REMOVAL OF AMBIGUITY OF NOUN USING MULTIMODAL APPROACH

Mrs. Arpita Dutta
Department of Computer Science and Engineering,
National Institute of Technology Silchar, Assam, India
*Corresponding author email: duttaarpi@gmail.com

Mr. Salam Michael Singh
Department of Computer Science and Engineering,
National Institute of Technology Silchar, Assam, India.
Email: salammichaelcse@gmail.com

Dr. S. K. Borgohain
Department of Computer Science and Engineering,
National Institute of Technology Silchar, Assam, India.
Email: samir@cse.nits.ac.in

Abstract:

It is truly amazing that human beings can easily understand and comprehend the intended meaning of an ambiguous word. The meaning of an ambiguous word differs with its usage in a different context. Still, human beings can Fig. out the meanings with ease. We have machine translation (MT) systems that can translate from a source language to its equivalent target language. The main intension of these MT systems is to seamlessly transfer the intended meaning of the source text to its target text. But due to ambiguous nature of natural language, MT systems do suffer from setbacks. Word sense disambiguation (WSD) is one of the greatest challenges to overcome. The researchers have contributed a number of WSD algorithms that operate over textual data. These algorithms were primarily developed to disambiguate an ambiguous word or to determine the exact meaning of an ambiguous word based on the context. It seems that context plays a decisive role while disambiguating an ambiguous word. A section of the research community are of the opinion that the neighbouring words that appear along with an ambiguous word in a sentence might help in finding the meaning or sense of an ambiguous word. This is commonly known as distributional semantics. In this paper, we have proposed a novel technique to remove the ambiguity of polysemous noun using a multimodal distributional semantics model (MDSM). The arduous task was to find a standard multimodal database for carrying out our desired experiments. This was compensated by using ImageNet database. ImageNet is a large-scale database containing tens of millions of annotated images organized by the semantic hierarchy of WordNet. Our MDSM exploits both the image features and textual features from the annotated images of ImageNet database. For both the training and testing purpose, we have used a total of 8 different synsets. The total no. of 800 related images to these synsets are used for training (each synset contains a reduced set of images i.e. 100 images). However, the total no. of images used for testing is 8 ((each synset contains 1 image) only. The 8 synsets that we have considered are {(bat: word, mammal: sense), (bat: word, cricket: sense), (bass: word, guitar: sense), (bass: word, fish: sense), (mouse: word, animal: sense), (mouse: word, device: sense), (bank: word, piggy: sense), (bank: word, river: sense)}. These synsets are carefully crafted from voluminous ImageNet database so that each of the synset represents a polysemous noun. The training phase generates two co-occurrence matrices namely (i) reduced weighted word-synset matrix of size |n * 6| where n=total of nouns and (ii) reduced weighted codeword-synset matrix of size |k * 6| where k=total of visual codewords. The value of k is not fixed but varies where k = 50, 200 and 400.Each noun that appears in the weighted word-synset matrix is a vector v_w. As per distributional semantics, neighbouring words that occur along with a polysemous word may help in disambiguation of a polysemous word. Keeping this mind, all neighbouring nouns that occur along with the polysemous noun vector in the weighted word-synset matrix are later used in the testing phase. Although the test data contains 8 annotated images related to the above-mentioned synsets, the textual data is entirely omitted out during testing; only the images are considered. By applying image processing algorithms, *m* no. of features are extracted out from each test image. The *m* no. of image features are assigned with a codeword label by measuring the Euclidean distance (nearest to the cluster centre) between the *m* image features and codewords. Since, test image features contain *m* number of codewords, so a single codeword vector cannot be used to represent an image vector. So, all the m no. of codewords are summed up to obtain a single vector v_i. To measure the semantic relatedness between the test image *I* and a word (noun) *w*, e.g. *chords*, we simply compute the cosine similarity between v_i and v_w. From the experimental results, we can draw the conclusion that our

proposed algorithm could disambiguate a polysemous noun with the help of neighbouring words (nouns) and image features. We may that our algorithm is based on distributional semantics and a joint semantic space of words (nouns) and images.

Keywords:**Word Sense Disambiguation, Ambiguous Noun, Visual Word, Semantic gap, Semantic Similarity.**

1. Introduction

Word Sense Disambiguation (WSD) is an open problem as well as one of the key task and most challenging and active research area of Natural Language Processing (NLP). WSD is considered an AI-complete problem that is a task whose solution is as hard as most difficult problem in artificial intelligence. There are many approaches to represent word meanings in context. But it is quite difficult to find a suitable representation of natural language which can be used in machine translation. The ambiguity is present in almost all natural languages spoken in the world. WSD is an open problem of natural language processing. Since, a picture is worth thousands words, a picture may convey an idea more quickly and effectively than the written word [Farhadi, M *et al.* (2010)]. Visualization always makes it easier to understand an idea better than only text. In our work, the surrounding words of an ambiguous word of a context, are achieved by measuring semantic relatedness between words and images where image is the representation of the ambiguous terms and its surrounding objects. If the semantic gap between words and images can be reduced then a model can be implemented using a joint semantic space for both words and images [C.W. Leong and R. Mihalcea (2011)]. The ambiguous word can be a noun or a verb or adjective. For example, the boy found a *bat* in his room where the ambiguous word *bat* may either refer to a cricket bat or a mammal. In the similar way, a *mouse* is in the box where the word *mouse* may be a computer mouse or an animal mouse. The meaning of the mouse may be understood if there is a context instead of a single sentence. Since, there are more surrounding terms in a context than a sentence which helps to understand the meaning of that particular word used in the context. Now, if these two examples are given as source text in a computer for machine translation, it is difficult to presume which meaning will be taken for translation. If exact meaning is not detected then the whole meaning of the sentence will be changed. So, there must be some method that will help machine to perceive the correct meaning of an ambiguous term in a context. Here, in our work, we are using multi-modal approach as because texts and images are individually ambiguous but together can remove the ambiguity. In unimodal approach, the previous example, ‘the boy found a bat in his room’, is very hard to find the meaning of ambiguous term *bat* in context even for a human being. Since there is a possibility that either cricket bat or mammal bat can be found in the room. This type of ambiguity is hard to resolve without any extra aid. Few researchers shifted the paradigm towards multimodal approach that is WSD using texts and images [K. Barnard and M. Johnson (2005)]. There are many approaches for WSD but sometimes it seems that only context words are not sufficient to remove ambiguity. In such cases, there is a need of extra aid to help disambiguation of polysemous word. In our work, for these circumstances, we have tried to use images as visual context to sentence containing ambiguous word and using multimodal approach to find exact meaning for ambiguous word. The multimodal approach can be split into two parts such as reducing semantic gap between images and texts and find out meaning of ambiguous noun word after defining a semantic relationship between image and text vectors.The major contribution of the research is summarized as follows:

