A NOVEL REFINEMENT METHOD FOR AUTOMATIC IMAGE ANNOTATION SYSTEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ERŞAN DEMİRCİOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JUNE 2011

Approval of the thesis:

A NOVEL REFINEMENT METHOD FOR AUTOMATIC IMAGE ANNOTATION SYSTEMS

submitted by **ERŞAN DEMİRCİOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _____

Prof. Dr. Fatoş Tünay Yarman Vural
Supervisor, **Computer Engineering Department, METU** _____

Examining Committee Members:

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Prof. Dr. Fatoş Tünay Yarman Vural
Computer Engineering, METU _____

Assist. Prof. Dr. Ahmet Oğuz Akyüz
Computer Engineering, METU _____

Onur Pekcan, M.Sc.
Civil Engineering, METU _____

Dr. Ahmet Sayar
Computer Engineer, TÜBİTAK UZAY _____

Date: **27.06.2011**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ERŞAN DEMİRCİOĞLU

Signature :

ABSTRACT

A NOVEL REFINEMENT METHOD FOR AUTOMATIC IMAGE ANNOTATION SYSTEMS

Demircioğlu, Erşan

M.Sc., Department of Computer Engineering

Supervisor : Prof. Dr. Fatoş Tünay Yarman Vural

June 2011, 79 pages

Image annotation could be defined as the process of assigning a set of content related words to the image. An automatic image annotation system constructs the relationship between words and low level visual descriptors, which are extracted from images and by using these relationships annotates a newly seen image. The high demand on image annotation requirement increases the need to automatic image annotation systems. However, performances of current annotation methods are far from practical usage. The most common problem of current methods is the gap between semantic words and low level visual descriptors. Because of the semantic gap, annotation results of these methods contain irrelevant noisy words. To give more relevant results, refinement methods should be applied to classical image annotation outputs.

In this work, we represent a novel refinement approach for image annotation problem. The proposed system attacks the semantic gap problem by using the relationship between the words which are obtained from the dataset. Establishment of this relationship is the most crucial problem of the refinement process. In this study, we suggest a probabilistic and fuzzy approach for modeling the relationship among the words in the vocabulary, which is then em-

ployed to generate candidate annotations, based on the output of the image annotator. Candidate annotations are represented by a set of relational graphs. Finally, one of the generated candidate annotations is selected as a refined annotation result by using a clique optimization technique applied to the candidate annotation graph.

Keywords: automatic image annotation, image annotation refinement, fuzzy sets, relational graphs, maximum weighted clique

ÖZ

OTOMATİK GÖRÜNTÜ ETİKETLEME SİSTEMLERİ İÇİN YENİ BİR İYİLEŞTİRME YÖNTEMİ

Demircioğlu, Erşan
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş Tünay Yarman Vural

Haziran 2011, 79 sayfa

Görüntü etiketleme, bir görüntüye görüntünün içeri ile ilgili kelimler atama işlemi olarak tanımlanabilir. Bir etiketleme sistemi kelimelerle, görüntülerden elde edilen düşük seviye görsel özniteliklerle arası ilişki kurar ve bu ilişkiyi kullanarak yeni gelen bir görüntüyü etiketler. Görüntü etiketleme ihtiyacı arttıkça, görüntü etiketleme sistemlerine olan ihtiyaçta hızla artmaktadır. Literatürde yoğun olarak çalışılmakta olan bu problem için hali hazırda önerilen yöntemlerin performansları pratik kullanım için yeterli değildir. Bu sistemlerin ortak problemi yüksek seviye anlamsal kelimelerle düşük seviye görsel öznitelikler arasındaki anlamsal boşluktur. Bu anlamasal boşluk nedeniyle, sonuçlar ilgisiz kelimelerde içerebilmektedir. Daha iyi sonuçların verilebilmesi için iyileştirme yöntemlerinin kullanılması gerekmektedir.

Bu çalışmada, otomatik görüntü etiketleme sistemleri için yeni bir iyileştirme yöntemini sunulmaktadır. Önerilen yöntem anlamsal boşluk problemine veri kümesinden elde ettiği kelimeler arası ilişkiyi kullanarak çözüm üretir. Kelimeler arası ilişkinin kurulması iyileştirme işleminin en önemli problemdir. Bu çalışmada, kelimeler arası ilişkiyi modellemek için olasılıksal ve bulanık yaklaşımlar sunduk. Kelimler arası ilişkiler, görüntü etiketleyici tarafından sağlanan etiketler üzerine uygulanarak aday etiketlemeler üretir. Aday etiketlemeler bir ilişkilse

çizge üzerinde gösterilebilmektedir. Son olarak üretilen aday etiketlemelerden bir tanesi klik optimizasyon teknikleri kullanılarak seçilir ve iyileştirilmiş etiketlene olarak sunulur.

Anahtar Kelimeler: otomatik görüntü etiketleme, görüntü etiketinin iyileştirilmesi, bulanık kümeler, ilişkisel çizgeler, maksimum ağırlıklı klik

To my family

ACKNOWLEDGMENTS

First of all, I would like to thank Prof. Dr. Fatoş Tünay Yarman Vural for her guidance and support throughout this study. Without her, this study would not be accomplished.

I also would like to thank members of Image Processing and Pattern Recognition Laboratory for their helpful advices.

Finally, I would like to thank my lovely wife for her support and patience.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Digital imaging instruments find place for themselves not only on digital cameras but also almost every commercial electronic devices like cellular phones and mobile PCs. Only 10 year ago people could take rare photos of important events. Indexing and retrieving these images were relatively easy task. After widespread usage of digital cameras, the cost of taking one photo is dramatically decreased and this leads to producing large amount of images in short a time. Manually indexing and retrieving these images need intensive workforce and exposes the need of automatic indexing and retrieving systems.

Early image retrieval systems are focused on retrieving images according to their visual similarities. User selects an image which is similar to the requested image and the system matches visually similar images. Unfortunately, finding similar image is not always an easy task for the user. Later, the systems are focused on retrieving images according to their contents. Users describe requested image by words and the system finds images according to input words. These kind of systems are relatively easier than the classical image retrieval systems. However, all images in the database should be annotated with content related words to prepare for the retrieving process.

Automatic image annotation could be defined as the process of assigning content related words to the image. Most of the current automatic image annotation systems are based on machine learning techniques to design an automatic image annotation task. These annotation systems are trained by using a training dataset, which are manually annotated by human annotators. These systems extract a set of visual descriptors from the images in a dataset and construct relationships between low level visual descriptors and words. According to these relationships, a newly seen image is annotated and a list of word is offered as annotation result

of the underlying image.

Because of wide area of applications, image annotation problem is an active research area in computer vision. However, performances of current automatic image annotation systems are still far from practical usage. One of the major problems of designing an automatic image annotation system is, the difficulty of assigning high level semantic words into images by using low level visual descriptors. In fact low level visual descriptors contain only information like shape, color and texture, but not semantic information. For example, a visual descriptor describes an image as blue and homogeneous, but is short to distinguish image of *sky* or *sea*. These kinds of confusions lower the annotation quality drastically and lead to unrelated words in annotation results.

In order to attack the above mentioned problem, refinement systems are proposed to remove unrelated words from annotation results. Most of the current refinement systems update an annotation result by using semantic information. Semantic information defines relationship between words. A refinement system receives annotation result from an automatic image annotator and identifies the unrelated words by using relationship between words. For example, let us assume that an annotator could not clearly decide if an image contains a *sky* or *sea* and assigns both of these two words to the image at the initial step. If the annotator finds an additional object like *plane*, by using relationship between *plane* and *sky*, the refinement system decides that this is an image of *sky* and removes the word *sea* the from annotation result.

Current refinement systems could be categorized under two groups according to their source of semantic information. Systems under the first group establish the relationship between words from external sources such as a thesaurus or a dictionary. External sources provide generic relationship between words which is independent from the dataset. WordNet [1] is one of the most popular word database which constructs relationship between words according to their semantic and lexical relations. Because of large and generic coverage, they are far from reflecting the specific properties of a small dataset. Systems under the second groups extract relationship between words from dataset itself. These systems align the relationship of words by modeling a dataset.

In the scope of this thesis, we propose a novel refinement method for automatic image annotation system, based on the second group of approach. Proposed refinement method extracts relationship between words from dataset and combines with annotation results which are pro-

duced by an automatic image annotator.

We introduce two new approaches in this thesis. The first one is about extracting relationship between words from the image dataset itself. We employ fuzzy framework for extracting the relationship between words. Distribution of words in many image annotation dataset is not stable. Some words occur more frequently than others. The proposed fuzzy framework provides a flexible workbench and represents the relationship between words without effecting occurrence frequency.

The second approach is about refining a raw annotation result of an automatic image annotator. Proposed refinement method generates a candidate annotation result for a given word by using raw annotation result and relationship between words. The candidate annotation generation process generates local optimal solution for a given word. By using this process, the proposed refinement method generates a candidate annotation for each word in raw annotation result and generates a solution space for raw annotation result. One of the candidate solution is selected as the refined annotation result.

1.1 Thesis Outline

The thesis is organized as follows;

Chapter 2 contains state of the art techniques for automatic image annotation and various refinement techniques. First, we make a mathematical definition of automatic image annotation and image annotation refinement problem. Then, we describe commonly used metrics to measure the performance of annotation and refinement systems. Finally we list the state of the art techniques for automatic image annotation and image annotation refinement.

Chapter 3 provides the detailed description of the proposed refinement system. First of all we draw outline of proposed method and introduce the major components of systems. Then, the algorithm for each component is given. Sample run is used to explain the details of the suggested method.

Chapter 4 is dedicated to experiments, where the major strength and weakness of the proposed method is shown.

Chapter 5 concludes the thesis by giving a brief summary of the suggested method and discussion on future works.

CHAPTER 2

STATE OF THE ART TECHNIQUES FOR AUTOMATIC IMAGE ANNOTATION AND ITS REFINEMENT TECHNIQUES

This chapter aims to explain the state of art techniques for image annotation and image annotation refinement problem. First, we give a mathematical definition of automatic image annotation problem. Then, we explain why we need a refinement system and make a mathematical definition of automatic image annotation refinement problem.

To explain and compare state of art techniques, we introduce the commonly used performance measures in image annotation problem. Throughout this thesis, we also use and refer to these performance measures. In the following sections, we investigate the state of art image annotation and image annotation refinement techniques.

2.1 Definition of Automatic Image Annotation Problem

The major goal of the image annotation problem is to assign high level semantic words to an image depending on the content. This task can be automatically achieved by machine learning techniques, provided that there is already an annotated set of images which can be employed for training of an automatic image annotation system.

Therefore, an automatic image annotation system receives a set of annotated images as input and automatically annotates an image by the words in its vocabulary. This approach enables us to annotate huge amount of images, using a small training data set which is annotated manually.

Training of an automatic image annotation system basically involves associating the high level annotation words to the low level visual descriptors extracted from the images. These low level descriptors may be extracted either from the whole image or from the segments of an image. Selection of low level descriptors depends on the application domain and mostly involves color, texture or shape features. However, we expect that selected low level descriptors should somehow represent the high level semantic words in the vocabulary.

Mathematically speaking, suppose that a training image database $S = \{I_1, I_2, \ldots, I_{SN}\}$ contains SN manually annotated images. An image $I_i = \{D_i, W_i\}$ is represented as a composition of set of visual descriptors and a set of content related semantic words. $D_i = \{\delta_{i_1}, \delta_{i_2}, \ldots, \delta_{i_{DN}}\}$ is a set of visual descriptors that contains DN visual descriptors. A visual descriptor, δ_{i_j} describes a low level visual characteristic of the image I_i. According to domain, it could describe low level features like edge, color and texture or high level features like objects and regions. $W_i = \{w_{i_1}, w_{i_2}, \ldots, w_{i_{WN}}\}$ is a set of content related semantic words, which are assigned to the image I_i. An image could be annotated by at most WN words. All possible words are defined in a vocabulary $V = \{w_1, w_2, \ldots, w_{VN}\}$ and W_i is a subset of vocabulary V.

Image annotation problem is to assign a set of high level semantic words W_x to a newly seen image $I_x = \{D_x, \emptyset\}$ according to its set of visual descriptor D_x by using training image database S.

An automatic image annotation system provides a function $Annotator(I_x)$ which maps visual descriptors to words and produces vector A_x as an intermediate output. Vector A_x contains possibility value of existence $p_{i,x}$ ($0.0 \leq p_{i,x} \leq 1.0$) for each word w_i in vocabulary V for image I_x. Values in A_x are sorted in descending order and top WN words with highest possibility values are assigned to W_x for image I_x.

$$Annotator(I_x) : D_x \rightarrow A_x \rightarrow W_x \,, \tag{2.1}$$

where

$$A_x = \{p_{1,x}, p_{2,x} \ldots, p_{VN,x}\} \,. \tag{2.2}$$

Therefore, the crucial problem of automatic image annotation problem is to find $Annotator(I_x)$ which represents the relationship between D_x and W_x.

2.2 Definition of Automatic Image Annotation Refinement Problem

Unfortunately, the above definition of automatic image annotation problem is ill-conditioned due to the vague association between the high level words and low level visual descriptors. This fact is known as the semantic gap problem and should be handled with some heuristics. Because of the semantic gap between the visual descriptors and semantic words, annotation results usually contain unrelated and/or noisy words. One way to attack the semantic gap problem is to develop a refinement system at the output of the image annotator. The aim of an image annotation refinement system is to remove these noisy words by employing semantic similarities between words in the annotation system. Semantic similarity is a metric based on the relatedness or likeness of terms in semantic content. Unfortunately, there is no well defined methodology to find the semantic similarities between the words. In this study, we define similarity matrix and explore on the entries of this matrix.

Mathematically speaking, semantic similarity matrix R is a symmetric $VN \times VN$ matrix which contains semantic similarity values for words in vocabulary V.

