), then said representation is limited [2]. Harnad [5] formalizes this concern in what he calls the "symbol grounding problem," explained below. Multimodal distributional semantic techniques thus seek to address this problem.

3.1 The Symbol Grounding Problem

Harnad [5] explains how the meanings we attribute to textual words is ultimately based on the actual things that the words describe, which we only actually experience in the real world. Note that this idea applies to the most concrete terms, such as "dog," and the most abstract terms, such as "love." Because all words are rooted, or "grounded," in these experienceable things, Harnad [5] conjectures that representations more closely associated with experiencing the things themselves provides information more fundamental to their meaning. This suggests we could provide a better overall semantic representation for each thing, and associate it with the corresponding word, in this case in the form of an improved semantic vector. To illustrate his point, Harnad [5] presents his "Chinese/Chinese Dictionary Go-Round" scenario. That is, if you had to learn Chinese, and the only source you had was a Chinese/Chinese dictionary, you would never learn a thing: you would instead be stuck in an infinite loop, forever going back and forth between the dictionary's various foreign symbols, because even though symbols refer to things in real life, no meaning can be derived solely from the symbols without first knowing the things to which they refer. In other words,

the symbol's meaning is determined entirely by that to which they refer.

Harnad [5] goes on to explain that we can derive lower-level meaning for concepts from two kinds of nonsymbolic (nontextual, for the purposes of this experiment) representations: iconic and categorical. Iconic representations serve the purpose of allowing humans to "discriminate," [5], i.e., judge the degree to which two things being represented are similar or different. Take, for example, the iconic representation of a horse. An horse's iconic representation can be observed visually, in the form of say an image. Obviously, if this icon were to be compared with the image of a banana, someone could likely easily discriminate the two based on shape, color, etc. Categorical representations, meanwhile, help "identify" [5] specific, individual things, separate from other things that might have iconic representations similar to its own. If we again use images as our model for iconic representation, a horse and a zebra's icons would look fairly similar, yet a zebra is identifiable by its trademark striped coat. Naturally, the greater the similarity between two concepts, as with horses and zebras, the more likely the concepts' identifying features need to be "learned from experience," [5], in order to consistently identify each concept correctly. These nonsymbolic representations thus emulate how humans actually learn things through their experiences.

The issue of learning a meaning for a given concept can thus be reduced to obtaining all icons, filtered by their features, that most fully account for the concept's semantics. In the case of zebras, they are similar to horses in appearance, but their stripes suffice to distinguish them. Therefore, were we to encode the word "zebra" in a semantic vector space, we would want to integrate information from the vector for "horse" with information from the vector for "stripes" [5]. When evaulating their Skip-gram model, Mikolov et al. [10] performed similar arithmetic operations on the vectors their model produced. For example, given vector k for "king," vector m for "man," and vector 'w for "woman," $k - m + w$ produced a vector very similar to that for "queen," because while kings and queens are both royal, their gender alone can distinguish them. Therefore, although the symbol grounding problem hypothesizes that textual representations of concepts are limited when compared to their nonsymbolic counterparts, DSMs are able, by relying on distributions of words in corpora, to identify, to some extent, distinguishing features between words as they occur in text. To further improve DSMs' results, multimodal distributional semantics focuses on integrating information, not just between individual concepts' representations, but between symbolic and nonsymbolic data.

3.2 Past Multimodal Distributional Semantic Techniques

Multimodal models are distinguishable from traditional DSMs by their utilization of extralinguistic data to improve words' semantic representations. Due to the availability of both "large amounts of mixed media" [2] and semantic representations of images on the Web, images are the most common source of extralin-

guistic input for multimodal models. However, images may be represented differently based on what techniques are being used, as the projects described below illustrate. Also, how a model constructs images' semantic representations determines how the perceptual information can be integrated with the linguistic information.

Feng and Lapata [4] propose the first multimodal distributional semantic model. Their model uses a corpus comprising multimodal online documents: text documents that contain images, under the assumption that the document's textual content and images are related. Their model first constructs a term-document matrix, such that the columns representing each document comprise counts of each word in that document. Feng and Lapata [4] thus describe each column's contents as a "bag of words," because given a document d's corresponding column, one could construct a list of the words that appear in d, based on which rows yield nonzero values in the matrix. For example, say the rows in the matrix for a document corpus corresponded, from top to bottom, to the words, "apples," "bananas," "but," "I," "like," and "you." Then, if a document d within the corpus only contained the phrase, "I like apples, but you like bananas," the column for d could be represented as a list of numbers:

$$[1, 1, 1, 1, 2, 1].$$

But similarly, Feng and Lapata [4]'s model creates, for each image in the document corpus, a "bag of visual words" (BoVW). Just as a regular bag of words focuses on which words are actually present in a document, BoVW extracts, for each image, the visual features that are present [2]. Figure 6 illustrates the process by which a BoVW is built for an image of a violin. Given a set of images I over a document corpus, a multimodal model may apply computer vision techniques to identify, for each $i \in I$, the visual features containing "rich local information" [2], determined by clusters of colors and shapes that occur in i. These local features are represented in a low-level feature vector, based on their frequency in i. Figure 6 does not actually show this step, because it is from these low-level vectors that the final BoVW for the violin can be formed. That is, each of the features in the figure 6's histogram actually comprise clusters of these local features; each of these clusters is determined by the concentration of local features that are nearby each other in the image of the violin. These "visual words" correspond to the categorical features that Harnad [5] deems useful for identifying terms. Together, these clusters of local features form visual words: for example, the histogram in figure 6 indicates that the violin's base is especially prominent in the image, so when the frequencies of its visual words are encoded in the image's BoVW, the visual word corresponding to the base will be largely representative within the vector.

After constructing their term-document frequency matrix and their matrix of BovWs, Feng and Lapata

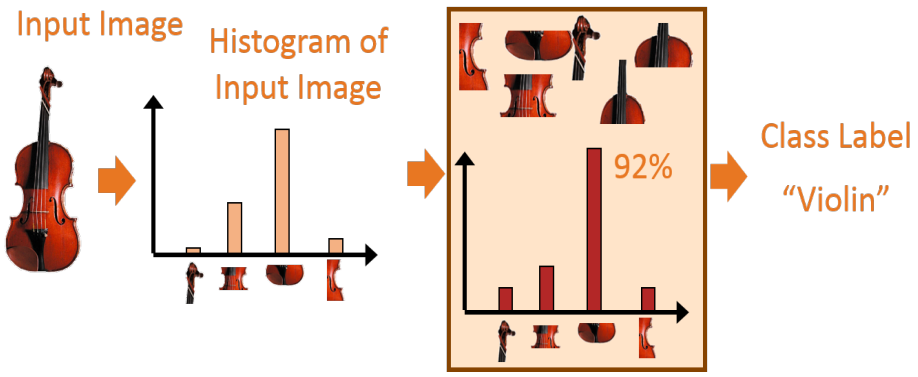

Figure 6: The process by which a BoVW is built, as a semantic representation for visual data, i.e. images.