A semantic bridge is established in between images and texts. Further, it is used for removing ambiguity of polysemous noun. Text documents can be represented in a vector space in which a semantic similarity or relatedness can be measured between words or linguistic terms. Word2vec [J. Searle (1980)] is a word embedding technique which is a vector space model for textual data in which words or phrases are mapped to vectors of real numbers. Here, we use a document as a bag of words (BOW).Now, BOW model is used here which is commonly used for information retrieval. In BOW, we count the number of times each word appears in a document which is the frequency of each word of the document and make a frequency histogram from it. Local features play a key role in object recognition as features are distinct across different objects or scene. Scale invariant Feature Transform algorithm is used for local feature extraction. A lot of research works has been performed to solve WSD problem. But none of those methods shows remarkable accuracy in removing ambiguity of word in context. In unimodal approach, the sense of the ambiguous word which maximizes the number of common words in the dictionary definitions of the given sense and surrounding words. But it depends on the calculation of the word overlap between the dictionary definition of the ambiguous word and its surrounding words. Since dictionary definitions are not very long so it fails to maximise the word overlapping. A multimodal approach which uses visual information as contexts to an ambiguous sentence is promising in Word Sense Disambiguation. But in some cases, the multimodal approach was not targeted to resolve ambiguity of text but was confined in finding out the semantic relatedness. Therefore, a novel multimodal distributional semantics model (MDSM) is proposed to remove the ambiguity of polysemous noun. The main contribution of the research work is summarized as follows:

- A novel technique to remove the ambiguity of polysemous noun using a multimodal distributional semantics model (MDSM) is proposed.

- The MDSM exploits two types of features that are extracted from a set of images and textual features from a standard text corpus.
- At training phase, the synsets and glosses are extracted from the annotated images.
- Then, the text data is pre-processed to remove the special characters, stop words and other irrelevant texts.
- Then, PCA is used to reduce the dimension of the generated word-synset document matrix. Finally, the term frequency inverse document frequency (tf-idf) of the reduced Word-Synset document matrix is calculated.
- At testing phase, first the images should be pre-processed to convert into gray-scale images. Then, the images are resized to reduce the computational complexity.
- Harris Corner detection algorithm is used to detect corners or edges of the pre-processed images and then Scale-Invariant Feature Transform (SIFT) is used to obtain more dominant features.
- The extracted feature vectors are then clustered into k-cluster centers using K-means clustering algorithm.
- Dimensionality reduction algorithm is applied on each generated codeword-synset document matrix. Finally, Tf-idf weighing scheme is applied on reduced codeword-synset document matrix.
- Two types of visual features such as global features and local features are then extracted.
- Finally, the performance of the proposed model is analyzed by using various evaluation metrics.

1.1. Organization

This paper is organized as follows-section2 explores related works. Section 3 unveils the methodology of the proposed system. Section 4 discusses dataset used for experiment. Section 5discloses the experimental environment and design. Section 6 discusses various results of the proposed system. Section 7 reveals conclusion and future work.

2. Related Works

WSD was considered as a distinct computational task of Machine Translation from the late 1940s [5].In those days, machine translation seems an impossible task. There are a rich variety of techniques have been researched from dictionary based methods to supervised machine learning method where classifier has been trained for each distinct word of a corpus, to completely unsupervised methods and also hybrid method. Among these, supervised machine learning approaches is the most successful algorithm to date. Dictionary based or knowledge based WSD requires knowledge sources like dictionaries or sense inventories to perform disambiguation. In this case, the most popular resource is Word Net thesauri [Hillel and Yehoshua (1960)]. Weaver, in 1949, first introduced the problem in a computational context. Later, Bar-Hillel in 1960 [K. Barnard and D. A. Forsyth (2001)] used the above example to argue that WSD could not be solved by computer as it requires in general to model all world knowledge. In the 1970s, WSD was a subtask of semantic interpretation system that was developed within the field of artificial intelligence. But, at that time WSD was mostly rule based and hand coded that suffer from a knowledge acquisition bottleneck. By the 1980s, large scale lexical resources like Oxford Advanced Learner's Dictionary of Current English (*OALD*),Word Net, became available that replaced hand coding by using knowledge extracted from those resources, but WSD was still dictionary based or knowledge based. In the 1990s, there was a revolution in WSD problem after applying supervised machine learning technique to solve it by considering WSD a paradigm problem. After that, in 2000s, satisfactory accuracy has been obtained by supervised techniques but it is difficult to solve in general since WSD is a problem which is present in almost all natural languages of the world. So, attention has shifted to semi-supervised, unsupervised corpus based system and combination of different methods and also knowledge based system using graph based method. However, supervised machine learning system continues perform best.

Much research work has been performed to solve WSD problem by different approaches of WSD using text. But none of those methods shows remarkable accuracy in removing ambiguity of word in context. So, a new idea is conceptualized based on the hypothesis that both image and text separately ambiguous, but together they remove the ambiguity effectively. For example, if someone introduces the difference between a bus and a truck to a child. If he or she provides some pictures bus and truck in the description, then it is more beneficial to that child to understand the meaning. This approach was first proposed by Kobus Barnard which dates back to Year 2001 [M. Lesk (1986)]]. All these approaches have been elaborated in following paragraphs given below.

2.1. Unimodal Approach

In this approach of WSD only the text features are used. In this instance, pioneer work is classical Lesk algorithm [Turdakov (2010)] which is based on the assumption that the ambiguous word and its contexts refer the same sense [S. Banerjee and T. Pedersen (2002)]. This algorithm used the dictionary definition as a knowledge source to disambiguate polysemous noun. It finds the overlapping between the dictionary definitions of the different senses of the ambiguous word and the surrounding words for that sentence. The sense of the ambiguous word maximizes the number of common words in the dictionary definitions of the given sense and surrounding words. But it depends on the calculation of the word overlap between the dictionary definition of the ambiguous word and its surrounding words. Since dictionary definitions are not very long so it fails to maximise the word overlapping [K. Barnard *et al.* (2001)]. Later, the original LESK algorithm has improved by introducing the

“Extended Gloss Overlap Measure” which as the name suggests increased the glosses (dictionary definitions) of the words to be compared by using Word Net [Hillel and Yehoshua (1960)] synset. Word Net is a computational lexicon of English which stores sets of synonym in a hierarchical order. Word Net contains 1, 55,000 words organized in over 1, 17,000synsets. Each synsets contains a brief definition (gloss).