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,VN} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,VN} \\ \vdots & \vdots & \ddots & \vdots \\ r_{VN,1} & r_{VN,2} & \cdots & r_{VN,VN} \end{bmatrix}, \tag{2.3}$$

where $r_{i,j}$ represents semantic similarity between words w_i and w_j. Semantic similarity between two words should satisfy the following properties;

Semantic similarly is represented by a real number values between 0.0 and 1.0. 0.0 indicates almost no similarity and 1.0 indicates high similarity.

$$0.0 \leq r_{i,j} \leq 1.0 \tag{2.4}$$

A word has a maximum similarity with itself.

$$r_{i,i} = 1.0 \tag{2.5}$$

Semantic similarity value of w_i and w_j has to be equal to semantic similarity value of w_j and w_i.

$$r_{i,j} = r_{j,i} \tag{2.6}$$

A refinement system is developed for a specific annotator system. This annotator system is denoted by $Annotator^{raw}$. As described in the previous section, the intermediate output of $Annotator^{raw}$ for newly seen image I_x contains possibility value of existence for each word in vocabulary. This intermediate output is entitled as A_x^{raw} and $Annotator^{raw}$ for an image I_x is defined as a mapping;

$$Annotator^{raw}(I_x) : D_x \rightarrow A_x^{raw} \rightarrow W_x^{raw} \, , \tag{2.7}$$

where,

$$A_x^{raw} = \{p_{1,x}^{raw}, p_{2,x}^{raw} \ldots, p_{VN,x}^{raw}\} \, . \tag{2.8}$$

A refinement system defines a function $Refiner(A_x^{raw}, R)$ and receives raw annotation vector A_x^{raw} from raw annotator $Annotator^{raw}$ for image I_x and refines raw annotation vector A_x^{raw} by using semantic similarities matrix R to output $A_x^{refined}$. Mathematically speaking,

$$Refiner(A_x^{raw}, R) = A_x^{refined} = \{p_{1,x}^{refined}, p_{2,x}^{refined} \ldots, p_{VN,x}^{refined}\} \tag{2.9}$$

Values in $A_x^{refined}$ are sorted in descending order and top WN words with highest possibility value are assigned to $W_x^{refined}$ for image I_x.

2.3 Performance Measures for Automatic Image Annotation Systems

To be able to compare methods, there are four popular performance measures, namely *precision, recall, f-score* and *number of words with non-zero recall (NZC)*, which are commonly used to measure the performance of image annotation systems. *Precision* and *recall* are calculated per word defined in vocabulary V. *Precision* of word w ($precision_w$) is the ratio of the number of images correctly annotated with word w by automatic image annotator (denoted by hit_w) to the number of images annotated with word w by automatic image annotator (denoted by $auto_w$), given as follows;

$$precision_w = \frac{hit_w}{auto_w} \, . \tag{2.10}$$

Recall of word w ($recall_w$) is the ratio of the number of images correctly annotated with word w by automatic image annotator (denoted by hit_w) to the number of images annotated with word w in test set (denoted by $ground_w$), given as follows;

$$recall_w = \frac{hit_w}{ground_w} \, . \tag{2.11}$$

Table 2.1: Nomenclature

S	:	Training image database which contains set of manually annotated images ($\{I_1, I_2, \ldots, I_{SN}\}$)
SN	:	Size of training set
I_i	:	i^{th} image in dataset S, represented as a union of D_i and W_i
D_i	:	Set of visual descriptors $\{\delta_{i_1}, \delta_{i_2}, \ldots, \delta_{i_{DN}}\}$ of image I_i
δ_{i_j}	:	j^{th} visual descriptor of i^{th} image
DN	:	Number of visual descriptors which define an image
W_i	:	Set of content related words $\{w_{i_1}, w_{i_2}, \ldots, w_{i_{WN}}\}$ of image I_i
w_{i_j}	:	j^{th} word assigned to i^{th} image according to its content and $w_{i_j} \in V$
WN	:	Maximum number of words could be assigned to an image
V	:	Vocabulary which contains all possible words ($\{w_1, w_2, \ldots, w_{VN}\}$).
VN	:	Size of vocabulary
I_x	:	Newly seen image, it only contains visual descriptor, set of content related words are empty set ($I_x = \{D_x, \emptyset\}$).
$Annotator(I_x)$	:	Function which annotates input image and produce intermediate output A_x and final output W_x.
A_x	:	Set of possibility value of existence of words in vocabulary V ($\{p_{1,x}, p_{2,x} \ldots, p_{VN,x}\}$).
$p_{i,x}$	:	Possibility value of existence of word w_i in image I_x, ($0.0 \leq p_{i,x} \leq 1.0$)
W_x	:	Set of contend related words assigned to image I_x ($\{w_{x_1}, w_{x_2}, \ldots, w_{x_{WN}}\}$)
A_x^{raw}	:	Intermediate output of annotator which is subject of refinement process.
$Refiner(A_x^{raw}, R)$	:	Function which refines A_x^{raw} by using relation between words R and produces $A_x^{refined}$.
R	:	$VN \times VN$ semantic similarities matrix which contains semantic similarity values of words defined in vocabulary V
$r_{i,j}$	:	(i, j) entry of R and represents semantic similarity of words w_i and w_j

F-Score is the harmonic mean of precision and recall. It is used for combining precision and recall values and obtaining one comparable measurement for the method:

$$F\text{-}Score = \frac{2 * precision * recall}{precision + recall} \tag{2.12}$$

Number of words with non-zero recall (NZC) measures the vocabulary coverage of the method. In some datasets, some words are more frequently occurred than others. Only by annotating these words, the method would give high precision and recall values, but coverage of vocabulary is limited.

2.4 Automatic Image Annotation Techniques

Although there are dozens of different techniques for automatic image annotation problem, most of them share the same workflow. First of all, the automatic image annotator establishes relationship between low level visual descriptor and high level words by using training images. Sometimes, this relationship is represented by a mapping which is trained by using a manually annotated image dataset. Sometimes, this relation established by joint or conditional probability function between visual descriptors and words. Then, the automatic image annotator extracts low level features from the input image. By using relationship between low level visual descriptor and high level words, the automatic image annotator calculates possibility values for each word in the vocabulary. Finally, the automatic image annotation system assigns top n words with highest possibility value to the input image.

As it can be seen, the main difference between various automatic image annotation techniques is the definition and learning methodologies in the mapping function.

Most of the time, the definition of the mapping function depends on the application domain and the properties of the available learning data set. If there is statistically sufficient amount of manually annotated data, the probabilistic approaches can be employed [2, 3]. Then, the mapping function is defined by joint and/or conditional probability density functions among the images and the words. Otherwise, deterministic or fuzzy machine learning techniques, such as support vector machines, fuzzy k-nearest neighbor methods, linear discriminant analysis techniques can be employed [4, 5, 6].

Automatic image annotation techniques can be grouped depending on the representation of

the low level visual features. Sometimes a set of features are extracted from the whole image and these features are then associated with the words in the vocabulary [7]. This approach avoids the expensive and erroneous process of segmentation. However, associating the whole image into set a set of words has its own problems. It is quite obvious that one cannot match a certain object in a image using a feature extracted from the whole image. In this study, we do not focus on a specific image annotation technique, nor we try to enhance the available system. Instead, we employ the output of an image annotation system in a refinement process to improve the performance of an annotation system. Therefore, in the following sections rather than giving a thorough survey of automatic image annotation systems, we suffice to summarize some state of the art techniques. Any of the available techniques can be employed in the refinement technique, suggested in Chapter 3.

2.4.1 Relevance Models

Relevance Models assume that, there exists a probability distribution $P(.|I)$ for each image I [8]. To annotate image I_x with appropriate words, the model estimates $P(w_i|I_x)$ for each word w_i in vocabulary V. Since the new image I_x does not contain any information about words, the joint distribution $P(w_i, I_x)$ over training set could be computed over previously annotated images S instead of maximum-likelihood estimation.

$$P(w_i, I_x) \approx P(w_i, \delta_{x_1}, \ldots, \delta_{x_{Dn}}) = \sum_{I_j \in S} P(I_j) P(w_i, \delta_{x_1}, \ldots, \delta_{x_{Dn}} | I_j) \,, \tag{2.13}$$

where, $P(I_j)$ is the probability of picking image I_j in training set S. If there is no prior information about $P(I_j)$, it could be assumed uniform and taken $\frac{1}{SN}$ [9].

Relevance Model is used by three well know annotation systems, namely Cross-Media Relevance Model (CMRM) [8], Continuous Relevance Model (CRM) [9] and Multiple Bernoulli Relevance Model (MBRM) [10].

CMRM uses *blobs* to identify an image ($I = \{b_1, \ldots b_n\}$). *Blobs* are segments which are generated by clustering the image features, like color, texture and edge. Each *blob* could be identified by more than one *word*. By clustering feature vectors, CMRM quantizes continuous features to discrete blobs [9]. According to CMRM, observing w and *blobs* are mutually

independent events and according to this assumption, $P(w_i, I_x)$ could be computed as follows:

$$P(w_i, I_x) \approx P(w_i, b_{x_1}, \ldots b_{x_n}) = \sum_{I_j \in S} P(I_j) P(w_i|I_j) \prod_{m=1}^{n} P(b_m|I_j) \tag{2.14}$$

This method has been tested on Corel 5000 dataset and average precision and recall values are given as 0.10 and 0.09, respectively. Also, CMRM predicts 66 over 260 words in dataset. Since CMRM uses image feature segments instead of image features itself, performance of CMRM is dependent on clustering parameters and errors [9].

To improve performance of CMRM, CRM is proposed. CRM uses continuous features rather than *discrete blobs*. Therefore, it could be thought as continuous extension of CMRM. By using continuous features, clustering errors are eliminated. In this model, image I_x is represented as a set of regions ($I_x = \{r_{x_1}, \ldots, r_{x_n}\}$). These regions contains low level image features like pixel color. In this model, a function G maps region r_i to feature g_i ($G : r_i \rightarrow g_i$) like position, size, texture. By using these continuous features $P(w_i, I_x)$ is defined as follows:

$$P(w_i, I_x) \approx P(w_i, r_{x_1}, \ldots, r_{x_n}) = \sum_{I_j \in S} P(I_j) P(w_i|I_j) \prod_{m=1}^{n} \int P(r_m|g_m)\, P(g_m|I_j)\, \mathrm{d}g_m \,. \tag{2.15}$$

Using continuous feature instead of discrete *blobs* causes a performance improvement. Precision, recall and non-zero recall values for Corel 5000 dataset of CRM are reported as 0.16 and 0.19 and 107 over 260, respectively. However, CRM model still has two main problems [10]:

1. CRM uses automatic segmentation methods for generation of region. Thus, performance and process time of model are highly dependent on the quality of segmentation process. In order to avoid the errors of segmentation, CRM-Rectangles model [10] divides images into fixed-size rectangles instead of segments. This approach improves the performance significantly (precision, recall and non-zero recall values for Corel 5000 dataset are 0.22, 0.23 and 119 respectively).

2. The CRM model computes only the probabilities of the presence of words. However, absence of words is as important as presence for annotation quality.

MBRM [10] provides an improvement for previously mention CRM model. MBRM is based on CRM-Rectangles and reflects not only presence of words but also absence of words by using Multiple-Bernoulli model. Performance results for Corel 5000 dataset are reported as 0.24

for precision, 0.25 for recall and 122 for non-zero recall. The result shows that, consideration of words other than the annotated ones, increases annotation performance positively.

2.4.2 Probabilistic Latent Semantic Analysis Model

Probabilistic Latent Semantic Analysis, proposed by Hofmann [3] is a technique for analyzing latent topics in the documents. It assumes that, two similar words may have different or two different words have the same meaning according to the content of documents. Because of this assumption, directly assigning a word to a document could rise an ambiguity problem. To resolve this ambiguity, documents are represented with several topics. The words are assigned to the documents according to these topics. Each word $w \in W = \{w_1 \dots w_n\}$ belongs to a topic $z \in Z = \{z_1 \dots z_k\}$ and a document $d \in D = \{d_1 \dots d_m\}$ may contain several topics [11]. According to this approach, the probability of word w in document d is calculated by the following equation [3].

$$p(w, d) = P(d)P(w|d) = P(d) \sum_{z \in Z} P(w|z)P(z|d) \,, \tag{2.16}$$

where $P(d)$ is the probability of picking document d and parameters of $P(w|z)$ and $P(z|d)$ are calculated by using Expectation-Maximization algorithm [12]. In expectation step (E-step), $P(z|d, w)$ is computed by using previously estimate parameters, as follows,

$$P(z|d, w) = \frac{P(w|z)P(z|d)}{\sum_{z_i \in Z} P(w|z_i)P(z_i|d)} \,. \tag{2.17}$$

In maximization step (M-step), $P(w|z)$ and $P(z|d)$ is recalculated by using new $P(z|d, w)$. For unseen document, $P(z|d)$ is recalculated by using the previously learned $P(w|z)$ from the following equations;

$$P(w|z) = \frac{\sum_{d \in D} n(d, w)P(z|d, w)}{\sum_{w_i \in W} \sum_{d \in D} n(d, w_i)P(z|d, w_i)} \,, \tag{2.18}$$

$$P(z|d) = \frac{\sum_{w \in W} n(d, w)P(z|d, w)}{n(d)} \,, \tag{2.19}$$

where $n(d_w)$ is the count of word w in document d.

For unseen document d_{new}, partial version of Expectation-Maximization algorithm is used. In this method $P(w|z)$ is kept fixed in M-step and model maximizes $P(z|d_{new})$.

Monay and Perez adapted Probabilistic Latent Semantic Analysis (PLSA) to image annotation problem [11] and proposed a model called PLSA-Mixed [13]. In this model, images are

considered as documents. Then, the original model for $P(w_i|I_x)$ is employed. Precision performance of method for Corel 5000 dataset is reported as 0.12. The main problem of PLSA-Mixed is that an image (I_j) is represented as a concatenated vector of both visual (δ) and textual (w) features ($I_j = \{\delta_{j_1}, \ldots, \delta_{j_n}, w_{j_1}, \ldots, w_{j_m}\}$) and model gives equivalent importance to both visual and textual features when identifying latent topics. However, visual similarities do not indicate semantic similarities. To overcome this situation, Monay and Perez proposed a new model, namely PLSA-Words. [13]. In this model, two PLSA models are constructed, one for textual (2.20) and one for visual (2.21) features, as follows;

$$p(w, I_j) = P(I_j)P(w|I_j) = P(I_j) \sum_{z \in Z} P(w|z)P(z|I_j) \,, \tag{2.20}$$

$$p(\delta, I_j) = P(I_j)P(\delta|I_j) = P(I_j) \sum_{z \in Z} P(\delta|z)P(z|I_j) \,. \tag{2.21}$$

These models share same $P(z|I)$ distribution. The textual PLSA model estimates $P(w|z)$ and $P(z|I)$ by using training set. According to $P(z|I)$ distribution which is previously estimated by textual PLSA model, visual PLSA model calculates $P(\delta|z)$. For unseen image (I_x), visual PLSA model computes $P(z|I_x)$ and by using $P(z|I_x)$, textual PLSA model computes $P(w_i|I_x)$.

$$P(w_i|I_x) = \sum_{z \in Z} P(w_i|z)P(z|I_x) \tag{2.22}$$

These modifications improve performance of PLSA-Words and precision is increased to 0.16 for Corel 5000 dataset.