[4] concatenate the two to get a multimodal semantic representation, d_{Mix}. Since theirs is the first multimodal DSM, this concatenation is the first instance of integrating information of different modes in some way. By integrating these different data structures, they attempt to propagate information given by the images' BoVWs to the documents' term-frequency matrix and improve the semantic representation of each word in the document corpus's vocabulary. Specifically, in the d_{Mix}'s resulting vector space, each word's vector representation can be determined both by the documents in which they occur, and visual features of the images in those documents. Feng and Lapata [4]'s results showed that their multimodal vectors outperform purely textual vectors in word similarity and word association tasks.

Hill and Korhonen [6] implemented a multimodal semantic model that improved upon Mikolov et al. [10]'s Skip-gram model, which only constructs DSMs for purely textual corpora. For textual data, they used the Text8 textual corpus, which comprises the first 100 million characters in Wikipedia. For perceptual data, they use the ESP-Game dataset, which consists of 100,000 images, "each annotated with a list of lexical concepts that appear in that image" [6]. Specifically, these images were annotated by humans, who were supposed to indicate what they saw within each image. They therefore provide a way of determining the visual features of each image, via human description, without resorting to computer vision techniques, to find distributions of the visual features themselves, as in [4]. Additionally, they use the Center for Speech, Language, and the Brain (CSLB) Norms dataset, of "semantic properties for 638 concrete concepts" [6], again produced by human annotators. Table 2 shows annotations for a few of these words in each perceptual dataset. For each word w that tags an image in these perceptual datasets, Hill and Korhonen [6] create a bag of words comprising all words that annotate the same images as w. These bags are referred to as "bags of physical features" (BoPFs) [6]. Using these annotating words, instead of actual visual words, follows from Harnad [5]'s conjecture that perceptual representations of concepts contain lower-level meaning; this implies that these annotations hold somewhat similar value to any visual clusters produced by computer

	Image 1	Image 2
ESPGame	red	wreck
	chihuahua	cyan
	eyes	man
	little	crash
	ear	accident
	nose	street
	small	

	Word 1: *Crocodile*	**Word 2:** *Screwdriver*
CSLB	has 4 legs (7)	has handle (28)
	has tail (18)	has head(5)
	has jaw (7)	is long (9)
	has scales (8)	is plastic (18)
	has teeth (20)	is metal (28)
	is green (10)	
	is large (10)	

Table 2: The ESP-Game dataset associates, with each image, a bag of relevant words. The CSLB dataset is a feature matrix, in which the more frequently annotators associate a feature with an image, the more relevant that feature is to the image.

vision techniques.

To evaluate the accuracy of the multimodal vectors their model produces, Hill and Korhonen [6] use association norms from the University of South Florida dataset. These association norms are such that humans were presented with a cue word, and they were supposed to respond with the (target) word they thought to be significantly relevant to the cue word. So given a cue word c, the association strength between it and each of its target words t is determined by the number of people who responded with t, over the total number of annotators who responded to c with some target word. Hill and Korhonen [6] find, for a given association norm with cue word c and target word t, the Spearman correlation between c and t's association strength, and the cosine similarity between c and t's multimodal vectors, as produced by their extended Skip-gram model. The extended version of Skip-gram is as follows. For each word w in Text8 that also annotates one of the images among the perceptual datasets, i.e., that has a BoPF, Hill and Korhonen [6] insert "pseudosentences" into Text8, of the same length as the sentence in which w appears, and comprising those words in w's BoPF. This is their method of integrating linguistic and extralinguistic information, and with this modified corpus, they perform Skip-gram, as in Mikolov et al. [10]'s original experiment. Hill and Korhonen determine cosine similarity between multimodal vectors to analyze the accuracy of their models' multimodal semantic representations. Upon evaluating the resulting vectors with the USF dataset, they note that semantic representations for concrete terms improve significantly; likewise, some more abstract words' representations are more accurate. However, for the most abstract terms, integrating perceptual information to Text8 is actually detrimental to the corresponding vectors' performance. Therefore, determining which words are similar to each other, and thus finding which are the relevant images that a TBIR should return, appears to in part be a function of the amount of perceptual information we propagate to a textual corpus. In [8], Hill and Korhonen try to address this concern.

In [8], Hill and Korhonen filter exactly when they add perceptual information to a textual corpus, based on the concreteness of each word in the text. Just as before, they construct BoPFs from the annotated

perceptual datasets they use, but based on their results in [6], they only add a pseudosentence when a word in the text with a BoPF is concrete enough. They make the assumption that if the images a word annotates are diverse, the word itself may have a meaning that applies to various more specific, concrete things, and should thus be considered abstract. If the images a given word tags are relatively monotonous, they likewise assume that the word is concrete. They determine the variety of the images a word annotates by constructing, for each image, the same type of BoVW that Feng and Lapata [4] used; they then take the cosine similarity, denoted α, of the visual features contained in these BoVWs, between all images tagged by that word. They then take the median of all such α, and if a word in the textual corpus has α below that median, it is considered concrete. If it is above this threshold, it is considered abstract. This is in line with my definitions of "concrete" and "abstract" in section 1, that whereas concrete words refer to more narrow concepts, abstract words generally have more overarching meaning, and thus may encapsulate the meaning of that to which a concrete word may describe. Filtering in this way yields significant improvements in the accuracy of semantic representations, from the experiment in [6], for concrete and abstract words alike. In section 4, I show the results of my having replicated Hill and Korhonen [6]'s experiment, as I needed to understand how to implement these techniques for my own experiments.

4 Preexperiment

When replicating Hill and Korhonen [6]'s experiment, I used the same linguistic and extralinguistic datasets as input, and the same evaluation methods. Table 3 shows my evaluation results. Note that each row in the table involves a different subset of the total association pairs from the USF dataset. Specifically, where it says, "Concrete/N" in the first row, it is referring to the fact that I here used association pairs in which both the cue and target word were concrete, and both were nouns. The USF dataset provides concreteness ratings for all words that it uses, both cue and target, so I ordered the cue words by concreteness into quartiles, such that the first quartile contained the most abstract cue words and the fourth quartile contained the most concrete cue words. Then, if an association pair had a cue word in the fourth quartile and a target word whose concreteness rating was at least as high as the least concrete cue word in the fourth quartile, and both words were nouns (also specified by the USF dataset), I used that pair when producing my correlations for the first row in the table. For instance, the word, "wire" has the lowest concreteness rating of all cue nouns in the fourth quartile, at 5.81 out of 7, and its target word, "telephone," has a concreteness rating of 6.15, and is a noun, so I included this pair when computing my correlation for the first row of table 3. I did similarly with pairs whose cue and target words' concreteness ratings were both within the range of the first quartile. So since the highest cue concreteness of all nouns in the first quartile is 3.38, and since

Perceptual Dataset	Concreteness/POS	My Correlation	H&K Correlation
ESP-Game	Concrete/N	0.259	0.301
CSLB	Concrete/N	0.169	0.239
Both	Concrete/N	0.221	0.296
Both	Abstract/N	0.278	0.250
Both	Abstract/V	0.180	0.175

Table 3: Spearman correlations, between association strengths for association pairs from USF dataset, and the cosine similarity between multimodal vector representations of each word in the given pair.