2.2. Multi-modal Approach

From the literature survey, we have found that multimodal approach was used by a few researchers notably. Barnard [M. Lesk (1986)] described approach of linking words with pictures using unsupervised machine learning approach for object recognition in 2001.In the same year, in another work of Bernard[K. Barnard *et al.* (2005)] ,have highlighted that if texts and images are separately ambiguous, together they are not. The new approach was perceived for modelling multi modal data sets based on segmented images with associated text. A number of models are reported on joint distribution of segmented image and text. Although, measuring the performances of those models was difficult as it was hard to decide whether word has been placed on the right region of the segmented image or not. In 2003, Bernard in his magnificent work, proposed a new method of predicting words for images that could be from image dataset with associated text. He developed a large dataset of images where each image has set of keywords that remove ambiguity from the interpretation of the images. The major limitation of this work was multiple keywords being polysemous in automatic image annotation. In 2005, Roberto Navigli [R. Navigli and P. Velardi (2005)] introduced a Knowledge base WSD algorithm Structural Semantic Interconnections (SSI).There are a set of choices of senses in SSI algorithm which also illustrate a semantic graph of interconnections that structurally clarify those choices. The major drawback of this approach was lacking of too much dependency on availability of general purpose knowledge. In another pioneering work of Bernard [K. Barnard (2006)], conveyed the idea of reducing the ambiguity in the domain of image and text. James [N. James and C. Hudelot. (2009)], in 2009, have proposed the abstract of using both semantic and visual knowledge for removing keyword disambiguation from semantic Image annotation. In 2010, a new approach was presented for determining score linking between an image and a sentence by two approaches such as illustration and annotation [Farhadi, M *et al*. (2010)]. In 2010, Borgohain and S. B. Nair [S. Borgohain and S. B. Nair (2010)] proposed a new translation system for people who are not well known to each other’s language is able to communicate through an intermediate language called pictorially grounded language (PGL).In this paper, both source and target are grounded by a common set of annotations and images. In the same year, Feng and Lapata [Y. Feng and M. Lapata (2010)] tried to identify the meaning of ambiguous term from visual and textual data using supervised learning method without computing the semantic relatedness between arbitrary pairs of word and image. In 2011, Leong et al. [C.W. Leong and R. Mihalcea (2011)] discussed the idea of finding out semantic relatedness of words and images by reducing the semantic gap between word and image through extracting information from visual data. Their new method found a score using joint semantic space of images and words. In the same year, Westerveld et al. [T. Westerveld (2000)], came up with the concept of a promising joint textual words and simple visual features extracted from news images using colours and textures. The authors used Latent Semantic Indexing, a method that uses co-occurrence statistics to uncover hidden semantics and showed that it is successful in both monolingual and cross lingual text retrieval which can be used for multi-modal and cross-modal information retrieval. But the work was only performed on newspaper data. Su et al. [Y. Su and F. Jurie (2011)], also in the same year, presented a mechanism to disambiguate polysemy of visual words in bag-of-visual-words model. As a visual word may have multiple meanings, they improved the performance of bag-of-visual-words model by using the semantic contexts to disambiguate these meanings. But their disambiguation task was focused on the visual code-words, not on the natural language text. Although multimodal approach was reported by Feng et al. [Y. Feng and M. Lapata (2010)], Leong et al. [C.W. Leong and R. Mihalcea (2011)] and Westerveld et al. [T. Westerveld (2000)], but their works was not targeted to resolve ambiguity of text but was confined in finding out the semantic relatedness. In a recent paper, Orkphol et. al[Orkphol, Korawit, and Wu Yang (2019)] has conveyed a method which map word2vector with a corresponding word embedding vector to construct both sense signature and the context to the sentence vector in various configurations. Each word sense has given a score using cosine similarity which is computed from these two sentence vectors that is sense signature and the context. If the score is not higher than the specific threshold, the score will be combined with the probability of sense distribution learned from the large sense-tagged corpus, SEMCOR [Mihalcea, R. (1998)] otherwise possible senses can be obtained from high scores. In 2020, Wang et al. [Wang, Yinglin et al. (2020)] has shown a naïve information retrieval method to retrieve documents from the Wikipedia and this document retrieval process has been validated using latest standard WSD dataset. The focus of this work is to imitate the way human disambiguates words using latent semantic factors and connections between senses.

3. Proposed Methodology

Text documents can be represented in a vector space in which a semantic similarity or relatedness can be measured between words or linguistic terms. The hypothesis of distributional semantics is that the words that are used and occur in the same context tends to exert similar meanings [T. Mikolov et al. (2013)]. The basic approach is to gather distributional information of high-dimensional vectors and define semantic similarity in terms of vector similarity [B.B. Rieger (1991) and Z.S. Harris (1954)]. Here, we use a document as a bag of words (BOW). In BOW, we count the number of times each word appears in a document which is the frequency of each word of the document and make a frequency histogram from it. Similarly, the same concept is used in bag of visual words (BOVW) which is mostly used in scene classification but image features are considered as "words" instead of words. Image features are unique pattern for any image. Now, features of image consists of key points and descriptors. Key points are "stand out" points of an image so no matter whether the image is rotated, shrink or expand, its key points will be always same. Descriptors are the description of the key point. Therefore, each image can be represented as a frequency histogram of features that are in the image. Later, we can find another similar images or predict the category of the image from the frequency histogram. The steps that are followed by creation of visual code words are feature detection and description and clustering of the feature descriptor. Scale invariant Feature Transform algorithm is used for local feature extraction which is introduced by Lowe [H.R. Kher and V.K. Thakar (2014), D.G. Lowe, (2004)]. The advantage of our proposed multimodal approach is that it does not require any sense tagged corpus.

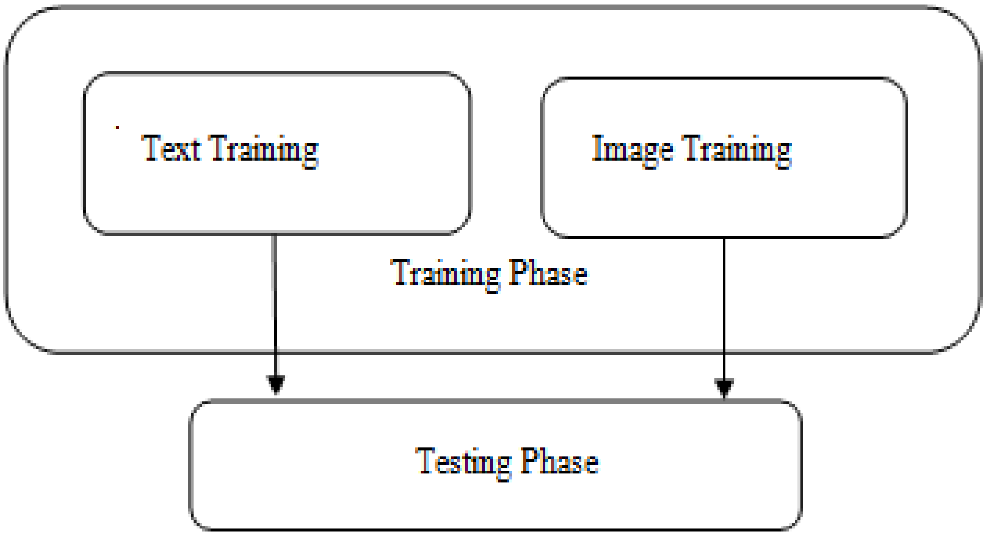

Fig. 1: The broad view of architecture of proposed model

The proposed architectural model can be divided into training phase and a testing phase. The training is carried out separately for both the text as well as images, which is illustrated in Fig. 1.The details of each of the components of the architecture are described in the rest of the sections.

Architecture of Proposed Model

The schematic diagram Fig.2 represents the architecture of our work. The proposed methodology is explained in the sub-sections below.