2.4.3 Hierarchical Image Annotation System Using Holistic Approach Model

Hierarchical Image Annotation System Using Holistic Approach Model (HANOLISTIC) [7], extracts low level visual features (such as color structure, homogeneous texture) from whole image. HANOLISTIC combines different annotators in two-level hierarchical ensemble learning architecture (Figure 2.1).

In the first level, called *Level-0*, for each low level visual descriptor δ_i, an annotator $Annotator^{\delta_i}$ is constructed. $Annotator^{\delta_i}$ annotates an input image I_x by using it's δ_{x_i} low level descriptor and outputs annotation result $A_x^{\delta_i}$. HANOLISTIC employs fuzzy k-nearest neighbor algorithm for annotators at the *Level-0*. The $Annotator^{\delta_i}$ assigns the possibility value of existence of the word w_j by using nearest k-neighbors training images.

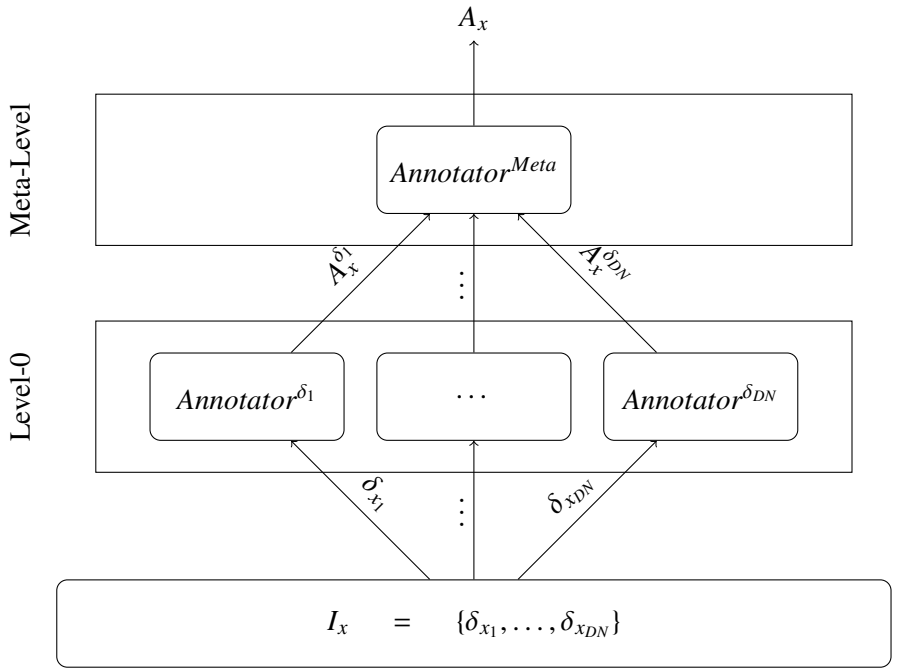


Figure 2.1: System architecture of HANOLISTIC.

Table 2.2: Performance of HANOLISTIC according to approaches used in *Meta-Level*.

	Precision	Recall	F-Score	NZC
Summation	0.39	0.22	0.28	103
Weighted Summation	0.35	0.24	0.28	113
Max Selection	0.26	0.20	0.22	97

In the second level, called *Meta-Level*, annotation results $A_x^{\delta_i}$ from annotators $Annotator^{\delta_i}$ are aggregated and final annotation A_x is proposed. Three different approaches is used at $Annotator^{Meta}$ [7]. First approach is named as *Summation Annotator*. *Summation Annotator* assumes all annotators in the *Level-0* are equally reliable. Thus, summation of annotation results of *Level-0* annotators is used for final annotation. Second approach is *Weighted Summation Annotator*. In this approach, the major assumption is that each annotator in the *Level-0* has different reliabilities. Thus, $Annotator^{Meta}$ assigns weight values to annotators in the *Level-0* according to their reliabilities and sums annotator results by using weights of their annotators weights. Third approach selects maximum possibility value of existence for a word among annotation result provided by *Level-0* annotators. Table 2.2 represents the performance of HANOLISTIC on Corel dataset according to approach used in *Meta-Level*.

Since every descriptor has strength and weakness to represent an image, every annotator has annotation strengths and weakness. By combining these annotation results in a hierarchical structure, weakness of an annotator is recovered by strength of another one and this property leads relatively good performance results compared to other well known annotators, for Corel 5000 dataset.

2.5 Automatic Image Annotation Refinement Techniques

As it is seen from the above techniques, the available annotation techniques are quite far from the practical needs, in terms of performances. These low performances need to be improved for many application domains. The available techniques suffer from the so called ”semantic gap” problem. In other words, an image annotation system annotates images with semantic words by using similarities between image descriptors. The gap between semantic words and low level descriptors causes unrelated and/or noisy words in annotation. These noisy words affect annotation quality and decrease performance of annotation method. One way to attack this problem is to add semantic similarities between words to annotation system. Refinement methods combine semantic similarities between words and annotation results provided by an annotator to remove noisy words. According to the model of relation between words, refinement methods could be divided into two categories, namely external and internal source based methods.

2.5.1 External Source Based Refinement Methods

External source based refinement methods extract semantic similarities between words from generic word databases. One of the well known example of these databases is WordNet [1], where the words are connected according to their semantic and lexical relations. Figure 2.2 [14] shows a sample hierarchy for the word *studio*.

Jin et al. [14] proposed a refinement method which uses WordNet. To refine the annotation result, relationship between annotation words are constructed according to their semantic similarities. Unrelated words are accepted as noisy words and removed from annotation. To calculate semantic similarities, this method uses several semantic distance measures defined on WordNet. According to Dempster-Shafer evidence combination theory, the semantic mea-

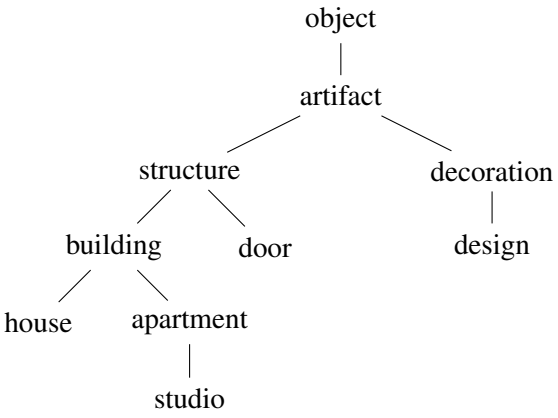


Figure 2.2: An example of WordNet hierarchy.

surements are combined and semantic similarity of two words is calculated. This method is applied to *Translation Model* and reported performance results shows that the suggested method improves precision value from 0.20 to 0.30, but recall value is decreased from 0.35 to 0.21.

Method, proposed by Jin et al. [14], has two main arguable points. First of all, possibility values in raw annotation are not used and effects of automatic annotator on final annotation is decreased [15]. However combining semantic similarities with possibility values provided by annotator for each word, could effect performance positively. Secondly, WordNet is focused on semantic relationship rather than contextual relationship. For example, according to WordNet, *mountain* and *rock* has stronger relationship than *mountain* and *lake*. However according to general usage on WWW, *mountain* and *lake* are more frequently used together and has stronger relationship than *mountain* and *rock* [16].

Alternatively, Wang and Cong [16] proposed a refinement method which extracts semantic similarities between words from Google search engine. Semantic similarities of words are calculated by using Normalized Google Distance [17]. Similarity between two words is calculated by using the following equation:

$$NGD(w_i, w_j) = \frac{\max\{\log f(w_i), \log f(w_j)\} - \log f(w_i, w_j)}{\log M - \min\{\log f(w_i), \log f(w_j)\}} , \tag{2.23}$$

where $f(w_i)$ number of web pages returned by Google for w_i and M is the index size of Google [16]. By using the conditional random field (CRF) model, this method also combines possibility values in raw annotation and word's semantic relations and calculate probability

of $P(w_i|I_x)$ as follows:

$$P(w_i|I_x) = \frac{e^{\psi(w_i,I_x;\alpha_1,\alpha_2)}}{\sum_{w_j \in W} e^{\psi(w_j,I_x;\alpha_1,\alpha_2)}} , \tag{2.24}$$

where

$$\psi(w_i, I_x; \alpha_1, \alpha_2) = \alpha_1 \omega^1(w_i, I_x) + \alpha_2 \sum_{w_j \in W} \omega^2(w_i, w_j) \tag{2.25}$$

$$\omega^1(w_i, I_x) = \log A_x^{raw}[w_i] \tag{2.26}$$

$$\omega^2(w_i, w_j) = \begin{cases} -\log NGD(w_i, w_j) & A_x^{raw}[w_i] \text{ and } A_x^{raw}[w_j] > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

and α_1, α_2 are weight parameters which are control balance between raw annotation and semantic similarities.

This method has been tested with Corel 5000 dataset. The results are provided for the most frequent 50 words of 374 words in dataset. This restricted refinement results provides 0.55 precision and 0.45 recall values, while selected raw annotator gives 0.45 and 0.40 respectively.

2.5.2 Internal Source Based Refinement Methods

The major drawback of using an external source is to model words in the dataset effectively. To overcome this difficulty, some of the methods extract semantic similarities between words from dataset. The method, proposed by Wang et al. [15], is one of the well known methods in this category. This method constructs a similarity matrix $R^{V_n \times V_n}$ which contains similarity values between words. Value of $r_{i,j}$ is calculated by similarity function $sim(w_i, w_j)$, as follows,

$$sim(w_i, w_j) = \frac{num(w_i, w_j)}{\min\{num(w_i), num(w_j)\}} , \tag{2.28}$$

where $num(w_i)$ is the number of image annotated with word w_i and $num(w_i, w_j)$ is the number of image annotated with both word w_i and w_j [15]. By using similarity matrix R and $V_n \times 1$ raw annotation vector A^{raw}, the method calculates a $V_n \times 1$ refined annotation vector $A^{refined}$ as follows:

$$A^{refined} = c(Identity - (1 - c)S)^{-1} A^{raw} , \tag{2.29}$$

where $Identity$ is $V_n \times V_n$ identity matrix, c is the control parameter and S is the similarity matrix.

This method is applied to CMRM and compared with WordNet based method proposed by Jin and performance results are represented at Table 2.3. As it can be seen from this table,

Table 2.3: Performance results of refinement method proposed by Wang

	Precision	Recall	F-Score
CMRM	0.36	0.57	0.44
WordNet based method	0.35	0.56	0.43
Method proposed by Wang	0.41	0.55	0.47

the method suggested by Wang outperforms the WordNet based method. The reason for this significant improvement may be attributed to the computation of local similarities, rather than using a generic dictionary.

2.6 Chapter Summary

In this chapter, we provide mathematical definition of image annotation and image annotation refinement problems. Also we explain common performance measurement metrics which are used by automatic image annotation methods. Then, we explain the state of art image annotation and image annotation refinement techniques.

CHAPTER 3

A NOVEL REFINEMENT METHOD FOR AUTOMATIC IMAGE ANNOTATION SYSTEMS

In this chapter, we propose a novel refinement method for image annotation systems. In the first section, we give an overview of the proposed refinement method and introduce the major components of proposed system. In the following sections, we aim to explain the components of the proposed method. We use an example to explain components effectively.

3.1 Overview of The Proposed Refinement Method

The major motivation of this study is to employ the semantic similarities between words to refine and improve the annotation results obtained at the output of an existing annotator. This task is achieved by introducing extra information to the annotation system by measuring the semantic similarities between the words of the image database. The problem of refinement is then reduced to make a formal definition of *semantic similarity*. The refinement procedure is based on the definition of semantic similarities.

The proposed method refines the raw annotation result A_x^{raw} which is provided by an automatic image annotator $Annotator^{raw}$ for a newly seen image I_x. Throughout this chapter, we call automatic image annotator as $Annotator^{raw}$ and the annotation result of this annotator is called as A^{raw}. A^{raw} is a set which contains possibility values of existence of words in the vocabulary, which contains all words used in the image dataset.

Definition 1 (Possibility value of existence) *The intermediate output A_x^{raw} of $Annotator^{raw}$*

for an image I_x is defined as

$$A_x^{raw} = \{p_{1,x} \dots p_{VN,x}\}\,, \tag{3.1}$$

where VN is the size of vocabulary and $p_{i,x}$ is the possibility value of existence of word w_i in image I_x. In other words $p_{i,x}$ shows the possibility of the word w_i in the image I_x. Possibility value of existence can take real number values between 0.0 and 1.0. 0.0 indicates lowest possibility and 1.0 indicates highest possibility of existence. ■

The intermediate output A^{raw} for an image I_x is the input to the suggested refinement process. The workflow of the proposed system is summarized as follows:

Initially, an image I_x is annotated by $Annotator^{raw}$. The raw annotation result A_x^{raw} is given as an input of the proposed refinement method. The proposed refinement method contains three main components as illustrated in Figure 3.1. The first component calculates relationship between words from dataset S and keeps them in fuzzy sets (μ_{w_i}), which are constructed for each word w_i in the vocabulary. The second component generates candidate annotations $A_x^{candidate_{w_i}}$ for each word w_i in raw annotation A_x^{raw} by using relationship between words. Third component selects the most suitable candidate annotation result and proposes a refined annotation result $A_x^{refined}$. Values in $A_x^{refined}$ are sorted in descending order and top WN words with highest possibility value are assigned to $W_x^{refined}$ for image I_x. Following sections explains the components of proposed method, briefly introduced above.

3.2 Finding Relations Between Words

The proposed refinement method is based on the utilization of relationship among words, covered under a meaningful image. In the literature, well known refinement methods find related words in raw annotation result and remove unrelated words from the annotation result. For example, let us assume that an image contains *jet, plain, sky* and suppose that an automatic image annotator annotates this image with *jet, plain, sky* and *train*. It is obvious that there is a strong relationship among *jet, plain, sky*, but not with *train*, so the word *train* is assumed to be noise for this image.

Assumption 1 (Relatedness of words) *Meaningful images contain objects which are semantically related to each other.* ■

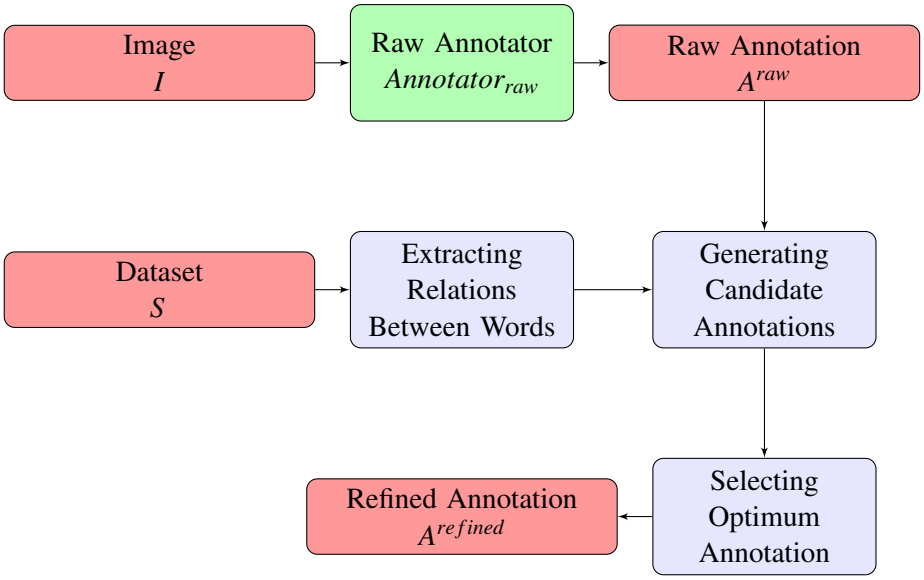


Figure 3.1: Block diagram representation of the proposed refinement method.