"virtue" and "goodness" have lower concreteness, their pair was included when I computed the correlation for the fourth row. Also note that the first column specifies from which of the two perceptual datasets, ESP-Game or CSLB, I inserted pseudo-sentences into Text8, before finding Spearman correlations. In order to ensure that each pair within each subset of the USF dataset could be actually be evaluated against vector representations, Hill and Korhonen [6] filtered those words for which they insert pseudosentences based on whether they appear in an association pair.

Discrepancies between my Spearman correlations and Hill and Korhonen [6]'s own may be attributed to differences in the datasets we each used. Hill and Korhonen [6] mention that their USF dataset does not actually have part-of-speech (POS) attributes for cue and target words, whereas the dataset I used, did. This suggests that they used, in their own experiment, an older iteration of the dataset, such that it may have been differently sized from my own. Furthermore, they mentioned their Text8 corpus contained 400 million words, while the link they provided took me to a page with a corpus of the same name, that contained only 100 million characters. It seems unlikely that they overlooked the actual description of the dataset they actually used, but if we truly used such different corpora, then Hill and Korhonen [6] would have much more data on which to run their model, which would likely contribute to them having results that are different from my own. That being said, I used the same Skip-gram model, and the same information propagation techniques as Hill and Korhonen [6] to derive my textual and multimodal vectors, so I know the cosine similarity between two such vectors will be valid. Furthermore, in my actual study, I am using those same vectors I produced during my replication, to retrieve images from the same ESP-Game dataset that I used to propagate information. Therefore I am assured that when I look for words that have vectors that are similar to the query word's, and that also tag images in the ESP-Game dataset, I will still be testing the effects of information propagation on my image retrieval system. This is independent of whether or not I used the same USF dataset as Hill and Korhonen [6].

5 Experiment Design

5.1 Approaches to Image Retrieval

The focus of this study is to determine which distributional semantic techniques retrieve the most relevant images, to better understand their effectiveness in building words' semantic representations. Therefore, I recruited human annotators to rate images retrieved from the ESP-Game dataset, by various such methods, based on the images' relevance to a query term. So given a set of query terms, I applied five image retrieval approaches to each word. The five approaches are as follows:

1. Retrieving only those images that contain the query term in their caption (direct tagging),
2. Technique A, using word vectors derived from the plain Text8 corpus,
3. Technique B, applied to the abovementiond plain Text8 corpus,
4. Technique A, applied to Text8, augmented by the perceptual information that was propagated to it from the ESP-Game and CSLB datasets, via Hill and Korhonen [6]'s technique, and
5. Technique B, applied to the abovementioned augmented Text8 corpus.

Direct tagging does not utilize distributional semantic techniques, but served as my control model. Since the query terms I picked for this experiment ranged over varying levels of concreteness and imageability, there was no guarantee that all terms would directly tag an image; for several of the terms, direct tagging retrieved zero images, which is the very problem discussed in section 1. So by applying the latter four approaches to these terms, I was able to best examine the effectiveness of distributional semantic techniques, on words that are primarily used in a textual context: these words are highlighted in red in figure **??**. These results are discussed below, in section 6.

Regarding the other four approaches, I applied each of the vector comparison techniques discussed in section 2.3, to each of the plain and augmented iterations of Text8, to account for how the propagation of perceptual information (or lack thereof) impacts their performance. That is, one could have hypothesized that Technique B would retrieve more relevant images for plain Text8, than for augmented Text8, because Technique B's filter step uses vector similarity in the textual context; since the vectors constructed from plain Text8 use only textual information, the information they provide may be free from any irrelevant or confusing information propagated to the augmented corpus's vectors, which would otherwise inhibit performance. On the contrary, if we were to assume that most instances of perceptual information propagated to Text8 is relevant to the words they surround, one could have hypothesized that Technique

A would retrieve more relevant images for augmented Text8, than for plain Text8; each image's average vector comprises information from its caption words, which is purely perceptual information, so the information propagated to Text8 would naturally help. Since these are speculations, I still needed to work with all four corpus-technique combinations above. I also hypothesized that between two approaches, one using Technique A and the other using Technique B, the latter would, if anything, outperform the former. The reason for this is that Technique B filters words on the same basis that Technique A ultimately compares the vectors for a query term and images' averaged vectors. Therefore, Technique B appears, at the surface, to simply be applying this vector comparison a second time, and thus considering images at a higher granularity.

For any of these latter four approaches, when taking the average vector representation over an entire image, if one of the image's caption words does not have a corresponding word vector in the text corpus, then it is simply not included in the construction of the average vector. This allows, to some extent, the chance for these approaches to mistakenly rank an image as overly relevant to a query term because of its average vector, even when it contains many caption words that were discounted because of lack of their vector representations. However, given the nature of how the ESP-Game is played, which is also how the images' captions were generated, these discounted words should not have affected our results greatly. That is, ESP-Game participants were required to annotate images, each with a single tag, within a short period of time. Image 1 illustrates how annotators generally tag images with the most obvious, concrete descriptors, *i.e.*, the most imageable, in the case of images. Furthermore, past studies have shown that concrete terms have relatively narrow meaning when compared to abstract terms [1]. Therefore, we can assume that any words we discount, due to their lack of a corresponding semantic vector, represent only a narrow visual aspect of the overall image. Conversely, if the vectors for an image's remaining caption words produce an average vector that is very similar to the query term's, then by definition of concreteness, the visual aspects of that image that pertain to the remaining caption words should appear clearly to participants when they view the image, such that they can appropriately rank the image's relevance to the query term.

5.2 Query Terms & Corresponding Images

I used 32 query terms in my experiment, of varying concreteness and imageability, as was necessary for testing the effectiveness of my different image retrieval approaches. Half were nouns and half were verbs, because while nouns are generally considered more concrete than verbs, there was considerable variation in concreteness and imageability rankings, within the total set of nouns and verbs each. While the USF datset provided concreteness rankings, it did not have imageability ratings. The Bristol Norms dataset, however,

does provide imageability ratings, so I chose words from that dataset, that also occurred in the intersection of Text8 and USF's vocabularies. Note that whereas USF's concreteness ratings range from 1 to 7, Bristol's imageability ratings range from 100 to 700. To extract my final set, I first grouped this list by part of speech, into 16 nouns and 16 verbs, to avoid the risk of then just taking all of the most concrete words I would have to use for this experiment, based on some absolute upper imageability bound. That is, whereas the upper concreteness bound for nouns is nearly maximal (7), the upper bound for verbs is significantly lower, in which case, were I to try to retrieve all highly concrete words in general, at once, I would have too many nouns and not enough verbs, if any. Therefore, a "concrete verb," or an "imageable verb" here refers to a verb whose concreteness or imageability rating is high, relative to the other verbs included in this dataset. The same holds for nouns. After grouping the total set by POS, I sorted each subset of 16 by concreteness, as per the USF dataset. I then grouped each set of 16 into more concrete and less concrete batches, so that I now had subsets of eight words each, which comprised, respectively, abstract nouns, concrete nouns, abstract verbs, and concrete verbs. Finally, within each set of eight, I sorted the terms by their imageability ratings in the Bristol dataset, so that I could then group each subset of eight further, into groups of four. This allowed me to focus on subsets representative of a unique combination of POS, concreteness relative to the POS, and imageability relative to the POS. Table 8 in appendix A shows my final set of query terms, sorted and partitioned in this way.