4. Dataset

Annotated images from ImageNet database were taken as both text and image dataset are required for our work. ImageNet database consists of 14,197,122 tagged images and 21,841 synsets ordered in WordNet hierarchy. In our work, we have used a small subset of Image Net database that is four pairs of ambiguous noun synsets or eight synsets. We have used synset pairs such as bat mammal and bat cricket, mouse mammal and mouse device, bass guitar and bass fish, bank piggy and Bank River. Each synset has 100 images that is a total of 800 images are taken for training data and 8 images foreight synsets as testing data. The whole work is divided into training and testing phase.

Fig. 3: Example of images for four pairs of noun synsets mentioned above

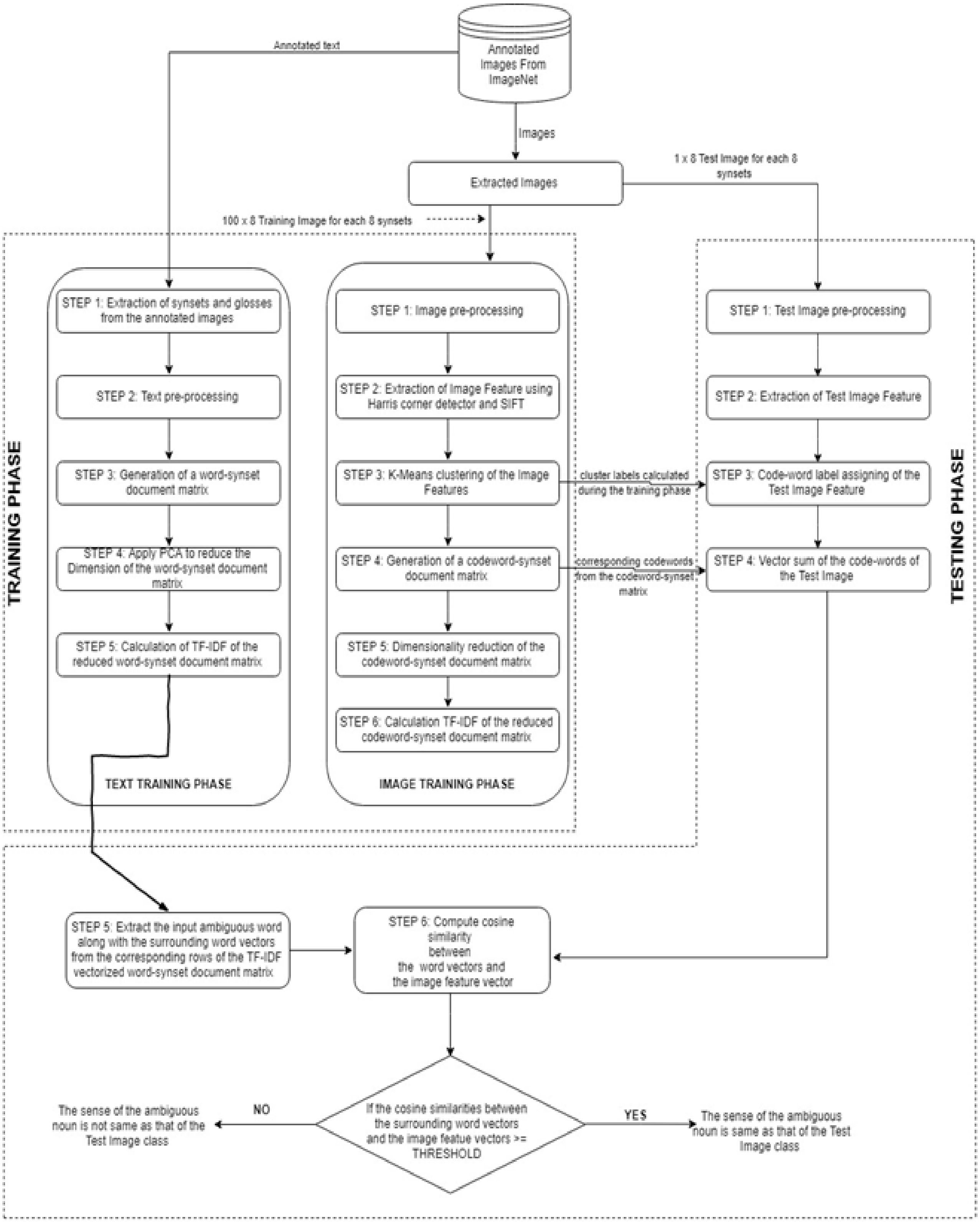
Annotated Images From ImageNet
Annotated text
Images
Extracted Images
1 x 8 Test Image for each 8 synsets
100 x 8 Training Image for each 8 synsets
TRAINING PHASE
STEP 1: Extraction of synsets and glosses from the annotated images
STEP 2: Text pre-processing
STEP 3: Generation of a word-synset document matrix
STEP 4: Apply PCA to reduce the Dimension of the word-synset document matrix
STEP 5: Calculation of TF-IDF of the reduced word-synset document matrix
TEXT TRAINING PHASE
STEP 1: Image pre-processing
STEP 2: Extraction of Image Feature using Harris corner detector and SIFT
STEP 3: K-Means clustering of the Image Features
STEP 4: Generation of a codeword-synset document matrix
STEP 5: Dimensionality reduction of the codeword-synset document matrix
STEP 6: Calculation TF-IDF of the reduced codeword-synset document matrix
IMAGE TRAINING PHASE
cluster labels calculated during the training phase
corresponding codewords from the codeword-synset matrix
TESTING PHASE
STEP 1: Test Image pre-processing
STEP 2: Extraction of Test Image Feature
STEP 3: Code-word label assigning of the Test Image Feature
STEP 4: Vector sum of the code-words of the Test Image
STEP 5: Extract the input ambiguous word along with the surrounding word vectors from the corresponding rows of the TF-IDF vectorized word-synset document matrix
STEP 6: Compute cosine similarity between the word vectors and the image feature vector
If the cosine similarities between the surrounding word vectors and the image featue vectors >= THRESHOLD
NO
The sense of the ambiguous noun is not same as that of the Test Image class
YES
The sense of the ambiguous noun is same as that of the Test Image class

Fig. 2: The architecture of proposed model to resolve the semantic gap between images and text

4.1. Training Phase

In the training phase, all the texts and images are pre-processed and features are extracted. It is further divided into two phases such as text training phase and image training phase.

Text training phase: This is the first phase of the proposed model. All the text related tasks like collection of glosses, pre-processing and transforming word to vector during training phase are described in the following sub-sections.

Extraction of synsets and glosses from the annotated images: We have extracted the synsets and glosses by using the annotation that is the synset id (Synset id is the unique id provided by Image Net). This mapping of the synset id and the glosses is available in the Image Net database. Glosses are the dictionary definition for the synset word. The following is an example of the synset id and synset word mapping.

Synset id	Synset
n02139199	bat, chiropteran
n03793489	mouse, computer mouse
n03935335	piggy bank, penny bank

Table 1: An excerpt table for synset id and synset mapping

In Table 2 the definition (gloss) for each synset of Table1 is given. After obtaining the mappings, we extracted all the text data for the synsets for our experiment. The raw text data are arranged according to their synsets.