The most crucial point in Assumption 1 is to define a measure of relatedness of two words. Before defining semantic relatedness formally, we need to answer two major questions: First of all, how can we decide which objects are related to each other? Secondly, how can we decide an image is meaningful? As described in Section 2.5, relationship between words could be extracted from different resources, such as WordNet, WWW or dataset itself. However, defining a meaningful image is not a simple task. Meaningfulness is highly dependent on dataset. For example, for a dataset which contains medical images, an image which contains *grass* is meaningless so if an annotator annotates an image with *grass* we can easily indicate that *grass* is unrelated word.

In the scope of this thesis, semantic relationships between words are extracted from dataset. If two words occur in the same images, strong relation between these words is inferred. To calculate the relation between words, first of all, co-occurrence statistics of words are calculated and kept under a table like Table 3.1. By using co-occurrence statistics, semantic similarity matrix R is calculated. The entries, $r_{i,j}$ of matrix R represents the relationship between words w_i and w_j,

Definition 2 (Semantic similarity matrix R) *Semantic similarity matrix R is a symmetric*

Table 3.1: Co-occurrence statistics of words.

	w_0	w_1	...	w_n
w_0	$num(w_0)$	$num(w_0, w_1)$	...	$num(w_0, w_n)$
w_1	$num(w_0, w_1)$	$num(w_1)$	...	$num(w_1, w_n)$
...	...	...	...	...
w_n	$num(w_0, w_n)$	$num(w_1, w_n)$	...	$num(w_n)$

$num(w_i)$: Number of image annotated with word w_i.
$num(w_i, w_j)$: Number of image annotated with both word w_i and w_j.

$VN \times VN$ matrix which contains relationship between words in vocabulary V.

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,VN} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,VN} \\ \vdots & \vdots & \ddots & \vdots \\ r_{VN,1} & r_{VN,2} & \cdots & r_{VN,VN} \end{bmatrix} \tag{3.2}$$

The entries, $r_{i,j}$, represents relationship between words w_i and w_j and should satisfy following properties.

Property 1 *The degree of the relationship between two words is represented by a real number values between 0.0 and 1.0. 0.0 indicates no similarity at all and 1.0 indicates highest similarity. In other words,*

$$0.0 \leq r_{i,j} \leq 1.0 \, . \tag{3.3}$$

Property 2 *A word has a high relationship with itself and*

$$r_{i,i} = 1.0 \, . \tag{3.4}$$

Property 3 *Relationship between w_i and w_j has to be equal to relationship between w_j and w_i*

$$r_{i,j} = r_{j,i} \, . \tag{3.5}$$

■

To define relationships between words, probabilistic or fuzzy framework could be used. We explore these frameworks for defining the relationships between words under following subsections.

Table 3.2: Co-occurrence statistics of words in the subset of Corel 5000.

	sky	water	tree	people	branch	grass	bird	building	horizon	nest	fly
sky	883	190	221	41	13	71	9	117	12	1	0
water	190	1004	143	159	2	61	20	70	10	5	0
tree	221	143	854	59	25	102	34	75	2	13	6
people	41	159	59	670	0	11	2	67	6	0	0
branch	13	2	25	0	78	6	33	0	0	17	1
grass	71	61	102	11	6	446	23	9	1	10	0
bird	9	20	34	2	33	23	179	0	0	70	11
building	117	70	75	67	0	9	0	408	36	0	0
horizon	12	10	2	6	0	1	0	36	53	0	0
nest	1	5	13	0	17	10	70	0	0	71	1
fly	0	0	6	0	1	0	11	0	0	1	11

3.2.1 Probabilistic Framework to Define Semantic Relationship

Probabilistic framework provides us a useful tool to define the relationships between words. Prior and Joint probabilities are approximated by the relative frequencies as defined below. The prior probability of a word $P(w_i)$ represent $r_{i,i}$ entry of matrix R;

$$r_{i,i} = P(w_i) \approxeq \frac{num(w_i)}{\sum_{w_k \in V} num(w_k)} , \tag{3.6}$$

whereas, the joint probability between two words $P(w_i, w_j)$ represents $r_{i,j}$;

$$r_{i,j} = P(w_i, w_j) \approxeq \frac{num(w_i, w_j)}{\sum_{w_k \in V} num(w_k)} . \tag{3.7}$$

However, Equation (3.6) breaks the Property 2. To give a clear explanation, let us investigate the intuitive validity of the Equations (3.6) and (3.7) by the following example: Suppose that we use a subset of Corel 5000 dataset. Table 3.2 contains co-occurrence values of Corel 5000 dataset. Co-occurrence values are calculated by using 4500 images. As it can be seen from this table, the relative frequencies of words are not computed by using sufficiently large number of samples, due to the lack of manual annotation for some words. Specifically, while some words are annotated very frequently (such as "water", where $num(water) = 1004$), some other words are relatively rare (such as "fly", where $num(fly) = 11$).

Now, suppose that we calculate $r_{bird,bird}$ on dataset. If we apply Equation (3.6) to calculate $r_{bird,bird}$, we get 0.04, but according to Property 2 "bird" has to be the strongest relationship with itself.

Equation (3.7) obeys the Property 3, but could not reflect relationship between words. For example; according to Equation (3.7) relationship between "bird" and "fly" is 0.002 (= $r_{bird,fly}$ = $P(bird, fly) = \frac{11}{4500}$). This value indicates a weak relationship. However, all images which contain "fly" also contain "bird". Intuitively, there should be strong relation between these two words.

Another problem of representing the relationship among the words by joint probability densities, emerge when less frequent words in the vocabulary appears to be contextually important for an image. Alternatively, high frequent words are assigned to almost every image, regardless of content of image, represented by the rest of the annotated words.

To overcome these limitations, we can replace the joint probabilities by the conditional probability of words. The following relative frequencies can be used to approximate the conditional probabilities:

$$r_{i,i} = P(w_i|w_i) \cong \frac{num(w_i)}{num(w_i)} \tag{3.8}$$

and

$$r_{i,j} = P(w_j|w_i) \cong \frac{num(w_i, w_j)}{num(w_i)} \ . \tag{3.9}$$

Equation (3.8) meets the Property 2. However, Equation (3.9) is highly dependent on the dataset. If the words in dataset are well distributed, which means that the number of occurrences are sufficiently high and close to each other for all words, this equation satisfies Property 3. Unfortunately, datasets like Corel 5000 do not satisfy this property. For example; if we use Equation (3.9) to compute the relationship among words, we obtain Table 3.2. In this table, we calculate $r_{bird,fly} = \frac{11}{179} = 0.06$, which indicates a weak relationship. On the other hand, $r_{fly,bird} = \frac{11}{11} = 1.00$, which indicates a relatively strong relation compared to $r_{bird,fly}$. These relative frequencies do not make an intuitive sense.

Alternatively, we can use following joint probability approximation to calculate the relationship between words.

$$r_{i,i} = P(w_i) \cong \frac{num(w_i)}{num(w_i)} \ , \tag{3.10}$$

and

$$r_{i,j} = P(w_i, w_j) \cong \frac{num(w_i, w_j)}{(num(w_i) + num(w_j)) - num(w_i, w_j)} \ . \tag{3.11}$$

This joint probability approximation obeys the Property 2 and the Property 3. However, it could not reflect relationship between words properly. For example; according to Equation

(3.11) $r_{bird,fly} = P(bird, fly) = \frac{11}{(179+11)-11} = 0.06$, but, there is a strong relation between "bird" and "fly" since all images which contain "fly" also contain "bird".

3.2.2 Fuzzy Framework to Define Semantic Relationship

Due to the problems mentioned in the previous section, probabilistic framework provides us a limited success and improvement in the refinement methods. Another way of representing the relationships between words is to employ the fuzzy sets and compute the membership values.

In the fuzzy framework, we construct a fuzzy set μ_{w_i} for each word w_i in the vocabulary. Formal definition of fuzzy set μ_{w_i} is given as follows:

Definition 3 (Fuzzy set μ_{w_i}) *The fuzzy set μ_{w_i} for a given word w_i is defined as,*

$$\mu_{w_i} = \{\mu_{w_i}(w_0), \ldots \mu_{w_i}(w_j), \ldots \mu_{w_i}(w_n)\} . \tag{3.12}$$

The elements of μ_{w_i} indicates the Membership value $\mu_{w_i}(w_j)$ which gives the relationship between the word w_i and the word w_j. Membership values of the fuzzy set μ_{w_i} are calculated from the following equation:

$$\mu_{w_i}(w_i) = \frac{num(w_i)}{num(w_i)} \tag{3.13}$$

For calculating $\mu_{w_i}(w_j)$, we use the similarity function proposed by Wang at [15].

$$\mu_{w_i}(w_j) = \frac{num(w_i, w_j)}{\min(num(w_i), num(w_j))} \tag{3.14}$$

■

Equation (3.13) is used to calculate the entries $r_{i,i}$ of relation matrix R. It is easily verifiable that this equation satisfies Property 2. For calculating $r_{i,j}$ entries in matrix R, Equation (3.14) is used. This equation not only satisfies the Property 3, but also solves unbalanced word distribution problem of the dataset.

For example, Table 3.3 shows the relationship values of words which are calculated by using Equations (3.13) and (3.14). Each row in this table could be considered as a the fuzzy set μ_{w_i} of the word w_i. As you remember in the probabilistic framework the relationship between "bird" and "fly" could not be modeled in a meaningful way. However, by using Equation (3.14), strong relation between "bird" and "fly" is established.

Table 3.3: Matrix R calculated according to Table 3.2 by using Equations (3.13) and (3.14).

	sky	water	tree	people	branch	grass	bird	building	horizon	nest	fly
sky	1,00	0,22	0,26	0,06	0,17	0,16	0,05	0,29	0,23	0,01	0,00
water	0,22	1,00	0,17	0,24	0,03	0,14	0,11	0,17	0,19	0,07	0,00
tree	0,26	0,17	1,00	0,09	0,32	0,23	0,19	0,18	0,04	0,18	0,55
people	0,06	0,24	0,09	1,00	0,00	0,03	0,01	0,16	0,11	0,00	0,00
branch	0,17	0,03	0,32	0,00	1,00	0,08	0,42	0,00	0,00	0,24	0,09
grass	0,16	0,14	0,22	0,03	0,08	1,00	0,13	0,02	0,02	0,14	0,00
bird	0,05	0,11	0,19	0,01	0,42	0,13	1,00	0,00	0,00	0,99	1,00
building	0,29	0,17	0,18	0,16	0,00	0,02	0,00	1,00	0,68	0,00	0,00
horizon	0,23	0,19	0,04	0,11	0,00	0,02	0,00	0,68	1,00	0,00	0,00
nest	0,01	0,07	0,18	0,00	0,24	0,14	0,99	0,00	0,00	1,00	0,09
fly	0,00	0,00	0,55	0,00	0,09	0,00	1,00	0,00	0,00	0,09	1,00

3.3 Generating Candidate Annotations

If we recall to our major Assumption 1, the relationship between related words should have stronger than unrelated words. If we could know correct word assignments in annotation result, noisy words would be eliminated according to these correct words. However, as the nature of the annotation problem, correct word assignments are unknown. The only available information about words is their possibility values of existence, listed in raw annotation result A^{raw} provided by $Annotator^{raw}$. Thus, for each word w_i in raw annotation result a candidate annotations, $A^{candidate_{w_i}}$, is generated.

To represent and interpret relationship between words, a weighted relation graph $G = (V, E)$ structure is constructed. The structure of this graph is defined below:

Definition 4 (Weighted Relation Graph $G = (V_{center}, V, E)$) *Graph G is a weighted undirected relation graph, which contains a center vertex V_{center}, set of vertices V which are placed around center vertex and set of edges E which contains edges between center vertex and rounding vertices.*

The center vertex V_{center} represents a set of words S_{words} and a fuzzy set $\mu_{V_{center}}$ which defines the degree of relationship between the center vertex and words in the vocabulary.

$$V_{center} = \{\mu_{V_{center}}, S_{words}\} \tag{3.15}$$

$$S_{words} = \{w_{1,i} \dots w_{ik,i}\}\ w_{z,i} \in V \tag{3.16}$$

$$\mu_{V_{center}} = \{\mu_{V_i}(w_1), \dots \mu_{V_i}(w_{VN})\} \tag{3.17}$$

Other vertices of relation graph G is defined in the set V.

$$V = \{V_1 \dots V_n\} \tag{3.18}$$

For each word w_i in the vocabulary, a vertex V_i is constructed. These vertices are placed around of center vertex.