In applying each of my five approaches to each of these terms, I limited the number of retrieved images I would include to 25. Then, for each term, I took the union of the images retrieved by each approach. Therefore, if each of the five approaches returned at least 25 images, and there was no overlap between the approaches' results, then a term's union would contain 125 distinct images. However, I estimated there would be at least some overlap between the approaches' results for a given term. It turns out that each of the experiment's terms's results did have overlap, as the average number of images in the unions of the 32 words is 67.5. While none of these unions contained duplicate image URLs, there were some instances of two different image URLs presenting the same thing, within the same union. That is, what they presented was different only in size. In each case of this, I removed duplicates so that only one of them remained.

5.3 Data Comparison & Interpretation

During the experiment, when a participant rated an image's relevance, that rating contributed to the data I needed to then evaluate the performance of the approach that returned that image. Most commonly, to evaluate a single approach's performance, I took the mean average precision (MAP) of subjects' ratings of the images that approach returned. That is, I took the average precision of the ratings of each word's set

of images that were returned by the given approach, and then I took the mean of those average precisions. This is a popular way of evaluating information retrieval systems. However, each MAP is derived from multiple average precisions, each of which is to be calculated from the "hits" and "misses" in a set of retrieval results; that is, it is meant to only take binary retrieval results as input. As described in section 5.4, when one of the eight forms had recorded multiple participants' responses, I needed to somehow combine their responses to a given image, to produce a binary response value, to enable the use of the MAP.

Alternative to the MAP, and thus alternative to combining multiple subjects' responses to produce binary values, I considered taking the simple mean of multiple people's responses to the same query result. By taking the mean of each result in that form, I could still evaluate each approach's performance, but with more discrete, floating-point response values. While MAP is ideal for holistic evaluations, *i.e.*, for an approach's performance on all 32 words, I also wanted to also evaluate how a given approach performed on the much smaller, four-word groupings, specified in table 8. So while I still tried taking the MAP of each approach's performance on one of these smaller groupings, I also tried to utilize the more discrete average values provided by just taking the simple mean over multiple people's responses. Then, I took the average of this averaged responses. Results from these approaches are shown below, in section 6.

By taking the MAP or the average of averages of the responses to approaches' retrieval results, we are able, to some extent, to quantify and compare the approaches' performance. However, to test for significant gaps between the different approaches' results, I used either the paired or independent two-sample T-test. When comparing approaches in this way, I looked for results with a p-value below the threshold of 0.05.

5.4 Data Collection & Amalgamation

In order for human subjects to be able to rate images' relevance within a reasonable span of time (*i.e.*, 20 minutes), so that they would not get tired and provide faulty results, I partitioned each of the 32 words' image retrieval unions into eighths. That is, for each of the 32 words, I took an eighth of its union's images, so that each of the eighths contained an arbitrary number of images returned by each of the five approaches. Then, I took one eighth from each of the 32 words' unions, and combined these into an online form, to track the subjects' ratings. This resulted in eight unique forms, because no two forms contained the same eighth from any of the 32 words' unions. Each form comprised 32 pages, one for each term, where each page contained an eighth of the images taken from the page's corresponding query term's union. Forms were filled out between human subjects, *i.e.*, each subject filled out only one form. In each form, participants were asked to rate the relevance of each image, based on its relevance to the current page's corresponding query term. That is, they were asked to choose, below each image, one of "Relevant," "Semi-relevant," and

"Not relevant." Including "Semi-relevant" allowed participants to have more freedom in considering each image's relevance. It also accounted for the possibility of a given query term having multiple meanings, some more prevalent or commonly used than others. Then, if some images represented these lesser-used interpretations of the term, then participants would not have to choose whether to constrain themselves to only the most common meaning.

I gathered responses from 20 human subjects in total, over the eight forms. This resulted in the number of responses per form ranging from one to four, such that I needed to combine multiple participants' responses, for one of my eight forms, into a single set of responses. Doing so allowed me to combine, for each of the 32 terms, single responses to their retrieved images over the eight forms, so that I could then examine and compare images' relevance ratings homogeneously. I therefore needed different ways of amalgamating multiple responses to a given form, given the specific number of responses to that form. The pseudocode for COMBINE-RESPONSES in appendix B illustrates this. The idea was that any time a participant rated an image as either "Relevant" or "Semi-relevant," it counted as a hit, and any time they rated an image "Not relevant," it counted as a miss. By interpreting the data binarily in this way, I was able to combine each form's multiple responses, and then group the responses over all eight forms by query term. For each term, I could then take the average precision over the responses to its corresponding union of images, to finally calculate the MAP over the entire set of images returned for the current word, by a given retrieval approach. As indicated above, in section 5.3, I could also have simply averaged multiple people's responses to the same query results, when taking the MAP was not the goal.

Given one of the eight forms, and a question in that form, I associated a numeric value with each of the form's participants' responses to that question. That is, if one of the participants responded with "Relevant," then I interpreted that single response as having the value 1; similarly, I interpreted "Semi-relevant" as 0.5, and "Not Relevant" as -1. The decision to count "Semi-relevant" responses as positive, or a hit, may seem arbitrary, given that "Semi" indicates it is only halfway between relevant and otherwise. However, whereas images assigned "Not Relevant" could easily be interpreted as being actively irrelevant to the given query term, the same could not be said for images assigned "Semi-relevant." A "Semi-relevant" image must be somewhat relevant to the query, which means it either partially represents a concept that is clearly associated with the query term's, or it represents something closely related to one of the query's meanings. In the former case, the image belongs in the results, trivially. The latter case, meahwhile, is also useful for gauging the query's semantic representation, especially when the query is more abstract, because that "Semi-relevant" image is then accounting for the specific, perhaps more concrete aspects of the term's meaning. So in a form, given the form's set of participants, if the sum of the numeric values corresponding to their relevance ratings of an image is nonnegative, then COMBINE-RESPONSE produces a nonnegative

amalagamated response; otherwise, that single response is negative. Since we are producing binary hits and misses, that "nonnegative" response will be taken as a positive response, counted as a full hit. We do this because in the edge case where the sum of the participants' responses is exactly 0, *i.e.*, a "draw" between relevance and irrelevance, we can conclude the amalgamated response is vacuously relevant, since it cannot be not relevant, since 0 is nonnegative.