Synset id	Gloss
n02139199	nocturnal mouse like mammal with forelimbs modified to form membranous wings and anatomical adaptations for echolocation by which they navigate
n03793489	a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad; on the bottom of the device is a ball that rolls on the surface of the pad; "a mouse takes much more room than a trackball"
n03935335	a child's coin bank (often shaped like a pig)

Table 2: An excerpt table of synset id and gloss mapping

Text Pre-processing: The text data presented in the Table 2 containing special characters, stop words and other irrelevant texts. So text data are pre-processed by the following steps:

i) First the texts are changed into lowercase so that there won't be any duplicates.
ii) Tokenization is done to remove the white spaces or to separate the words according to the whitespace.
iii) Stop Words, Punctuation and Special characters are removed as they increase the noise during the procedure.

Generation of a Word-Synset Document Matrix: A word-synset document matrix is generated after computing a count matrix for the number of times the words occurred in each synset. A synset S_i can be represented as $S_i = \{w_1, w_2, \ldots, w_n\}$. where wq (q = 1 to n)is a word belonging to the synsetS_i.

Synset words	bat mammal	bass guitar	…	bat cricket
nocturnal	1	0	…	0
Ball	0	0	…	1
Chord	0	1	…	0

Table 3: Shows a Fig. of word-synset document matrix.

In the above table 3, the rows represent all the words present in the synsets while columns represent the synsets. Each element of $W\ |nw * 8|$, i.e w_{ij} represents the count of j^{th}word ini^{th}synset. For an instance, the synset representing *bat mammal* has the words {bat, nocturnal, chiropteran etc.}. So occurrence of *nocturnal* word in *bat mammal* synset is counted to one, and since it is not present elsewhere the count is zero in other synsets. In a similar way, *ball* and *chord* also occurs in the synsets *bat cricket* and *bass guitar* respectively, so their counts in set to one in the corresponding columns.

Apply PCA to reduce dimension of the word-synset document matrix: Dimension reduction often prompts out some hidden or latent features. These features tend to be determining factor. As the word-synset document matrix is sparse because most of the cells are blanks due to number of words in the document. So, principal component analysis (PCA) is used to remove sparseness of matrix by reducing the dimension of the matrix. The number of components are decided by calculating the variance for each *x* values from 1 to 8 since the dimension of matrix is 8.We found out the variance for 6 components was 0.93 which was more than 0.86 from the 5 components. So the word – synset matrix W\nw * 8\was reduced to W''\nw * 6\The resultant matrix has 6 dimensions now. In the next table, an illustration of the reduced document matrix is shown below.

Reduced column words	col1	col2	…	col6
nocturnal	2.3	1.2	…	-4.5
ball	6	-3.6	…	0.3
chord	1	0.6	…	3.3

Table 4: Shows a *Fig.* of reduced word-synset document matrix.

In Table 4, the columns {col1, col 2,…,col y} represents the columns for the new reduced word-synset document matrix W''\nw * 6\ where y is the number of columns in the Resultant matrix and each element w''_{ij} represents the new values after the feature reduction.

Calculation of tf-idf of the reduced Word-Synset document matrix: The reduced word-synset document matrix, W″in table 4 is transformed into term frequency inverse document frequency (tf-idf) weighing scheme. Hence an inverse document frequency factor which diminishes the weight of the terms of matrix that occur very frequently and increases the weight of terms that occur rarely. So, tf-idf is used to enhance up the importance of the terms of document.

Reduced column words	col1	col2	…	col6
nocturnal	0.4	0.1	…	0.2
ball	-0.2	0.7	…	0.3
chord	0.3	-0.1	…	0.1

Table 5: Shows the tf-idf of reduced word-synset document matrix.

Here, in the above table each row represents the vector for each word of length |6|.From Table 5, the word vector *nocturnal* can be represented as<0.4×col1,0.1×col2,…..0.2×col6> of length |y |, where y is the number of columns.

Image Training Phase: This is the second phase of the proposed model. Here all the image related tasks like pre-processing the images and transforming images to vector during training phase are described in the following sub-sections.

Image Pre-processing: In the dataset, the images are organized as per their synsets. These images are then converted to gray scale. It is done to reduce the number of colour channels from 3(RGB value) to 1(grayscale). This reduction in channel does not hamper the uniqueness of the image features instead it increases the computation speed and decreases the resource utilization with respect to memory and computation. The images are then resized to reduce the computational complexity.

Extraction of image features: Harris Corner detection algorithm is used to detect corners or edges of the pre-processed images and then Scale-Invariant Feature Transform (SIFT) is used to obtain more dominant features from this corner points and regions under the edge boundaries. SIFT is invariant to rotation, scale and illumination. The final output is a list of feature key-points and their corresponding descriptors where from each image 'n' number of descriptor vectors are obtained with each of length 128.

K-means clustering of the image features: The extracted feature vectors are clustered into k-cluster centers using K-means clustering algorithm. Each 'k' clusters represents a quantized codeword and each image can be represented by a set of codewords.

Generation of codeword-synset document matrix: The codeword-synset document matrix is formed by computing the number of times the particular codeword has occurred in each synset.So the matrix $C|k * 8|$represents the codeword-synset document matrix of dimension $|k * 8|$ where $k$is the number of code words.

synsets **code words**	**synset 1**	**synset 2**	**…**	**synset 8**
cw_0	45	15	…	25
cw_1	12	25	…	78
cw_{k-1}	32	88	…	14

Table 6: Shows code word-synset document matrix

Each element of the matrix *C* in Table 6,c_{ij}, represents the occurrence of each j^{th}codewords in each i^{th}synsets.

Dimensionality reduction of codeword-synset document matrix: Dimensionality reduction algorithm is applied on codeword-synset document matrix as it has been also applied on word-synset document matrix. Here also the number of components remain same as word-(synset+gloss) document matrix that is resultant matrix has a dimension of 6 just like the same dimension of word-synset document matrix. Codeword-synset matrix is reduced from $\boldsymbol{C}^{|k * 8|}$ to $\boldsymbol{C}''^{|nw * 6}$ where the columns {col 1, col 2, …, col y} represents the columns for the new reduced matrix ***C*** and y = 6 is the number of dimensions of the resultant matrix.

reduced columns **codewords**	**col 1**	**col 2**	**…**	**col 6**
cw_0	33.5	5.6	…	-1.5
cw_1	12.3	-25.33	…	44.2
cw_{k-1}	2.112	0.2	…	2.6

Table 7: Shows the reduced codeword-synset document matrix.

Calculation of tf-idf of reduced codeword-synset document matrix: Tf-idf weighing scheme is applied on reduced codeword-synset document matrix where each row is a codeword of a vector of length |6|.Here, the codewordcw_0 is represented as

$cw_0 < 0.7 \times col1, 0.02 \times col2, \dots\dots\dots, -0.23 \times col6 >$is the columns of the matrix.