All edges in relation graph G is defined in the set E.

$$E = \{E_1 \dots E_n\} \tag{3.19}$$

E_i defines the edge between center vertex V_{center} and the vertex V_i. The Weight of this edge is assigned as follows;

$$EdgeWeight_i = \mu_{V_{center}}(w_i)\ . \tag{3.20}$$

■

Definition 5 (Merge operation on $G = (V_{center}, V, E)$) *Merge opertion merges a vertex $V_i \in V$ with V_{center}. After merge operation a new relational graph $G' = (V'_{center}, V', E')$ is generated. Merging operation is defined as follows;*

$$V'_{center} = \{\mu'_{V_{center}}, S'_{words}\} \tag{3.21}$$

$$S'_{words} = \{S_{words} \cup w_i\} \tag{3.22}$$

$$\mu'_{V_{center}} = \{\mu_{V_{center}} \cap \mu_{w_i}\} \tag{3.23}$$

where,

$$\mu'_{V_{center}}(w_k) = \min(\mu_{V_{center}}(w_k), \mu_{w_i}(w_k))\ , \tag{3.24}$$

$$V' = V - V_i\ , \tag{3.25}$$

and

$$E' = E - E_i\ . \tag{3.26}$$

■

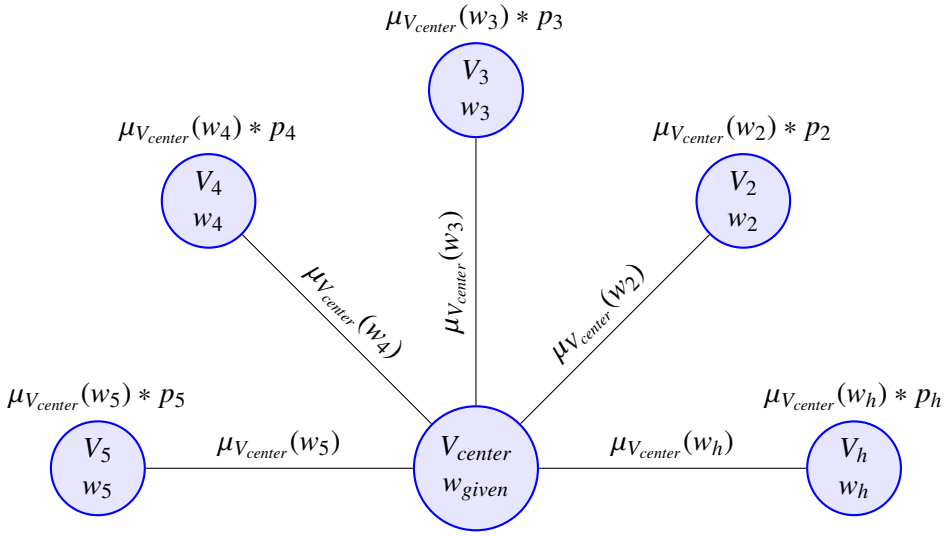


Figure 3.2: Relational Graph $G = (V_{center}, V, E)$ for w_{given}.

For each word in raw annotation result a candidate annotation is generated. First of all, for given word w_{given} a relational graph is created. A center vertex V_{center} is generated and w_{given} is places in center vertex. Fuzzy set of w_{given} is assigned to center vertex's fuzzy set.

$$V_{center} = \{\mu_{V_{center}}, S_{words}\}$$

$$S_{words} \leftarrow w_{given}$$

$$\mu_{V_{center}} \leftarrow \mu_{w_{given}}$$

Figure 3.2 shows the center vertex V_{center} with the given word w_{given}. Remaining words, w_i ($\in A^{raw}, w_i \neq w_{given}$) in raw annotation result are placed in vertices V_i around the center vertex.

Surroundings vertices (V_i) are connected with center vertex (V_{center}) with edges E ($\{(V_{center}, V_i) \in E, V_i \neq V_{center}\} \wedge \{(V_i, V_j) \notin E, V_i, V_j \neq V_{center}\}$). Center vertex, V_{center} has a fuzzy set $\mu_{V_{center}}$ to represent the relation with surrounding vertices. In other words, the value of $\mu_{V_{center}}(w_i)$ represents the weight of the edge between V_{center} and V_i. For each vertex V_i, a vertex weight $VertexWeight_i$ is calculated by using Equation (3.27). High vertex weight indicates strong relation between center vertex and corresponding vertex.

$$VertexWeight_i = \mu_{V_{center}}(w_i) * p_i \tag{3.27}$$

where p_i is the possibility value of existence of word w_i which is provided by A^{raw}.

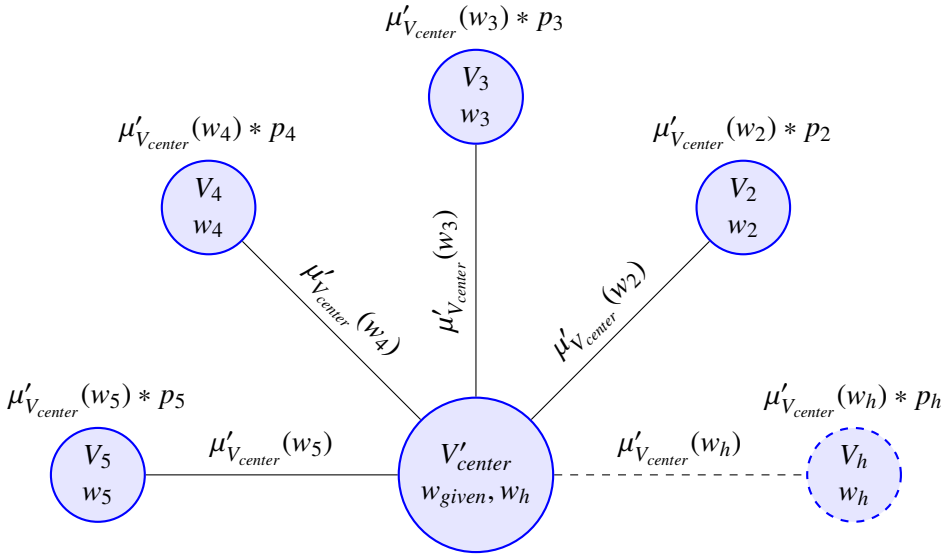


Figure 3.3: Relational Graph $G' = (V'_{center}, V', E')$ for w_{given} and w_h.

By using Equation (3.27), we could re-evaluate possibility values of existence of rest of the words in the raw annotation according to given word w_{given}. For example, if a word w_i has a high possibility value of existence (which means p_i is close to 1.0) in raw annotation result A^{raw} and relationship between center vertex is strong (which means value of $\mu_{V_{center}}(w_i)$ is close to 1.0), than the word w_i is reconsidered in final annotation. However, if a word has a high possibility value of existence in raw annotation, but weak relation with center vertex, it may be eliminated in the final annotation. In other words, the center vertex represents one of the correctly annotated words, then a noisy word in automatic annotation system may be eliminated by this equation.

After the relational graph $G = (V_{center}, V, E)$ for w_{given} is constructed, candidate annotation is generated according to this initial graph. Vertex with highest weight, let us name it V_h, is merged with center vertex (Figure 3.3) and a new relational graph $G' = (V'_{center}, V', E')$ is generated. After merge operation V'_{center} is generated and replaces V_{center} vertex of G graph in G' graph. Newly generated graph G' has the following center vertex;

$$V'_{center} = \{\mu'_{V_{center}}, S'_{words}\}$$

where

$$S'_{words} \leftarrow S_{words} \cup w_h$$

and

$$\mu'_{V_{center}} \leftarrow \mu_{V_{center}} \cap \mu_{w_h}$$

We employ fuzzy intersection operation to construct fuzzy set of $\mu'_{V_{center}}$. Intersection operation on fuzzy sets are defined as follows [18];

$$\mu'_{V_{center}}(w_k) = \min(\mu_{V_{center}}(w_k), \mu_{V_h}(w_k)) \; . \tag{3.28}$$

The reason of intersection is that now center vertex has two words and surrounding words should have to be related with both of these two words. In other words, newly added word restricts the relationship between center vertex and surrounding vertices, so that the fuzzy set of center vertex should be updated according to this new word. Weight of edges and vertex weights are calculated according to this new fuzzy set of center vertex.

The operation mentioned above proceeds until there is no remaining vertex with non-zero vertex weight. Words in center vertex are sorted according to append order and this sorted word list is proposed as a candidate annotation $A^{candidate_{w_{given}}}$. The Algorithm 1 represents operation flow as a pseudo code.

In order to clarify the process of candidate annotation generation, let us use the following example. Consider the image of Figure 3.4 as an input to the graph generation process of Algorithm 1. The input image is annotated by human annotators with the words *birds, branch, nest* words. As a raw annotator, HANOLISTIC is employed for this particular example. Top ten words in raw annotation result of HANOLISTIC with highest possibility value of existence which is listed in Table 3.4. For extracting relationship between words the same subset of Corel 5000 dataset which is represented at Table 3.2 and Table 3.3 is used as words relationship matrix R.

For the input image (Figure 3.4), HANOLISTIC correctly predicts *birds* and *nest* while it also assigns high possibility value of existence to words which do not exist in image like *sky, building* and *water*. If we investigate the reason for the high possibility values for these words, we observe that the unbalanced distribution of word in dataset effects annotation quality and words with high occurrence frequency get more chance to be assigned than low frequent words. For example, "water" occurs % 22 percent of images in training set. However, HANOLISTIC assigns "water" to % 68 of images in test set. To attack this problem, proposed refinement method uses relations between words, which are extracted from dataset as

Algorithm 1 Generating candidate annotation for the given word w_{given}.

Require: For image I, a raw annotation A^{raw}.

$A^{raw} = \{p_1, \ldots, p_j, \ldots, p_{VN}\}$

p_j : Possibility value of word w_j in raw annotation

$G = (V, E)$, $V \leftarrow \emptyset$, $E \leftarrow \emptyset$

$V_{center} \leftarrow w_{given}$

$V \leftarrow V \cup V_{center}$

$\mu_{V_{center}} \leftarrow \mu_{w_{given}}$ {//Center vertex has a fuzzy set to represent relation between surrounding vertices.}

List $T \leftarrow \emptyset$

$T \leftarrow T \cup w_{given}$

for all $w_j \in VOC$ $j \neq assumed$ **do**

  $V_j \leftarrow w_j$,

  $V \leftarrow V \cup V_j$

end for

while $\exists V_j \in V$, $j \neq center$ **do**

  for all (V_{center}, V_j), $V_j \in V$ **do**

    $E \leftarrow E \cup (V_{center}, V_j, \mu_{V_{center}}(w_j))$

  end for

  for all V_j **do**

    $VertexWeight_{V_j} \leftarrow \mu_{V_{center}}(V_j) * p_j$

  end for

  $MaxWeightedVertexIndex \leftarrow max_j(VertexWeight_{V_j})$,

  $V \leftarrow V - V_{MaxWeightedVertexIndex}$

  $V_{center} \leftarrow V_{center} \cup w_{MaxWeightedVertexIndex}$

  $\mu_{V_{center}} \leftarrow \mu_{V_{center}} \cap \mu_{w_{MaxWeightedVertexIndex}}$

  $T \leftarrow T \cup w_{MaxWeightedVertexIndex}$

  $E \leftarrow \emptyset$

end while

$A^{candidate_{w_{given}}} \leftarrow T$ {//T contains list of words sorted according to selection order}

return $A_{candidate^{w_{given}}}$

birds, branch, nest

Figure 3.4: A manually annotated input image used for the example run.

Table 3.4: For Figure 3.4, top ten words with highest possibility value of existence assigned by HANOLISTIC.

Word	p_j
bird	1,000
sky	0,812
nest	0,790
grass	0,568
building	0,540
water	0,523
horizon	0,462
tree	0,400
people	0,345
branch	0,328

described in Section 3.2. By using relationship between words, proposed refinement method could address word distribution instability in dataset and reflects relationship between words.

During generation of the candidate annotations, for each word in raw annotation, a candidate annotation is constructed. Figures 3.5, 3.6, 3.7 illustrates first three cycles of generating candidate annotation for the word *nest*. At first cycle (Fig. 3.5), the relation graph G is constructed according to the given word *nest*. The given word is placed on the center vertex of graph and other words are placed around this center. Weight of edges between center vertex and rounding vertices are assigned by using fuzzy set of center node ($\mu_{V_{center}}$), as follows

$$V_{center} = \{\mu_{V_{center}}, S_{words}\} ,$$

$$S_{words} \leftarrow nest ,$$

$$\mu_{V_{center}} \leftarrow \mu_{nest} .$$

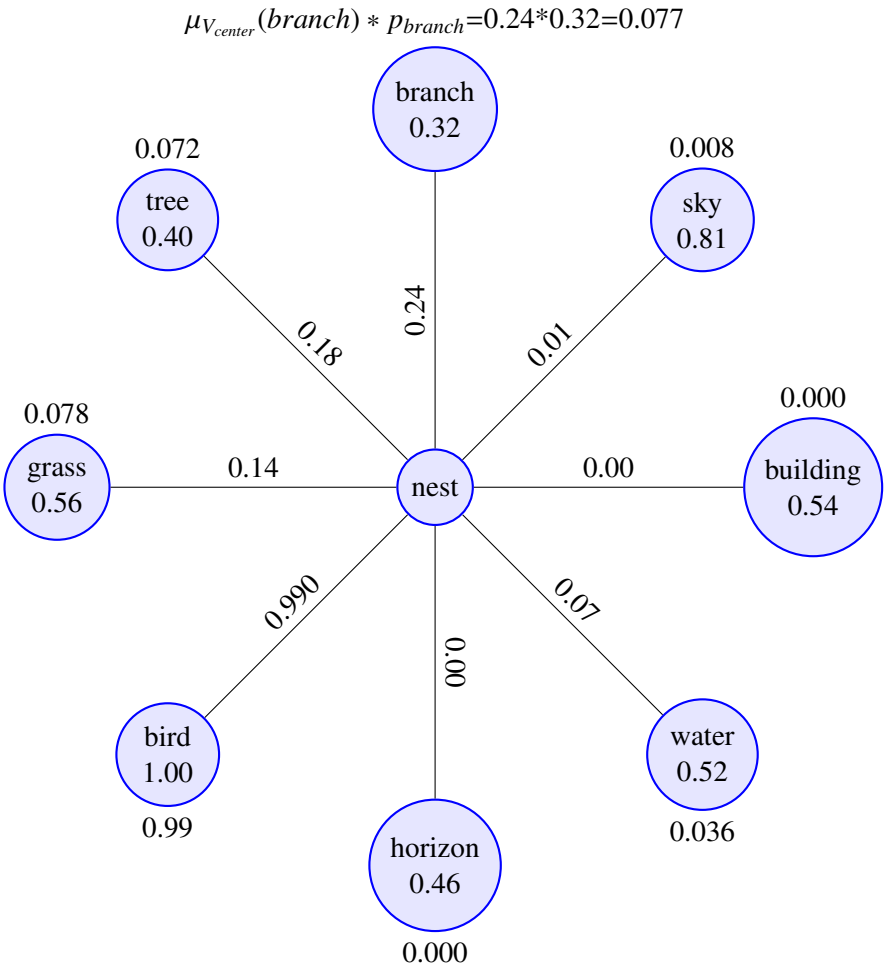
Vertex weights are calculated by using Equation (3.27). As a result, proposed method assigns higher vertex weights to related words and keeps them important, like *bird*, while assigns lower vertex weights to unrelated words, like *sky, horizon* and *water*. In the second cycle (Fig. 3.6), vertex with highest vertex weight, *bird*, is merged with center vertex and new graph G' is generated, with the following vertex at the center;