6 Results & Discussion

6.1 Evaluating Approaches on Entire Query Set

After collecting all 20 participants' responses, I combinined multiple responses to the same form into single binary values, via COMBINE-RESPONSES, and then grouped these binary ratings by their corresponding query term. Then, for one of my given approaches, for each of the 32 terms, I calculated the average precision of the binary ratings pertaining to those images retrieved by the current approach, for the current query. Taking the mean of these 32 average precisions produced the MAP of the current approach. Figure 7 shows these MAP results; there is clearly a steady decline in performance as we include more information in building semantic vectors. Table 4 shows the most meaningful results of having run a dependent paired T-test on these values. Running Technique A on plain Text8's vectors retrieved significantly more relevant images than does running Technique A on augmented Text8's vectors; likewise, running Technique B on plain Text8 retrieved significantly more relevant images than does running Technique B on augmented Text8. In both cases, the pure text corpus performed better, regardless of whether we filtered the images we considered for retrieval. In fact, between approaches that used the same iteration of Text8, there was no significant difference in performance. This indicates that the decision of whether to propagate perceptual information was a serious bottleneck for creating representative semantic vectors, much more than was the decision to filter which images to consider for retrieveal, as per Technique B. Specifically, propagating perceptual information to Text8 seems to have created semantic vectors that were not comparable across modalities. That is, since each of the four DSMs retrieve images on the basis of the cosine similarity, between a query's vector and an image's averaged vector, propagating perceptual information to Text8 seems to have inhibited the accuracy of the vector representations, of words in the text corpus or of the images' caption words.

If we continue to assume that by nature of captioning, an image's caption words tend to encapsulate the most intuitive or prevalent (at least visual) aspects within the image, then a given image's caption words

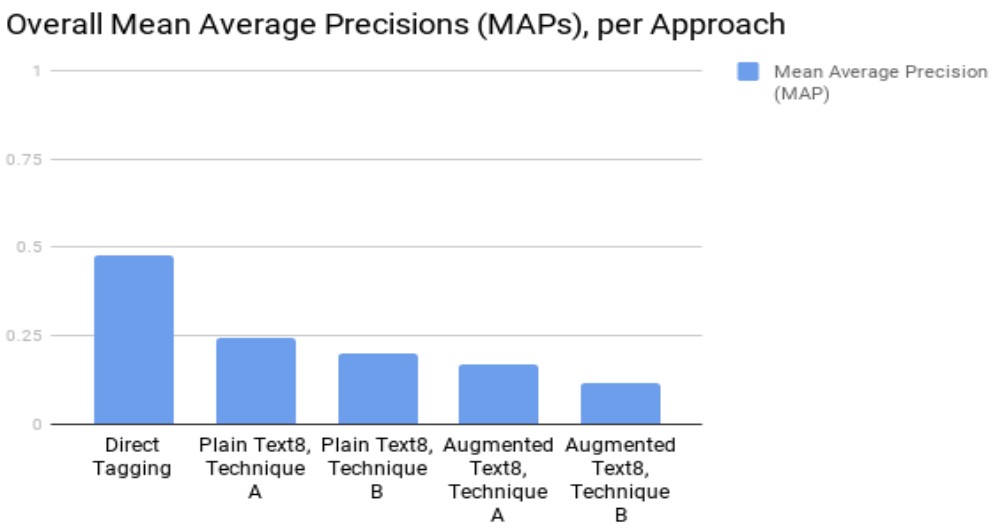

Figure 7: Mean average precisions (MAPs) of each approach, over all 32 words.

would most likely be similar, or related to each other. Therefore, these caption words' vectors should ideally have been similar to each other; this rules against the possibility that two totally unrelated caption words should mistakenly influence the comparison of the query's vector to that image's averaged vector. Furthermore if, during a DSM's propagation step, pseudo-sentences were to be inserted, they would have comprised caption words from images, such that each of these images would be tagged by a word appearing in the original text. If one of the query terms were to appear near one of these caption words in the original text, then by the Distributional Hypothesis, these words would appear in a similar context, and thus should have similar meanings. But by similar logic, if a DSM were to insert pseudo-sentences for the latter term, then the query term would now be surrounded by other words that share a similar context with the original caption word; the Distributional Hypothesis would thus suggest that the query term should also have similar meaning to these terms. At the same time, while the query term and the words in the pseudosentences share a common context, *i.e.*, they appear near the caption word, the modes of these two contexts are totally different, and the Distributional Hypothesis may not be able to account for this.

6.2 Caption Words vs. Non-caption Words

Direct tagging significantly outperformed all of the other four approaches, as shown in 4. If we again assume, by the nature of captioning, that when someone annotated the retrieved images with the query term, the term clearly represented a prevalent visual or otherwise perceptual aspect within the image. This, in addition to direct tagging's superior performance in the experiment, suggests that images retrieved via di-

t	Significance (two-tailed)	Approach 1	MAP 1	Approach 2	MAP 2
3.2503	0.0027	Plain Text8, Technique A	0.2429	Augmented Text8, Technique A	0.1677
3.5929	0.0011	Plain Text8, Technique B	0.2022	Augmented Text8, Technique B	0.1161
2.9371	0.0074	Direct Tagging	0.4763	Plain Text8, Technique A	0.2429
3.0551	0.0056	Direct Tagging	0.4763	Plain Text8, Technique B	0.2022
4.4311	0.0002	Direct Tagging	0.4763	Augmented Text8, Technique A	0.1677
5.3553	1.9406e-05	Direct Tagging	0.4763	Augmented Text8, Technique B	0.1161

Table 4: Significant results from dependent (related) paired T-test.

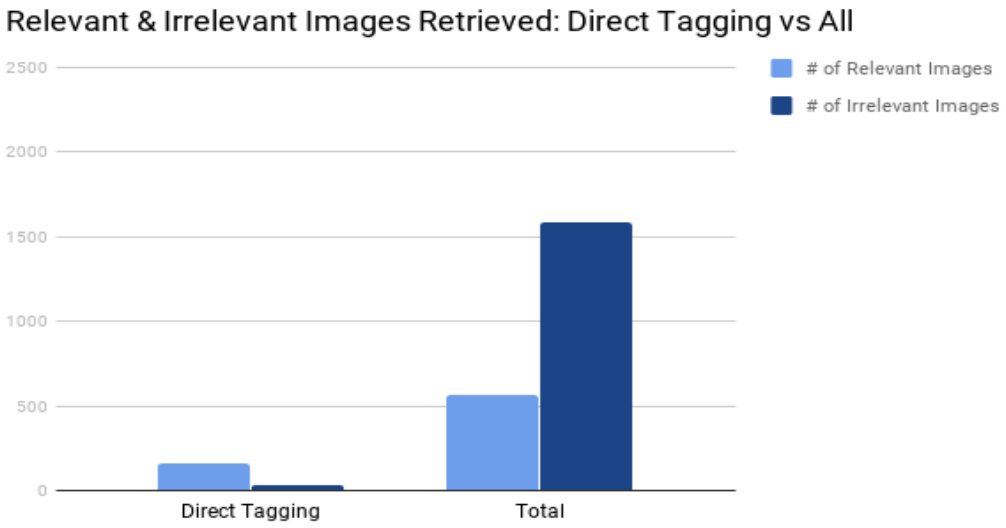

Figure 8: Relevant/irrelevant retrieval ratios, between direct tagging and all approaches.