4.2. Testing Phase

In the testing phase, 8 testing images has been taken(which was described in section 4) and input words to find the semantic similarity between the word vectors and test image vectors by cosine similarity method. Now, testing phase can be divided into test image pre-processing, feature extraction from image, codeword labelling of test image features, vector sum of codeword of test image, extraction of word vectors and computing cosine similarly. Then, the ambiguity of polysemous noun can be removed with the help of visual context.

reduced columns **codewords**	**col 1**	**col 2**	**…**	**col 6**
cw_0	0.7	0.02	…	-0.23
cw_1	-0.4	0.32	…	0.4
cw_{k-1}	0.3	-0.06	…	0.55

Table 8: Shows the tf-idf of reduced codeword-synset document matrix.

Image Pre-processing and Feature Extraction from the Test Images: The same pre-processing and feature extraction method were used in testing that are applied during the training phase. The images were resized in the same dimension as it has been done for training images to maintain the consistency. The images are converted to gray scale and then Harris corner and SIFT method are used to extract key feature points from the resultant image. Therefore, m numbers of feature points are obtained from each test image.

Codeword label assigning for test image features: The *m* test image features are assigned to a codeword label which was nearest to it using Euclidean distance.Then, finally 'm' number of code words are obtained for a test image.

Vector sum of quantized codewords: Since a test image features contain m number of codewords, so a single codeword vector cannot be used to represent an image vector. So, we need to sum up all the m code words.

The combined image vector 'v_i' is defined by:

$$v_i = \sum_{j=1}^{m} cw_j \qquad (1)$$

Here cw_j= code-word vector 'j', where 'j' is 1 to m. The codeword vectors are extracted from the weighted vector (tf-idf) codeword – synset document matrix, which was computed during the training phase.

Extraction of word vectors corresponding to the input sentence: All the pre-processing task for testing are same as training phase. All the noun terms including ambiguous noun and surrounding words are mapped into weighted vector of word-synset matrix which is represented as 'v_w'.

Cosine Similarity of image vector and word vector: The vector representation may uncover hidden relationship between the word and the image. Then the cosine similarity is computed between input words and an image by using Eq.1.

$$if(\cos(v_i,v_{w1}) \geq Threshold \ \&\& \ \cos(v_i,v_{w2}) \geq Threshold .... \&\& \cos(v_i,v_{wl}) \geq Threshold \ \&\& \cos(v_w,v_{w1}) \geq Threshold \ \&\& \cos(v_w,v_{w2}) \geq Threshold ... \ \&\& \cos(v_i,v_{wl}) \geq Threshold$$

……………………………………Eq. 1

then (

the sense of the ambiguous noun term is similar to that of the test image

)

Wherev_i is the vector representing the test image, $v_w$is the ambiguous noun word vector, v_{wx} is the surrounding word vectors where x ranges from 1 to l and *Threshold* value is the lowest maximum similarity values between the correct sense word and image pair.

5. Results and Discussion

For both the training and testing of our algorithm, we have used a total of 8 different synsets. The total no. of 800 related images to these synsets are used for training (each synset contains a reduced set of images i.e. 100 images). However, the total no. of images used for testing is 8 ((each synset contains 1 image) only. The 8 synsets that we have considered are {(bat: word, mammal: sense), (bat: word, cricket: sense), (bass: word, guitar: sense), (bass: word, fish: sense), (mouse: word, animal: sense), (mouse: word, device: sense), (bank: word, piggy: sense), (bank: word, river: sense)}. These synsets are carefully crafted from voluminous ImageNet database so that each of the synset represents a polysemous noun. Due to the occurrence of both textual data and images in the training dataset, we have segregated the training phase into text training phase and an image training phase. The text training phase and image training phase is disjoint in its working. Prior to the training of textual data/image data, we have separated out the gloss and synsets from the images. We may visualise, a set of documents that contain gloss and synsets only that forms the textual data and a set of its associated images. Prior to the text training phase, the set of documents that contain textual data is cleaned and converted into a bag of words. From the bag of words, a word-synset matrix of size |n * 8| is constructed where n=no. of textual words. Further the size of the word-synset matrix is reduced to a size of |n * 6| by applying principal component analysis. The reduced word-synset matrix is then subjected to *tfidf* weighting scheme. Each word in the weighted word-synset matrix or weighted co-occurrence matrix (Table 5) is a vector of length |6| where each word is assigned with weights.

To summarise, the training phase generates two co-occurrence matrices namely (i) reduced weighted word-synset matrix of size |n * 6| where n=total of nouns and the dimensionality of the matrix is reduced to 6 from 8 after dimensionality reduction. Each noun that appears in the weighted word-synset matrix is a vector v_w. So, we can say the weighted word-synset matrix contains a set of vectors $\{v_{w1}, v_{w2},..v_{wn}\}$. A snapshot of the vectors in the reduced weighted word-synset matrix is shown as Table 9.

Table 9: Shows reduced weighted word-synset matrix.

v_{w1} = [-0.174715657, -0.313004547, 0.883177644, -0.302489406, -3.86E-16] //vector of snout
v_{w2} = [-0.180740, -0.482592, -0.411249, -0.077170, 7.479019e-01] //vector of fish
v_{w3} = [-0.174716, -0.313005, 0.883178, -0.302489, -3.857055e-16] // vector of mole
v_{w4} = [-0.180740, -0.482592, -0.411249, -0.077170, -7.479019e-01] //vector of chords
v_{w5} = [0.943868, 0.285049, -0.112293, -0.123490, 2.426497e-16] //vector of device
v_{w6} = [-0.067344, -0.081374, 0.067541, 0.992110, -2.652221e-15] //vector of cricket bat
v_{w7}= [-0.648107, 0.747697, -0.140517,-0.034066, -7.109811e-16] //vector of piggy bank
v_{w8} = [-0.648107, 0.747697, -0.140517,-0.034066, 4.317531e-16] //vector of river side
v_{w9} = [-0.215782379, -0.327652091, 0.698006055, 0.599057276, 3.28E-15] //vector of nocturnal

As per distributional semantics, neighbouring words that occur along with a polysemous word may help in disambiguation of the polysemous word. Keeping this mind, all neighbouring noun vectors that occur along with the polysemous noun vector in the weighted word-synset matrix are later used in the testing phase. (ii) reduced weighted codeword-synset matrix of size |k * 6| where k=total of visual codewords which describe their associated training images and the dimensionality is reduced to 6 from 8 after dimensionality reduction using principal component analysis. Each visual codeword may be interpreted as a cluster centre that is obtained after applying k-means algorithm prior to dimensionality reduction. It would be worth to mention that, the value of k is not fixed but varies where k = 50, 200 and 400. A snapshot of the visual codeword vectors in the reduced and weighted codeword-synset matrix for *k=400* is shown in Table 10.
Table 10.Shows reduced and weighted codeword-synset matrix for k=400.

cw_0 = -0.46264694022160435,-.8320569594197285,-0.2602497904335511,0.0380547524427767,-0.1563998317793062
cw_1 = 0.925293025699388,0.13830598000793193,-0.34761991854772073,0.014379322290528863,-0.060480573785790845
cw_2 =-0.709868963974717,-0.22846753940122164,-0.30196908514856463,-0.5781292528351364,-0.13590392215178734
cw_3=0.05868932199214941,0.39830017486372893,0.42567704818423807,0.25059029382359843,-0.7706595159200859

Although the test data contains 8 annotated images related to the above-mentioned synsets, the textual data is entirely omitted out during testing; only the images are considered. By applying image processing algorithms, *m* no. of features are extracted out from each test image. The *m* no. of image features are assigned with a codeword by measuring the Euclidean distance (nearest to the cluster centre) between the *m* image features and codewords (cluster centres) that was obtained after applying k-means algorithm prior to dimensionality reduction in the image training phase. Since, test image features contain *m* number of codewords, so a single codeword vector cannot be used to represent an image vector. So, all the m no. of codewords are summed up to obtain a single vector v_i.
To measure the semantic relatedness between the test image *I* and a word (noun) *w*, e.g. *chords*, we simply compute the cosine similarity (using equation 1 Section 4.2) between v_i and v_w.