$$V'_{center} = \{\mu'_{V_{center}}, S'_{words}\} ,$$

$$S'_{words} \leftarrow S_{words} \cup bird ,$$

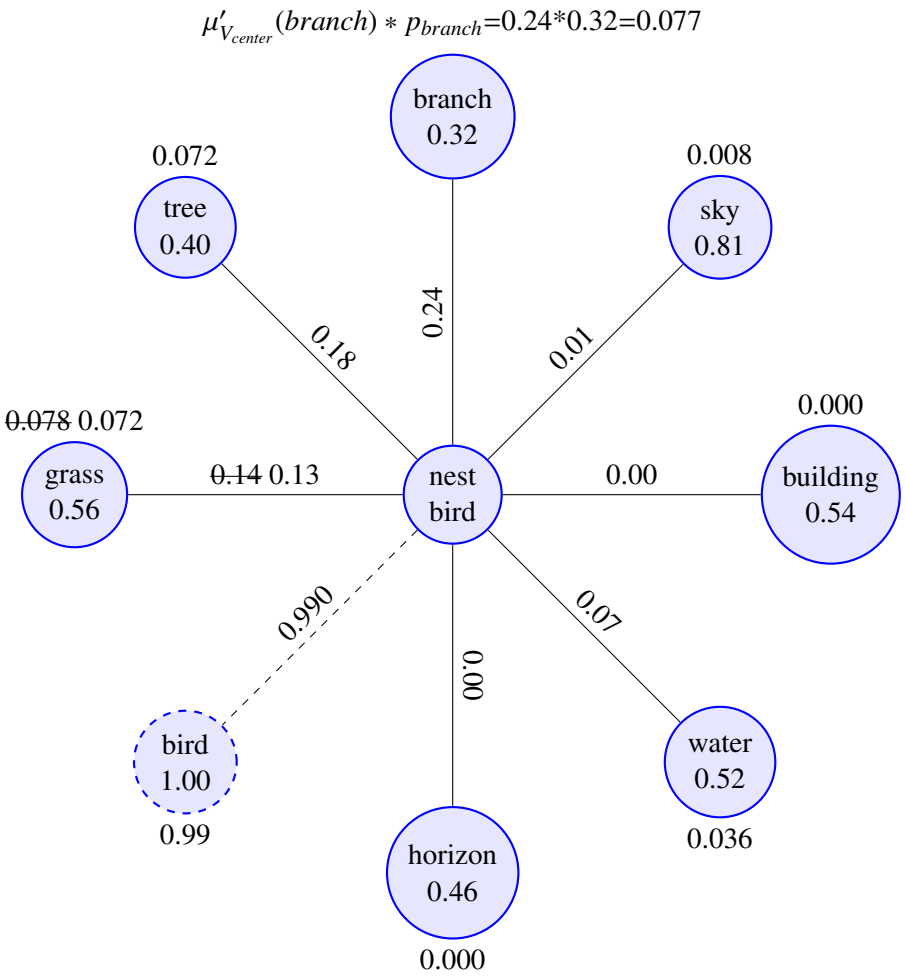
$$\mu'_{V_{center}} \leftarrow \mu_{V_{center}} \cap \mu_{bird} .$$

Adding *bird* to the center vertex, decreases the relation between center vertex and rounding vertices. At the first cycle, we have only one word, *nest*, to define relationships, but at the second cycle we have two words to define relationships. For example, according to fuzzy set of *nest*, relationship between *nest* and *grass* is 0.14. However, according to fuzzy set of *bird*, relationship between *bird* and *grass* is 0.13.The edge weight between center vertex and vertex which contains *grass* is updated by using new center vertex fuzzy set and because of this update the weight of *grass* reduced. This reduction allows us to consider the word *branch* for annotation during the refinement process.



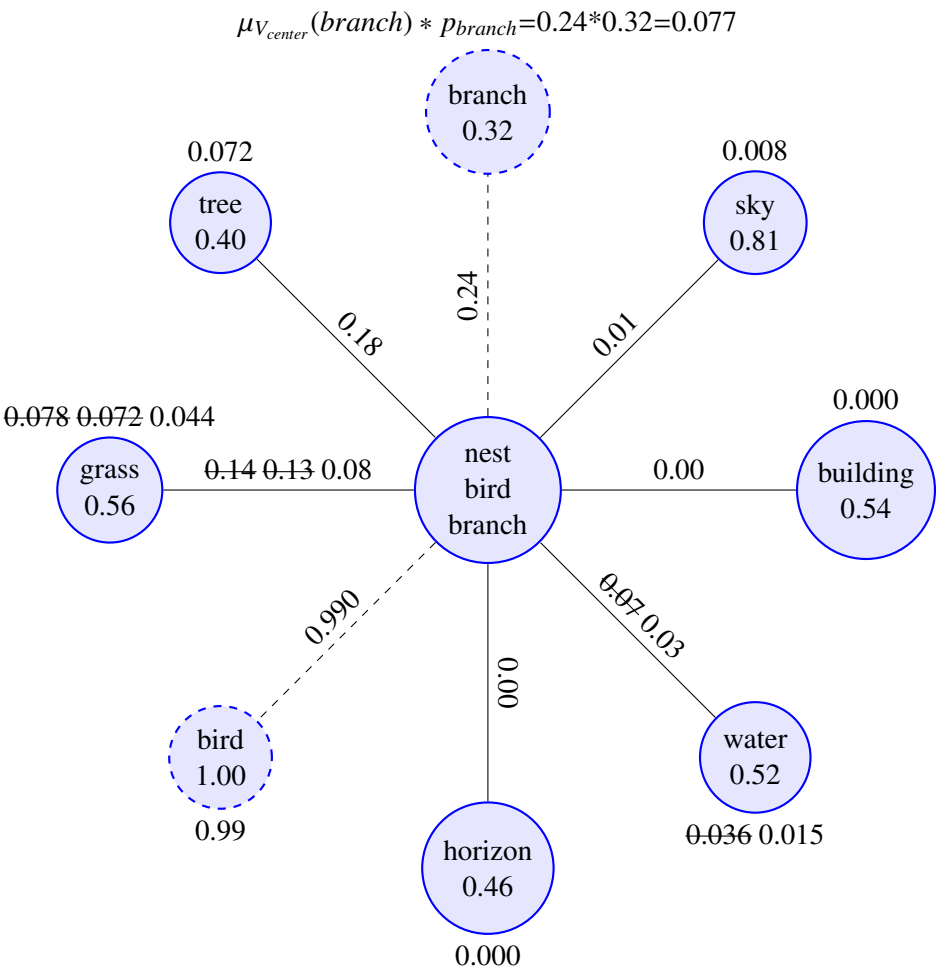
	sky	water	tree	people	branch	grass	bird	building	horizon	nest	fly
μ_{nest}	0.01	0.07	0.18	0.00	0.24	0.14	0.99	0.00	0.00	1.00	0.09
$\mu_{V_{center}}$	0.01	0.07	0.18	0.00	0.24	0.14	0.99	0.00	0.00	1.00	0.09

Figure 3.5: Graph G constructed for the word *nest* at the center vertex in the example run.



	sky	water	tree	people	branch	grass	bird	building	horizon	nest	fly
μ_{nest}	0.01	0.07	0.18	0.00	0.24	0.14	0.99	0.00	0.00	1.00	0.09
μ_{bird}	0.05	0.11	0.19	0.01	0.42	0.13	1.00	0.00	0.00	0.99	1.00
$\mu'_{V_{center}}$	0.01	0.07	0.18	0.00	0.24	0.13	1.00	0.00	0.00	1.00	0.09

Figure 3.6: Graph G' constructed for given words *nest* and *bird* in the example run.



	sky	water	tree	people	branch	grass	bird	building	horizon	nest	fly
μ_{nest}	0.01	0.07	0.18	0.00	0.24	0.14	0.99	0.00	0.00	1.00	0.09
μ_{bird}	0.05	0.11	0.19	0.01	0.42	0.13	1.00	0.00	0.00	0.99	1.00
μ_{branch}	0.17	0.03	0.32	0.00	1.00	0.08	0.42	0.00	0.00	0.24	0.09
$\mu_{V_{center}}$	0.01	0.03	0.18	0.00	1.00	0.08	1.00	0.00	0.00	1.00	0.09

Figure 3.7: Third cycle of generating candidate annotation in the example run.

Table 3.5: Generated candidate annotations for the example run.

w_{given}	$A^{candidate_{w_{given}}}$
birds	birds, nest, branch, tree, grass, stick, baby, water, sky, people
sky	sky, buildings, skyline, water, birds, stick, tree, baby, ground, nest
nest	nest, birds, branch, tree, grass, baby, water, sky, ground, rocks
grass	grass, sky, tree, water, birds, rocks, cubs, grizzly, baby, rodent
buildings	buildings, skyline, sky, water, people, tree, grass, rocks, bear, clouds

These cycles proceed until there is no vertex with non-zero vertex weight around center vertex. Table 3.5 contains five candidate annotations, which are generated according to first five words in Table 3.4. As clearly seen in this table, proposed method extracts the related words with given word and generates a set of candidate annotations for each word.

3.4 Selecting Optimum Annotation

After generating candidate annotation sets for each word in the raw annotation result of an image, there are as many candidate annotation as number of words in this raw annotation, so one of them should be selected as an optimum solution and re-assigned to the image. According to Assumption 1, in an ideal case, words in annotation result should be related with each other. To measure the rate of relatedness of a candidate annotation result, weight of maximum weighted clique is used.

Clique, C, is a subset of vertices V for given graph $G = (V, E)$, where every two vertices of C $(x, y \in C, x \neq y)$ are connected by an edge $((x, y) \in E)$. Figure 3.8 illustrates a clique for graph G. Maximum weighted clique is a clique in a given weighted graph with sum of vertices weight is maximum [19].

In order to measure the rate of relatedness of a candidate annotation, a fuzzy graph $G = (V, E)$ is constructed as shown in Figure 3.8. Each word, w_i, in candidate annotation is placed in vertices V_i. Weight of edge between two vertices $(V_i, V_j) \in E$ is equal to $\mu_{w_i}(w_j)$. Weight of a node is calculated by using Equation 3.27. In this study, we assume first order connectivity for fuzzy graph G which consider only two vertices to be connected if there is a non-zero membership value. Algorithm 2 provides the pseudo-code of the weight calculation process. Weight of maximum weighted clique is assigned to candidate annotation rating. The candidate

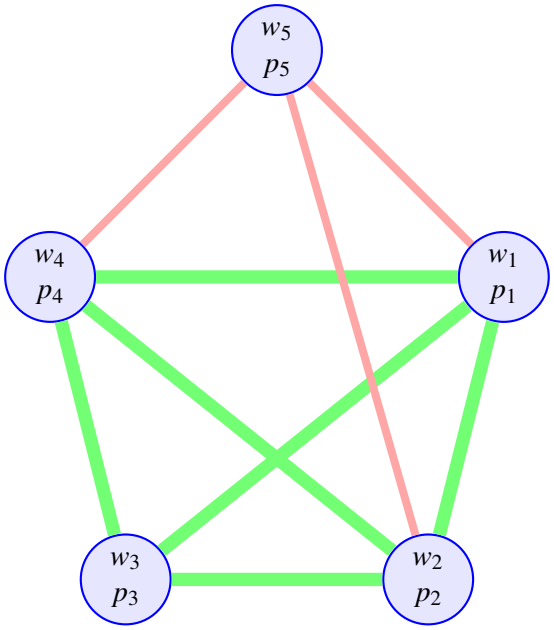


Figure 3.8: Clique $C = \{w_1, w_2, w_3, w_4\}$ of $G = (V, E)$.

Table 3.6: Ratings of generated candidate annotations for the example run

Candidate Annotation	Rating
birds, nest, branch, tree, grass, stick, baby, water, sky, people	0.99
sky, buildings, skyline, water, birds, stick, tree, baby, ground, nest	0.40
nest, birds, branch, tree, grass, baby, water, sky, ground, rocks	**1.13**
grass, sky, tree, water, birds, rocks, cubs, grizzly, baby, rodent	0.29
buildings, skyline, sky, water, people, tree, grass, rocks, bear, clouds	0.63

annotation with maximum rate is selected as an optimum solution and assigned to image.

Table 3.6 contains candidate annotations and their ratings for Figure 3.4. According to this table, third candidate annotation, which is generated based on *nest* is optimum for the input image. Combining raw annotations result with relation between words is guide us to find a refined annotation.

Top five words of this candidate annotation is assigned to input image. Figure 3.9 shows the result for input image. Proposed refinement method, removes noisy words like *sky* and *building* while adding related word *branch*.

Algorithm 2 Calculating weight of clique C.

Require: For image I, a raw annotation A^{raw}.

$A^{raw} = \{p_1, \ldots, p_j, \ldots, p_{VN}\}$

p_j : Possibility value of word w_j in raw annotation

$C = \{w_{c_1} \ldots w_{c_n}\}$

1: $Weight_C = 0$

2: $i = 1$ {//}

3: **for** $j = 2 \rightarrow n$ **do**

4: Weight of vertex $w_{c_j} = \mu_{w_{c_i}}(w_{c_j}) * p_{c_j}$

5: $Weight_C = Weight_C$+ Weight of vertex w_{c_j}

6: **end for**

7: **return** $Weight_C$

birds, branch, nest
HANOLISTIC : bird, sky, nest, grass, building
Refined : nest, birds, branch, tree, grass

Figure 3.9: Refinement result of the example run

3.5 Chapter Summary

In this chapter we introduce a new refinement method to improve the output of an automatic image annotation system. Proposed refinement method employs a fuzzy framework to extract relationship between the words in the vocabulary. Fuzzy framework application provides an opportunity for calculating and interpreting relationship between word for statistically unbalanced image datasets. Statistically unbalanced means that distribution of words in image dataset does not have the same statistical properties. More specifically, some of the words occur more frequently than others. In this chapter, we also investigate the probabilistic framework to calculate relationship between words. However, fuzzy framework is more suitable than the probabilistic framework for statistically unbalanced image datasets.

Also, we introduce new process to generate some candidate annotation depending on a given word. Proposed refinement method generates a candidate annotation for each word in annotation result of the automatic image annotation system. At the end of the candidate annotation generation, there are as many candidate annotations as the number of words in annotation result. This process provides us the set of alternatives among to strongly related word to search for local optimal in the annotation result.

CHAPTER 4

EMPIRICAL ANALYSIS OF PROPOSED ANNOTATION REFINEMENT METHOD

In this chapter, we present an experimental analysis of the proposed annotation refinement method. First, we describe experimental setup in terms of dataset, automatic image annotator and proposed refinement method parameters. Then, we investigate the strength and weakness of the refinement method by measuring the performance and compare annotation result of automatic image annotator and proposed refinement method.

4.1 Experiment Setup

4.1.1 Data Set

In order to compare the performance of the suggested annotation refinement method to the automatic image annotator $Annotator^{raw}$ systems, a subset of Corel Draw Photo Collection (Corel 5000) is used in the experiments. This dataset was also used in [8, 9, 10, 7, 13, 14, 16, 15]. In the dataset, there are 5000 images each of which is annotated by 374 distinct words. Each image is annotated with at most five words. Sample images of Corel dataset is provided in Figure 4.1.