t	Significance (two-tailed)	Approach	Tagged MAP	Untagged MAP
3.8681	0.0019	Plain Text8, Technique A	0.6868	0.3087
3.3459	0.0036	Plain Text8, Technique B	0.6346	0.2914
2.7062	0.0146	Augmented Text8, Technique B	0.4851	0.2165

Table 5: Significant results from independent paired T-test.

rect tagging can be viewed as a benchmark for image retrieval. Figure 8 further indicates its superiority to the DSMs: of the 562 total hits, across all five approaches, 153 were retrieved via direct tagging. Over all 32 terms, of the 562 hits over all five approaches, 157 came from direct tagging, which means direct tagging's recall was approximately 28%. The fact that its recall is this high, even when the total number of images retrieved via direct tagging is so much smaller than that of the other approaches, further emphasizes its effectiveness. However, its main drawback is that it only retrieved images for 24 of the 32 terms. This illustrates the limitation of human image annotation: humans are only likely to tag images with words of a certain kind. Table 8 shows, for instance, that save for the most abstract words in the experiment's set, most words directly tag less than five images.

Since captions generally comprise more imageable words, I wanted to test whether there was a correlation between an approach's performance on a query, based on whether or not the query term tagged any images in the datast directly. In table 8, we see that the words highlighted in blue tag zero images, which suggests they are less easily associated with an image, *i.e.*, less imageable, and perhaps less concrete overall. Therefore, I compared each of the DSMs' approaches, between two disjoint subsets of the 32 words: the 24 words that tagged a nonzero number of images, and the other eight words. The approaches' MAPs over these two subsets are shown in figure 9. Then, since these two subsets were mutually disjoint, they were independent of each other, so I ran the independent paired T-Test between the average precisions comprising each of their MAPs, for each of the four DSMs. Table 5 shows the significant results. Unsurprisingly, three of the four DSMs performed significantly better on the set of 24 tagging words than the set of eight non-tagging words. Even though the remaining DSM, with augmented Text8 and Technique A, did not show significant difference, the MAP over the tagging words was still higher than that of the non-tagging words. These results indicate that regardless of the amount of perceptual information we propagate with a DSM, much of the DSM's success is likely to be incident with the query term's imageability. This further suggests that direct tagging should be treated as a retrieval benchmark, despite the fact that it does not apply to many of the query terms in this experiment.

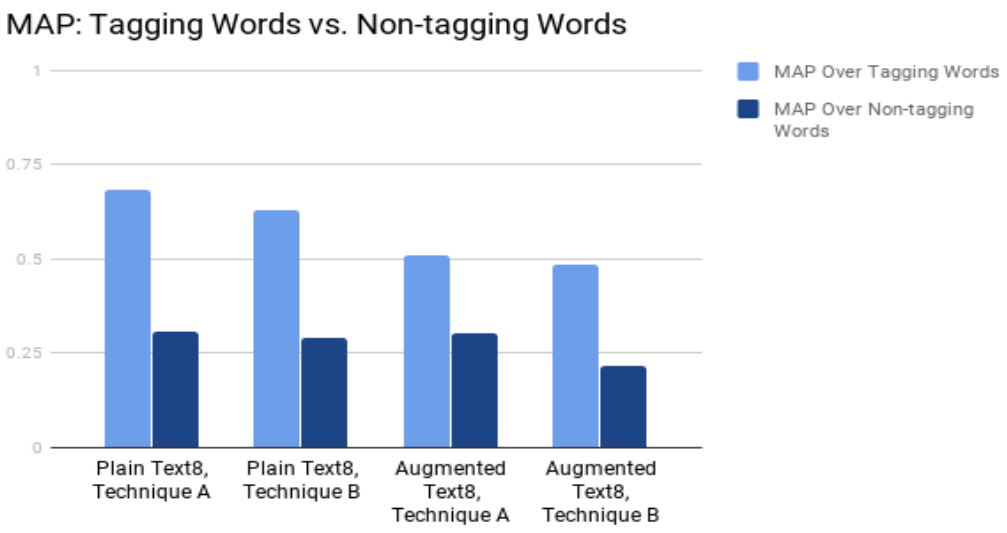

Figure 9: MAPs of each DSM's result ratings, when run separately on the subset of words that each directly tag an image in the dataset, and the subset of words that each tag zero images.

6.3 Performances on Subsets of Words

At a higher granularity, I looked at each approach's performance, on each group of four words in the total query set, as specified in table 8. One method I used for analyzing this per-group performance was the MAP, as before. Given a group of four words, and given one of the approaches, I took, for each word, the average precision of the ratings of the images returned for that word, by the given approach. Then, I averaged those four word's average precisions. Figure 10 presents each group's MAP, for each of the four DSMs. To test for significance, I ran dependent paired T-tests on the average precisions over the results returned by different approaches, but for the same group of four words. Table 6 shows that plain Text8 with Technique A retrieved significantly better results than augmented Text8 with Technique A, for the group of abstract, imageable nouns and the group of abstract, non-imageable verbs. The table also shows that plain Text8 with Technique B retrieved significantly better results than augmented Text8 with Technique B, for the group of abstract, imageable nouns. So for both of the DSMs that propagated perceptual information to the text, image retrieval for the more abstract groups of words was inhibited. This is consistent with Kiela et al. [8]'s reasoning to moderate the amount of perceptual information they added, based on the concreteness of the current word in the text.

While table 6's results were consistent with past results, I was not sure that taking the MAP over only four words' average precisions would provide an effective analysis. That is, the average precisions that make up their MAP is derived from binary values, which are inherently more limited in the amount of

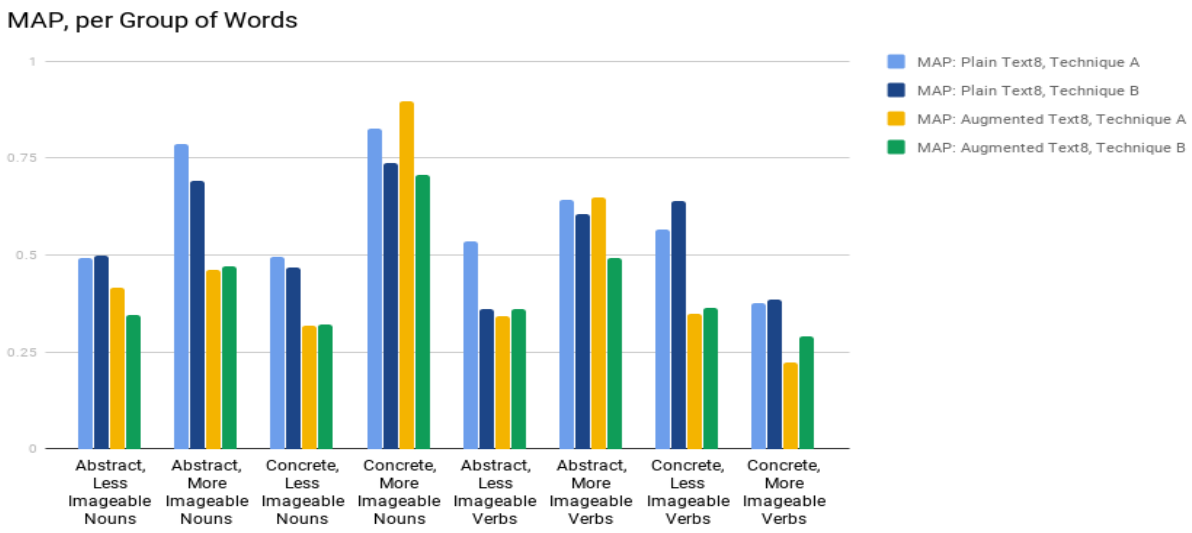

Figure 10: MAPs of the DSMs' retrieval result ratings, for different subsets of the 32 query terms, each containing containing only four words.