Images Words	**Mouse mammal image**	**Mouse device image**	**Bat mammal image**	**Bat cricket image**	**Bass fish image**	**Bass guitar image**	**Bank piggy image**	**Bank river image**
Chords	0.319255	0.324022	0.324199	0.328511	0.325303	0.329986	0.331097	0.324554
Cricket	0.39774	0.420871	0.403356	0.411727	0.411222	0.40981	0.415049	0.409276
Device	0.235278	0.240303	0.236368	0.243755	0.235849	0.246346	0.2451	0.235786
Fish	0.246239	0.360058	0.054145	0.446306	0.357017	0.156072	0.253959	0.353734
Snout	0.23813	0.391797	0.208563	0.015626	0.300419	0.200911	0.196992	0.409678
nocturnal	0.177789	0.179163	0.182210	0.180087	0.181285	0.184454	0.180392	0.181396
Piggy	0.288497	0.284783	0.29434	0.28641	0.290259	0.291567	0.289549	0.288899
riverside	0.422608	0.426347	0.428164	0.420415	0.427943	0.417986	0.42187	0.42601

Table 11: Cosine similarities between words and the test images taking 50 code words

Images Words	**Mouse mammal image**	**Mouse device image**	**Bat mammal image**	**Bat cricket image**	**Bass fish image**	**Bass guitar image**	**Bank piggy image**	**Bank river image**
Chords	-0.20114	0.139074	0.461989	-0.21429	0.455604	0.193188	0.311374	0.284403
Cricket	0.061663	0.041063	-0.17292	0.323852	-0.11745	-0.08789	0.00551	-0.26956
Device	-0.53154	0.453921	-0.2001	0.487456	0.197923	-0.02003	0.204325	-0.03291
Fish	-0.26469	0.097479	0.176798	-0.25621	0.2503	0.003112	0.201749	-0.1012
Snout	0.001866	-0.06077	0.41264	0.351674	0.234065	0.345618	0.343713	0.278421
nocturnal	0.055198	-0.02141	0.330215	0.194022	0.086854	0.286099	0.266309	0.359557
Piggy	0.955405	-0.93185	-0.74081	-0.66902	-0.93422	-0.21547	0.321457	-0.69823
Riverside	0.100201	-0.24525	-0.32146	-0.45148	0.124154	-0.80073	-0.97576	0.324148

Table 12: Cosine similarities between words and the test images taking 200 codewords.

Table 11 illustrates the cosine similarity between a word and an image. Each row represents the cosine similarity scores that are obtained against a word vs image features. For example, corresponding to the word "*chords*" the cosine similarity value obtained against "*bass guitar*" image feature is 0.329986, which is highlighted in bold, and the cosine similarity value obtained between "*chords*" and image feature of "*bank piggy*" is 0.331097. Similarly, the other cosine similarity values obtained between the word "*chords*" against the remaining image features are within the range of 0.3 to 0.35. From these cosine similarity scores we cannot easily equate out which image features are more relevant for "*chords*" as because 50 codewords have failed to capture the image properties. So we have increased the number of codewords to k value=200.

In Table 12 we can see that the similarity between the term *chord* and the image *bass guitar* has reduced from 0.329986 to 0.193188. Yet, even though *chord* and *bass guitar* is related, but the obtained cosine similarity value is not promising and thus we cannot conclude that chord and bass guitar are related. Not just this pair but almost all the correct pairs have similarities less than 0.5.So, the codeword is creased with k value=400.

Images Words	Mouse mammal image	Mouse device image	Bat mammal image	Bat cricket image	Bass fish image	Bass guitar image	Bank piggy image	Bank river image
Chords	0.017005	-0.47578	-0.36342	-0.13466	0.057731	0.707457	-0.44306	0.242109
Cricket	0.305485	-0.4986	0.246444	0.501409	0.295527	0.089551	-0.61229	0.027714
Device	0.06951	0.432569	0.228928	-0.51439	-0.00256	-0.79271	0.106483	0.149259
Fish	0.135917	0.282479	-0.57268	0.018587	0.679153	0.358496	0.20377	-0.91931
Snout	0.513997	-0.48533	0.648718	-0.49758	0.052146	-0.08366	-0.22865	-0.13797
nocturnal	0.637697	-0.78952	0.661866	0.066893	0.300108	0.053552	-0.68642	-0.09751
Piggy	-0.8853	0.400428	-0.34262	0.500317	-0.78712	0.1681	0.645183	-0.23451
riverside	-0.56552	0.258548	-0.12458	0.142176	-0.64513	0.241578	0.142154	0.455003

Table 13: Cosine Similarity between the words and the test images taking 400 codewords.

As we can see in Table 13, the cosine similarity pairs are more meaningful. The term *chord,* for example is now highly related with the image *bass guitar* as the cosine value is 0.707457. An interesting result is obtained for the terms {*snout* and *nocturnal*} with the images {*mouse mammal* and *bat mammal*}. These pairs are highly correlated because they share similar visual features. So our model is not only able to capture the relationship between the exact pairs, but also between the pairs which share some common features. Likewise all the correct word and image pairs are shown in bold. Through cosine similarity values we are able to reduce the semantic gap between a word and image. The higher the cosine similarity value, the more related is the word with the image. It was the first objective of our work. Now, this semantic relationships have been used to remove ambiguity of polysemous noun which is the second objective of our work.

In order to disambiguate a polysemous noun, we need to relate to its appropriate sense. Therefore, for testing purpose, a sentence is taken such as "We play cricket with *bat* and ball". Here *bat* is a polysemous noun, as *bat* may either mean a mammal or a cricket bat. We should be able to assign the correct sense to the polysemous term. So, the surrounding terms are taken from the given test sentence which are relevant such as {*cricket* and *ball*}. Then we have calculated the cosine similarities between these surrounding terms (*ball* and *cricket*) of the polysemy noun *bat* and the test image, and also with the *bat* word itself.

Input sentence: "We play cricket with bat and ball."
Ambiguous noun term: *bat*
Surrounding terms: {cricket, ball}, relevant terms from the input sentence

For validation, the term *nocturnal* is used which belongs to the other sense of the *bat* term (bat mammal). To check that {cricket, ball} are more related with the *bat* word than with *nocturnal* and *bat* for this given sentence.

Words	An image of cricket bat
ball	0.501409
cricket	0.501409
nocturnal	0.066893

Table 14: Cosine similarity between the surrounding terms and the test image.