The dataset is partitioned into two subsets, one is used as training set and contains 4500 images and the other is used as test set and contains 500 images. Images in test set are annotated with 263 of 374 words and 260 of 263 is also used in training set. Some words are used more frequently than others. As presented at Table 4.1, in the training set 25 high frequent words are occurs in % 50 of training set and 238 low frequent words are occurs in %

(a) people, sand, sky, water (b) dunes, hills, sand, sky (c) elephant, sky, tusks, water

(d) buildings, city, scotland, street (e) bear, polar, snow, tundra (f) buildings, city, sky

(g) kauai, people (h) jet, plane, sky, smoke (i) cars, formula, tracks, turn

Figure 4.1: Sample images and their manual annotations from Corel 5000 dataset.

Table 4.1: Frequency distribution of words in training set.

Class	# of Words	# of Occurrence in Training Set	Occurrence Percentage
High Frequent Words	25	7965	% 50
Low Frequent Words	238	7258	% 45
Words not in Test Set	111	624	% 5
Total	374	15847	% 100

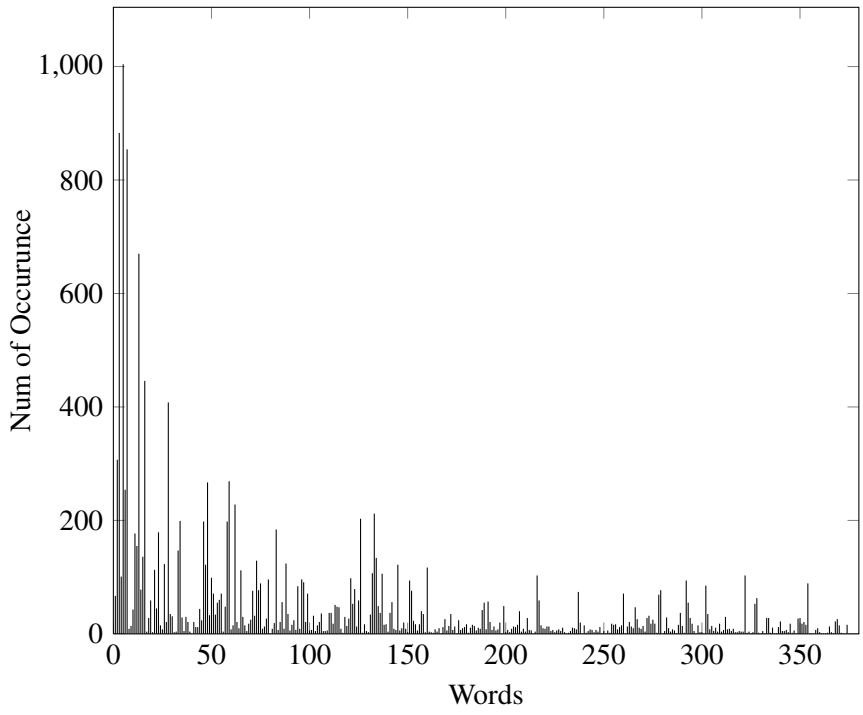


Figure 4.2: Distribution of words in training set.

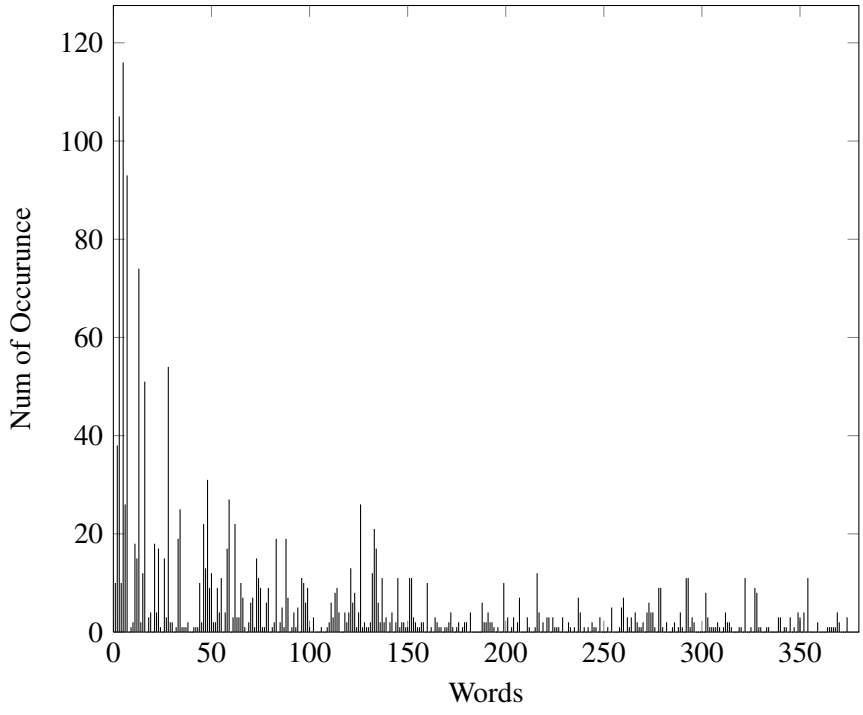


Figure 4.3: Distribution of words in test set.

45 of training set. Remaining 111 words are occurs in % 5 of training set and none of them occurs in test set. Word frequency histograms for training and test set are given in Figure 4.2 and Figure 4.3.

4.1.2 Automatic Image Annotator

HANOLISTIC is selected as a raw annotator ($Annotator^{raw}$) for experiments. The major reason for this choice is, as reported in [20], performance of HANOLISTIC is relatively better than the reported state of art annotator for Corel 5000 dataset.

To make a fair comparison between annotator and the improvement imposed by the suggested refiner, HANOLISTIC is implemented as described in [20]. For each image in dataset, five MPEG-7 descriptors, namely color layout, color structure, scalable color, homogeneous texture and edge histogram, are extracted and used as low level visual descriptor. For each descriptor, a fuzzy k-nearest neighborhood method is employed at *Level-0*. k parameters of each annotator are determined as 9, 2, 5, 17 and 15 respectively using one leave out cross validation. At *Meta-Level*, summation annotator is chosen.

4.1.3 Performance Measurement

In the scope of this thesis, we use four performance measures, namely precision, recall, f-score and non-zero recall, which are defined in Section 2.3. For each experiment, results are measured by using these four performance measures and illustrated with graphics, are generated according to following notation:

- X axis of a graphic (*Top n Words* in Figure 4.4) represents the number of words assigned to an image. As described in section 2.1, for an image, an automatic image annotator calculates possibility values of existence for all words in the vocabulary. At most WN words with the highest possibility values of existence are assigned to the image. Performance of an automatic image annotator system at WN shows the ranking performance of the annotator. In other words, if an automatic image annotation system assigns high possibility value of existence to words, which are exist in the image, performance of this annotation system would be high at lower WN values. To show ranking perfor-

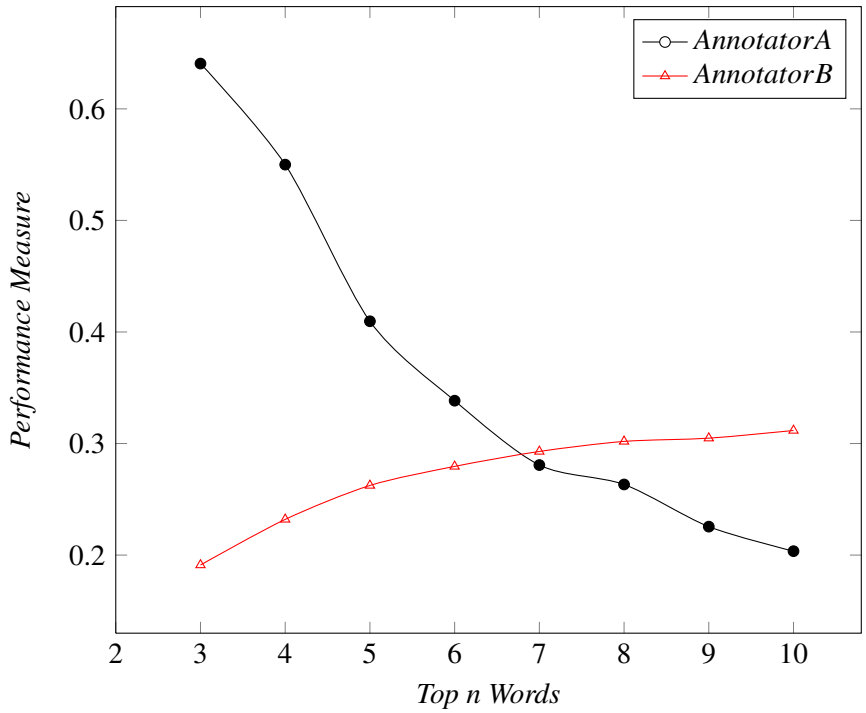


Figure 4.4: A sample performance measurement graphic.

mance of an annotator according to *WN* words, experiments are repeated with different *WN* values.

- Y axis of a graphic (*Performance Measure* in Figure 4.4) represents measurements corresponding *WN* words, calculated according to the selected *Performance Measure*.

4.1.4 Comparison of Fuzzy Framework and Probabilistic Framework For Representing Semantic Similarities Between Words

In Section 3.2, we suggest two approaches to represent semantic similarities between two words. In this section, we demonstrate the result of fuzzy and probabilistic framework on calculating semantic similarities between words. For this purpose, we conducted four sets of experiments.

In the first set of experiments, we compute the semantic similarities between two words, w_i and w_j by approximating the joint probability density functions using the following relative

frequency;

$$r_{i,j} = P(w_i, w_j) \cong \frac{num(w_i, w_j)}{\sum_{w_k \in V} num(w_k)} \ . \tag{4.1}$$

In the second set of experiment, we use the conditional probability distribution to calculate semantic similarities between two words as follows:

$$r_{i,j} = P(w_j|w_i) \cong \frac{num(w_i, w_j)}{num(w_i)} \ . \tag{4.2}$$

In the third set of experiment, following alternative joint probability approximation is used to computed semantic similarities between words:

$$r_{i,j} = P'(w_i, w_j) \cong \frac{num(w_i, w_j)}{(num(w_i) + num(w_j)) - num(w_i, w_j)} \ . \tag{4.3}$$

In the fourth set of experiment, fuzzy membership values are computed to represent semantic similarities between words. Membership values are calculated using the following equation;

$$r_{i,j} = \mu_{w_i}(w_j) = \frac{num(w_i, w_j)}{\min(num(w_i), num(w_j))} \ . \tag{4.4}$$

Figures 4.5, 4.6, 4.7 and 4.8 illustrate performance results of experiments in recall, precision, F-score and non-zero recall measures.

Figure 4.5 represents the recall performance of the refiners which are used for experiments. As clearly seen in this figure, the refiner which employs fuzzy framework provides highest recall values among other refiners. The reason is that, using fuzzy framework to calculate the semantic similarities between words stabilizes distribution in dataset. Coverage of words in vocabulary is dramatically increased as it can be seen from Figures 4.5 and 4.8. However, using joint probability of w_i and w_j for calculating semantic similarities between words, gives the lowest recall values. The reason is that joint probability biases high frequent words. Thus, refiner tends to keep high frequent words in annotation result and removes low frequent words and predicts small number of words in the vocabulary. Conditional probability distribution stabilizes unbalanced word distributions and improves results of joint probability distribution in terms of recall and non-zero call. Due to the insufficient amount of word frequencies, it does not reflect the true joint probability distributions between words. For this reason significant improvement can not be observed in this small dataset. The alternative joint probability approximation gives best recall performance among other probabilistic approaches. However, weakness in representing relationship between words effects refinement performance and could not improve annotation results.

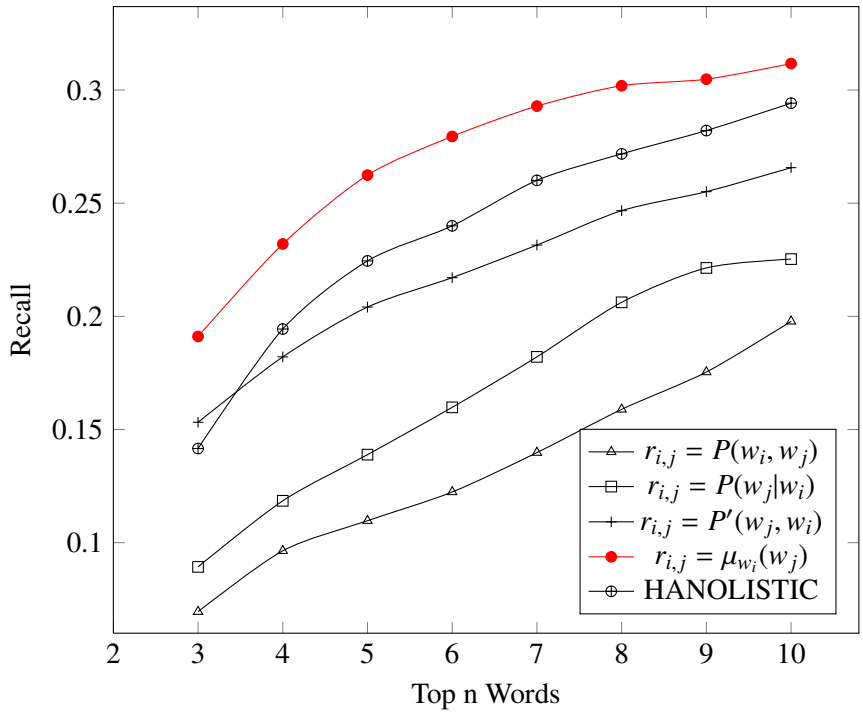


Figure 4.5: Effects of the methods for representing the semantic similarities on the recall values.

As illustrated in Figure 4.6, using joint probability of w_i and w_j for calculating semantic similarities between words, gives the highest precision values. The reason is that joint probability biases high frequent words. The sharp decrease between forth and fifth words in precision values of the refiner which employs joint probability indicates the refiner tends to keep high frequent words in top four words according to possibility value of existence. After fifth word, it could not improve HANOLISTIC precision trend. However, using fuzzy framework provides more steady precision curve than probabilistic framework. This steadiness in precision curve indicates that refiner with fuzzy framework keeps doing refinement on HANOLISTIC annotation results while other refiners follow HANOLISTIC annotation results.

As can be seen in Figure 4.7, even though, the suggested refiner which employs fuzzy frame could not provide highest result in precision values, superiority in recall performance gives highest f-score values for all number of word.