t	Significance (two-tailed)	Approach 1	MAP 1	Approach 2	MAP 2
4.2507	0.0239	Plain Text8, Technique A	0.7872	Augmented Text8, Technique A	0.4628
8.6622	0.0032	Plain Text8, Technique A	0.5592	Augmented Text8, Technique A	0.3380
4.0507	0.0271	Plain Text8, Technique B	0.6901	Augmented Text8, Technique B	0.4710

Table 6: Significant results from dependent paired T-test.

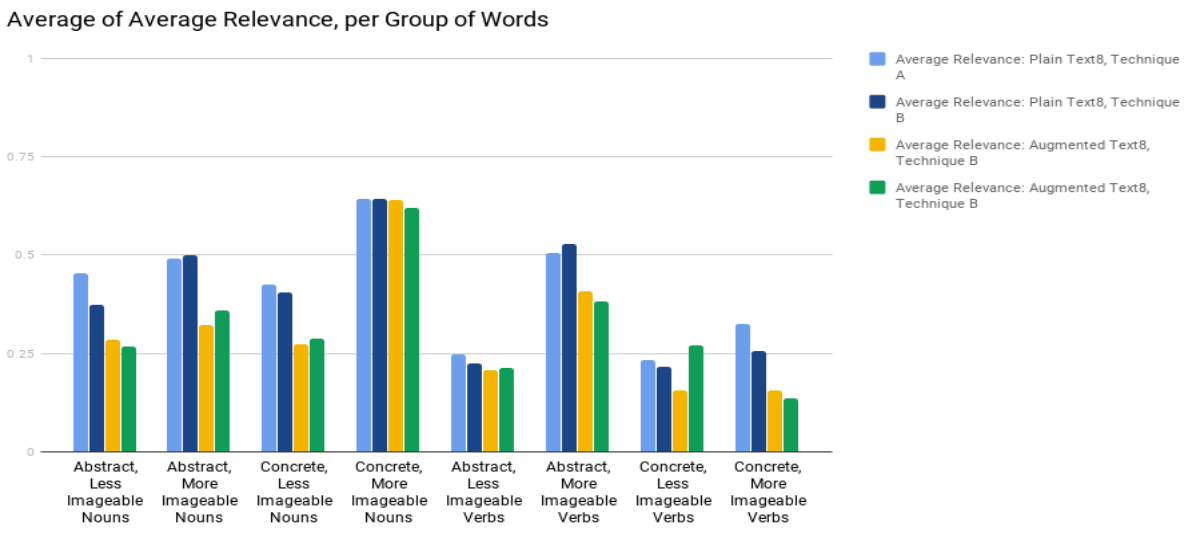

Figure 11: For each DSM, the average of averages of its retrieval result ratings, for different subsets of the 32 query terms, each containing containing only four words.

information they carry. This, on top of the small number of query terms per group, suggested that I should consider alternative evaluation techniques. As explained in section 5.3, an alternative method of analysis is to simply average, for each word in the group, the relevance ratings of the images returned by the current approach. The reason the averages do not comprise only binary values is that instead of combining multiple participants' responses via COMBINE-RESPONSES, as with average precision, you here simply take the average of the multiple responses; this produces a more discrete, floating-point value for each image retrieved by the current approach. Figure 11 shows the corresponding results. After applying a paired dependent T-test between each set of averages, taken over the results returned by different approaches, but for the same group of four words. These tests showed that between the four DSMs, none performed significantly better than any other. From these two evaluation methods, then, DSMs that utilize perceptual information propagation perform, at best, only as well as their text-only counterparts. Note that with each of these evaluation methods, I did not analyze the results from direct tagging, because several of these groups would include words that tag little or not words. Therefore if, for a group of four words, some of the words did not retrieve any images via direct tagging, then comparing direct tagging's results for that group, with the results from one of the DSMs for that group, would be equivalent to comparing two different, unequal groups.

t	Significance (two-tailed)	Approach 1	MAP 1	Approach 2	MAP 2
5.7045	1.3950e-08	Plain Text8, Technique A	0.4109	Augmented Text8, Technique A	0.3041
3.7351	0.0002	Plain Text8, Technique B	0.3918	Augmented Text8, Technique B	0.3209

Table 7: Significant results from independent paired T-test.

6.4 Alternative to MAP: Mean Over Total Results

As in section 6.3, I wanted to account for the possibility that using discrete, rather than binary values, would provide different information about the DSMs performance. Therefore, I took the average over all the words retrieved by a given approach. And as in section 6.3, I combined multiple participants' responses to a single question by simply averaging their values. The averages are shown in figure 12. I then ran the independent, rather than dependent paired T-test, between different approaches' entire sets of individual ratings. Even though I was including, in my calculation of each DSM's average, the images retrieved for all 32 terms by that DSM, the average was calculated over all images' individual ratings at once, rather than over the average of each term's images' ratings, as was done in 6.3. Therefore, the sets of values being compared here were not at all divided by the individual terms to which they corresponded, making for much more heterogeneity between two approach's overall averages.

Table 7 shows, as in 6.1, that the approaches that used the plain Text8 corpus signficantly outperformed the approaches that used the augmented corpus. So even when using different methods of analysis, the results are the same, which only further supports the notion that propagating perceptual information to text inhibits our a system's ability to retrieve relevant images. Furthermore, as in all of the above results, filtering which images to consider, based on whether they are tagged by words in the text with vectors similar to the query term's, indicates no significant impact on the relevance of retrieved images. All results thus indicate that given a query term, when we found words in Text8 whose vectors were most similar to its own, we could afford to filter the images based on whether they were tagged by those similar terms. The fact that filtering did not cause us to choose significantly less relevant images, suggests that vector comparison between two words in the same textual context is still effective; this is consonant with the Distributional Hypothesis. However, differences in results between DSMs that used different iterations of Text8 suggest that the Distributional Hypothesis is limited, at the very least, to same-context vector comparison.