From the Table 14 we have inferred that the test image which is of the synset *bat cricket* is more similar to the surrounding words (*ball and cricket*) of the polysemy noun bat as compared with the term *nocturnal.* Also we check if these similarity values are more than or equal to the threshold value. We have used the threshold value as 0.432569, since this value was the minimum among the similarity (defined in Table 4.3) values which was able to distinguish between a word and its actual sense image (*device* term and the *mouse device image*). And as we can see in the Table 4.4, the similarities between the correct pairs is more than the threshold value. Now we will find the similarity between the *bat mammal* image and the terms {*ball, cricket, and nocturnal*}.
Input sentence: "We play cricket with bat and ball."
Target/ambiguous noun term: *bat*
Surrounding terms: {cricket, ball}, relevant terms from the input sentence

Words	An image of mammal bat
ball	0.246444
cricket	0.246444
nocturnal	0.661866

Table 13: Cosine similarity between the surrounding terms and the mammal bat image.

In Table 13, when we have introduced the mammal bat image, its similarity with the surrounding terms {*ball, cricket*} is lesser than the threshold value. So this *bat* term do not have the sense of bat mammal.

6. Conclusion and Future Work

Removal of ambiguity of polysemous word is very hard as it depends on the context. We have used a supervised machine learning approach for our work. A semantic bridge is established in between images and texts. Further, it is used for removing ambiguity of polysemous noun in terms of machine translation system that can translate from a source language to its equivalent target language. Human being can easily find out the sense of any ambiguous term depending on the context but machine cannot. The term 'bat' generates ambiguity if the source text of machine translation is "We play cricket with bat and ball". Then the problem can be solved only by finding surrounding terms that are present in the context. The terms that have been obtained after training are considered most probable surrounding terms of a polysemous noun. The surrounding terms are validated with the image of both mammal bat and wooden stick to check the similarity. Our proposed work can increase the reliability and quality of machine translation by solving ambiguity of noun in context with multimodal approach. Unimodal approach only uses text. The uniqueness of our proposed approach is that we are using both image and text from same database to find the sense of ambiguous term. It is a well-known fact that picture always gives quick and effective idea than any written document. The advantage of our proposed model is that it does not require any sense tagged corpus. Finally, we have concluded that a multimodal approach which uses visual information as contexts to an ambiguous sentence is promising in Word Sense Disambiguation. . The experiment was done on different *k* number of code-words (i.e.) {50,200,400}. From the experimental results, it is observed that the proposed multimodal distributional semantics model (MDSM) accurately calculates both semantic similarity score and cosine similarity scores than the traditional ambiguous removal techniques such as unimodal and multimodal based approaches. In future, we can increase the number of synsets so that more polysemous nouns can be disambiguated as well as more surrounding terms can be obtained. Also, we shall introduce a novel machine learning based approach to improve the performance.

Acknowledgements

The corresponding author is highly thankful to the Ministry of Electronics and Information Technology; Government of India initiated "Visvesvaraya PhD Scheme for Electronics and IT" for the support to carry out the research work.

References

[1] Bar-Hillel, Y. (1960). The present status of automatic translation of languages. *Advances in computers*, *1*, 91-163.
[2] Barnard, K., & Johnson, M. (2005). Word sense disambiguation with pictures. *Artificial Intelligence*, *167*(1-2), 13-30.
[3] Barnard, K., & Forsyth, D. (2001, July). Learning the semantics of words and pictures. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (Vol. 2, pp. 408-415). IEEE.
[4] Barnard, K., & Forsyth, D. (2001, July). Learning the semantics of words and pictures. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (Vol. 2, pp. 408-415). IEEE.
[5] Barnard, K., Duygulu, P., & Forsyth, D. (2001, December). Clustering art. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (Vol. 2, pp. II-II). IEEE.

[6] Barnard, K., & Johnson, M. (2005). Word sense disambiguation with pictures. *Artificial Intelligence*, *167*(1-2), 13-30.

[7] Barnard, K., Yanai, K., Johnson, M., & Gabbur, P. (2006). Cross modal disambiguation. In *Toward Category-Level Object Recognition* (pp. 238-257). Springer, Berlin, Heidelberg.

[8] Banerjee, S., & Pedersen, T. (2002, February). An adapted Lesk algorithm for word sense disambiguation using WordNet. In *International conference on intelligent text processing and computational linguistics* (pp. 136-145). Springer, Berlin, Heidelberg.

[9] Borgohain, S., & Nair, S. B. (2010). Towards a Pictorially Grounded Language for Machine-Aided Translation. *Int. J. Asian Lang. Process.*, *20*(3), 87-110.

[10] Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D. (2010, September). Every picture tells a story: Generating sentences from images. In *European conference on computer vision* (pp. 15-29). Springer, Berlin, Heidelberg.

[11] Feng, Y., & Lapata, M. (2010, July). How many words is a picture worth? automatic caption generation for news images. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics* (pp. 1239-1249).

[12] Harris, Z. S. (1954). Distributional Structure, WORD, vol. 10, no. 2–3.

[13] James, N., & Hudelot, C. (2009). Towards semantic image annotation with keyword disambiguation using semantic and visual knowledge. In *Proceedings of IJCAI* (Vol. 9).

[14] Kher, H. R., & Thakar, V. K. (2014, January). Scale invariant feature transform based image matching and registration. In *2014 Fifth International Conference on Signal and Image Processing* (pp. 50-55). IEEE.

[15] Leong, C. W., & Mihalcea, R. (2011). Measuring the semantic relatedness between words and images. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011).*

[16] Leong, C. W., & Mihalcea, R. (2011, November). Going beyond text: A hybrid image-text approach for measuring word relatedness. In *Proceedings of 5th International Joint Conference on Natural Language Processing* (pp. 1403-1407).

[17] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91-110.

[18] Mihalcea, R. (1998). Semcor semantically tagged corpus. *Unpublished manuscript*.

[19] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

[20] Navigli, R., & Velardi, P. (2005). Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, *27*(7), 1075-1086.

[21] Orkphol, K., & Yang, W. (2019). Word sense disambiguation using cosine similarity collaborates with Word2vec and WordNet. *Future Internet*, *11*(5), 114.

[22] Rieger, B. B. (1991). *On distributed representation in word semantics*. Berkeley, CA: International Computer Science Institute.

[23] Searle, J. R. (1980). Minds, brains, and programs. Behavioral and Brain Sciences3: 41724.[aSL](1992) The rediscovery of mind.

[24] Su, Y., & Jurie, F. (2011, November). Visual word disambiguation by semantic contexts. In *2011 International Conference on Computer Vision* (pp. 311-318). IEEE.

[25] Turdakov, D. Y. (2010). Word sense disambiguation methods. *Programming and Computer Software*, *36*(6), 309-326.

[26] Wang, Y., Wang, M., & Fujita, H. (2020). Word sense disambiguation: A comprehensive knowledge exploitation framework. *Knowledge-Based Systems*, *190*, 105030.

[27] Westerveld, T. (2000, April). Image Retrieval: Content versus Context. In *RIAO* (pp. 276-284).