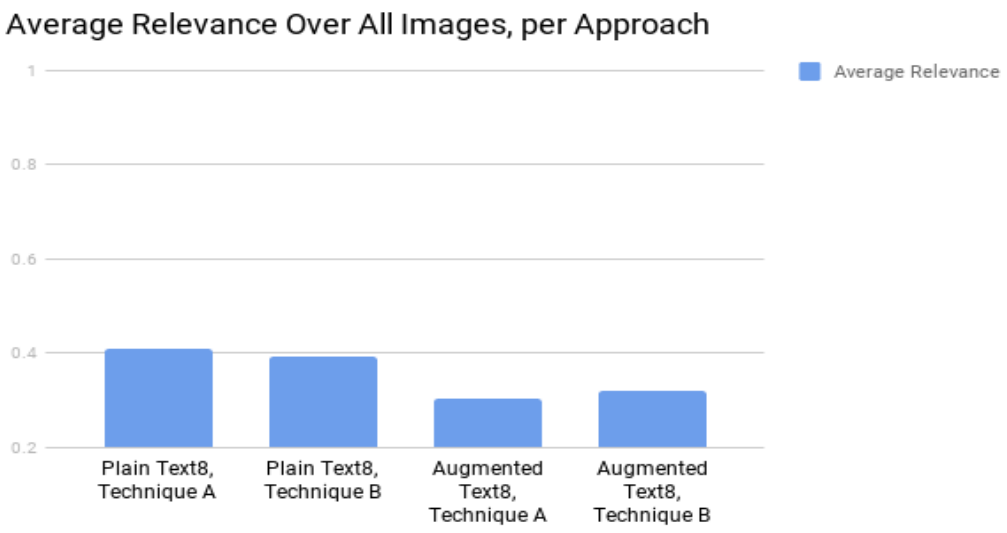

Figure 12: The DSMs' averages, over all of their individual retrieval result ratings.

7 Conclusion & Future Work

In this study, I implemented an image retrieval system. I used various distributional semantic models to identify images whose caption words had the most similar or relevant meaning to the query. That is, I utilized semantic vectors to represent and compare word's meanings, and thus retrieve the images whose caption words' average vectors were most similar to the query term's vector. I considered propagating perceptual data from datasets of captioned images and annotated words, to improve these vector representations by providing multi-modal, rather than purely textual information; I also tried filtering the images I considered for comparison, based on whether they were tagged by words whose vectors were similar to the query term's in the original text. Results showed that in general, propagating perceptual information inhibited the system's ability to retrieve relevant images; filtering by itself, meanwhile, had no significant impact. This suggests that across modes of representation, distributional semantic techniques are somewhat limited, even when we account for the different modes by combining their information.

Given the multi-modal approaches' worse performance, future work should follow in Kiela et al. [8]'s footsteps: focus on moderating the amount of perceptual information we propagate for a textual corpus's vocabulary word, based on the word's concreteness. The results in 6.3 show, in particular, that abstract or less imageable terms are more likely to be affected negatively by the propagation step. Therefore, we may have been inserted too much perceptual information into Text8 upon encountering these abstract query terms. Other work should focus on image retrieval for words of other parts of speech, and especially those

that are generally even more abstract than verbs, *e.g.*, adverbs. Improvements in these words' retrieval results would indicate the greatest improvement in our ability to semantically represent them.

Appendices

A

B

COMBINE-RESPONSES()

```
1  forms = { form — one of the eight response forms }
2  for form ∈ forms
3
4      Responses_collection = { responses — one participant's responses to the current form }
5      img_index = 0
6      for image ∈ form
7
8          hits = 0
9          for responses ∈ Response_collection
10             Here, "Relevant" adds 1, "Semi-relevant" add 0.5, and "Not Relevant" add -1 to hits
11             hits+ = responses[img_index]
12         if hits ≥ 0
13
14             Record single response across participants as positive, or "Relevant"
15         else
16             Record single response across participants as negative, or "Not Relevant"
17     img_index+ = 1
```

References

[1] Lawrence W Barsalou and Katja Wiemer-Hastings. "Situating abstract concepts". In: *Grounding cognition: The role of perception and action in memory, language, and thought* (2005), pp. 129–163.

Term	Concreteness	Imageability
Abstract, Less Imageable Nouns		
norm	2.18	142
expense	2.77	160
custom	2.99	166
concept	1.97	197
Abstract, More Imageable Nouns		
silence	3.09	413
chaos	2.50	426
hazard	3.38	459
demon	2.56	533
Concrete, Less Imageable Nouns		
roach	6.42	365
creek	5.95	378
nylon	6.16	415
jury	6.17	426
Concrete, More Imageable Nouns		
airport	6.31	650
bacon	6.46	650
tractor	5.86	655
leaf	5.89	655
Abstract, Less Imageable Verbs		
become	2.66	105
allow	2.64	170
restore	2.71	178
prove	2.54	221
Abstract, More Imageable Verbs		
choose	3.00	239
amuse	3.17	255
plead	3.08	265
send	3.08	274
Concrete, Less Imageable Verbs		
weigh	3.54	384
grind	4.37	390
argue	3.23	395
spell	3.49	429
Concrete, More Imageable Verbs		
tickle	4.69	450
knock	5.09	460
bake	4.76	481
marry	3.41	498

Table 8: The 32 words used in this corpus. Note that the words highlighted in red directly tag $0 < n < 5$ images in the dataset; the words highlighed in blue tag zero images.

[2] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. "Multimodal distributional semantics." In: *J. Artif. Intell. Res.(JAIR)* 49.2014 (2014), pp. 1–47.

[3] Luana Fagarasan, Eva Maria Vecchi, and Stephen Clark. "From distributional semantics to feature norms: grounding semantic models in human perceptual data". In: *Proceedings of the 11th International Conference on Computational Semantics*. 2015, pp. 52–57.

[4] Yansong Feng and Mirella Lapata. "Visual information in semantic representation". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2010, pp. 91–99.

[5] Stevan Harnad. "The symbol grounding problem". In: *Physica D: Nonlinear Phenomena* 42.1-3 (1990), pp. 335–346.

[6] Felix Hill and Anna Korhonen. "Learning Abstract Concept Embeddings from Multi-Modal Data: Since You Probably Can't See What I Mean." In: *EMNLP*. 2014, pp. 255–265.

[7] Dan Jurafsky and James H Martin. *Speech and language processing*. Vol. 3. Pearson London: 2014.

[8] Douwe Kiela et al. "Improving Multi-Modal Representations Using Image Dispersion: Why Less is Sometimes More." In: *ACL (2)*. 2014, pp. 835–841.

[9] Alessandro Lenci. "Distributional semantics in linguistic and cognitive research". In: *Italian journal of linguistics* 20.1 (2008), pp. 1–31.

[10] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[11] Alex Minnaar. *Word2Vec Tutorial Part I: The Skip-Gram Model*. 2014. URL: `http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_I_The_Skip-Gram_Model.pdf`.

[12] Peter D Turney and Patrick Pantel. "From frequency to meaning: Vector space models of semantics". In: *Journal of artificial intelligence research* 37 (2010), pp. 141–188.

[13] *Vector Representations of Words*. https://www.tensorflow.org/tutorials/word2vec. Accessed: 2017-11-14. URL: `https://www.tensorflow.org/tutorials/word2vec`.