Figure 4.8 represents the non-zero recall values of the suggested refiners. As it can be seen from this figure, the refiner which employs fuzzy framework provides largest coverage among other refiners which employ probabilistic framework.

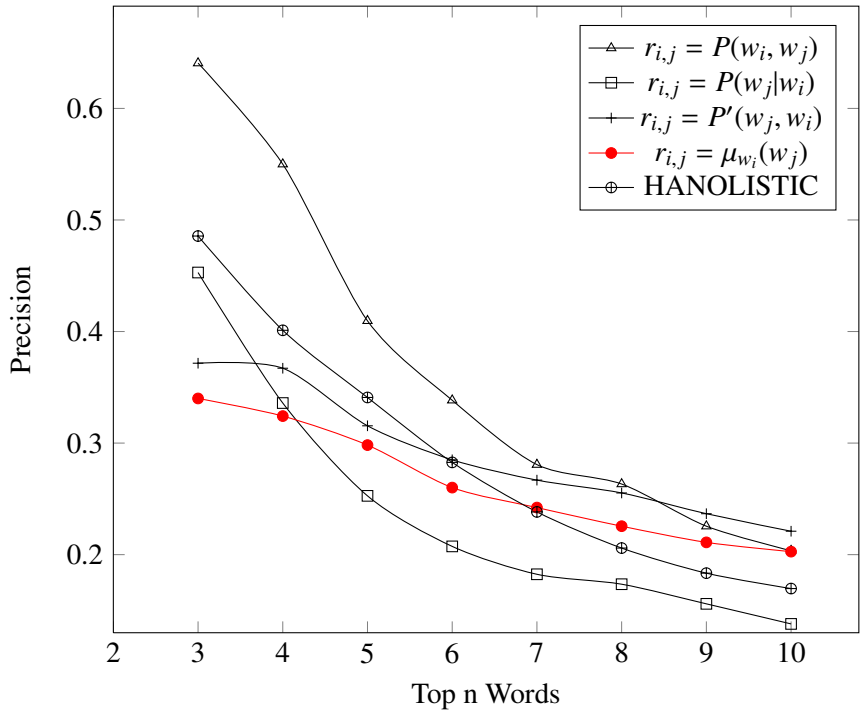


Figure 4.6: Effects of the methods for representing the semantic similarities on the precision values.

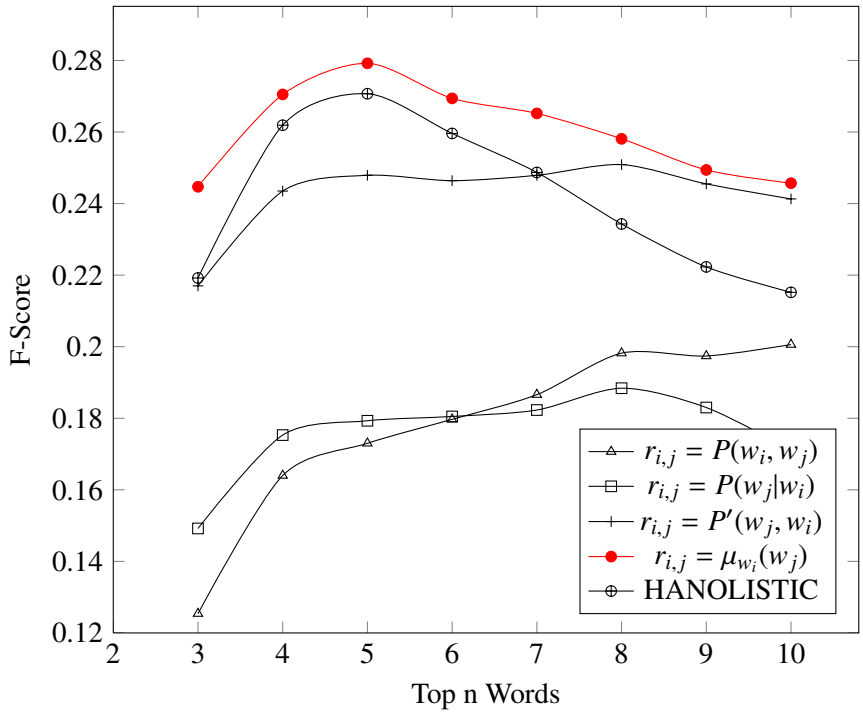


Figure 4.7: Effects of the methods for representing the semantic similarities on the f-score values.

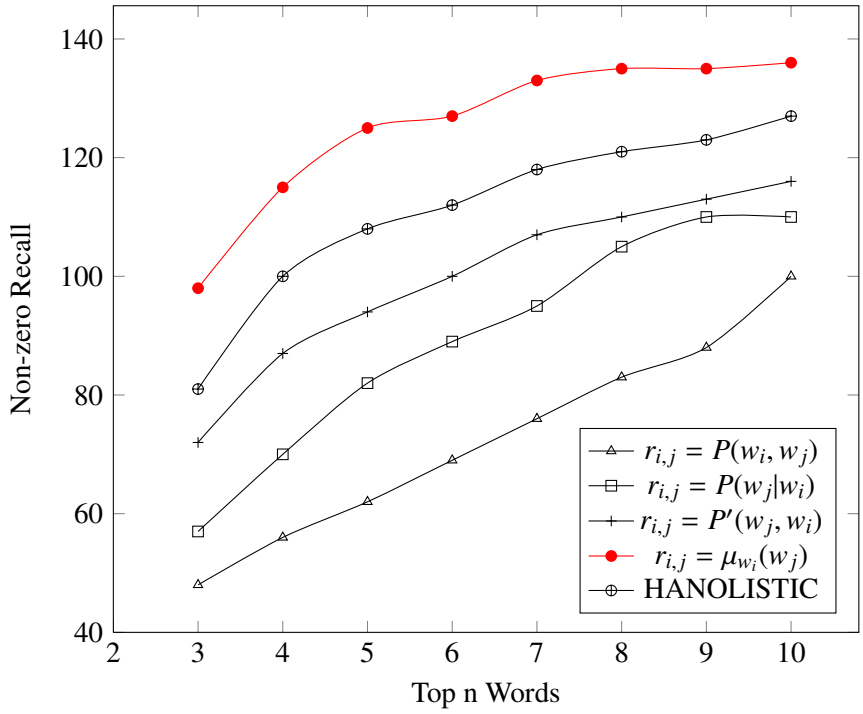


Figure 4.8: Effects of the methods for representing the semantic similarities on the non-zero recall values.

4.1.5 Estimating The System Parameters of The Refinement Method

As described in Section 3.4, in order to find an optimum annotation among candidate annotations, a ranking procedure is needed for each candidate annotation. The candidate annotation with highest rank is then selected as an optimum annotation. Weight of the maximum weighted clique in candidate annotation is assigned to that candidate annotation as a rank. Maximum weighted clique problem is known to be an NP-complete problem [21]. In the scope of this study, we use a brute force method for solving this problem instead of proposing an optimal solution. Execution time of proposed refinement method is bounded by the method which is used for finding maximum weighted clique. From the computational point of view, we define two parameters to optimize execution time performance.

First parameter is the number of vertices in a clique. Because of the nature of candidate annotation generation method, candidate annotation is represented by a fully connected graph. Therefore, the biggest clique is the graph itself. The problem is then reduced to process to find maximum weighted clique in this graph all cliques should be processed. In the worst case situation a candidate annotation may contain all words in the vocabulary. For example, in the Corel dataset, there is 374 distinct words in the vocabulary and brute force method generates $2^{374} - 1$ cliques. The execution time is reduced by defining the maximum number of vertex in a clique. In Corel dataset, images are annotated with at most five words, in average 3.5 words are assigned per image. Intuitively, one of the generated cliques with 3 - 5 vertices is expected to contain all of the related words we seek. In order to select the optimum number of vertices an experiment is applied. At each epoch, we only change number of vertices in a clique. Results clearly indicate that maximum number of vertices in a clique in does not effect result obviously, but execution time performance is highly dependent on number of vertices in a clique . Results of this experiment is reported at Figures 4.9, 4.10, 4.11, 4.12. As we expected, according to this experiment, four vertices is enough to generate a clique.

Second parameter is the number of generated candidate annotations. It is also possible to generate a candidate annotation for each word in vocabulary, but it would consume more time and would not effect the annotation performance positively. The reason is that, proposed system combines raw annotation and relationship between the words, and only top n words with highest possibility values in raw annotation change the annotation result. In order to find the optimum value of n, we conduct an experiment for measuring the performance of different

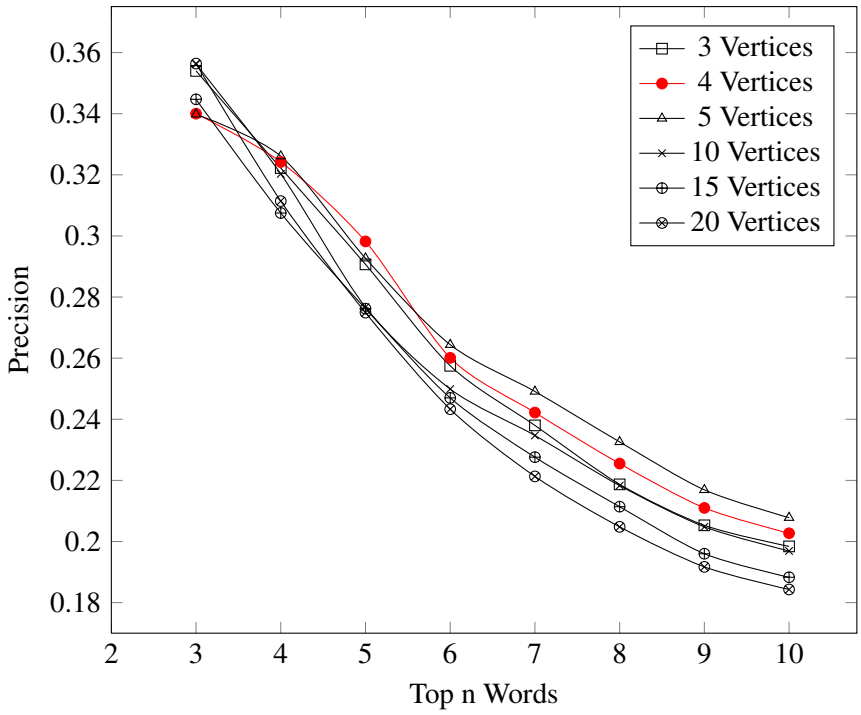


Figure 4.9: Effects of different number clique vertices on the precision values.

n values. Results of this experiment is reported at Figures 4.13, 4.14, 4.15, 4.16. According to this experiment, generating 5 candidate annotations for top 5 words with highest possibility values builds a balance between execution time and annotation performance.

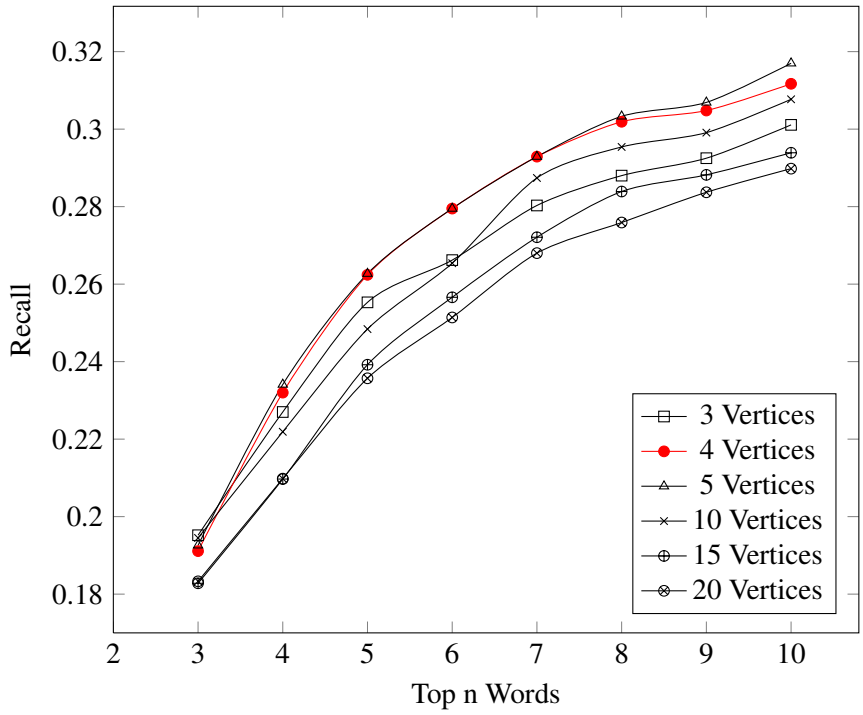


Figure 4.10: Effects of different number clique vertices on the recall values.

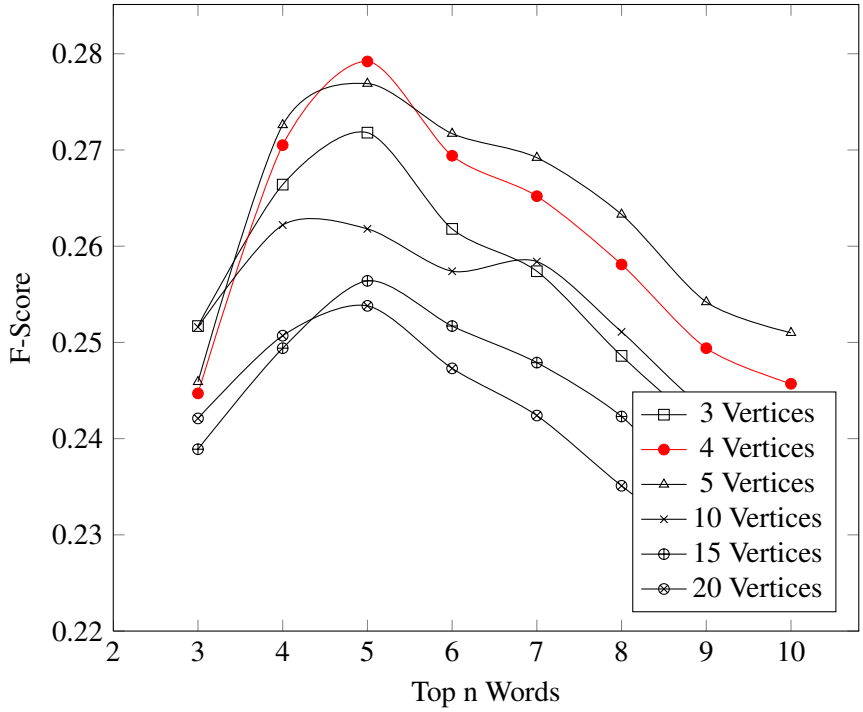


Figure 4.11: Effects of different number clique vertices on the f-score values.

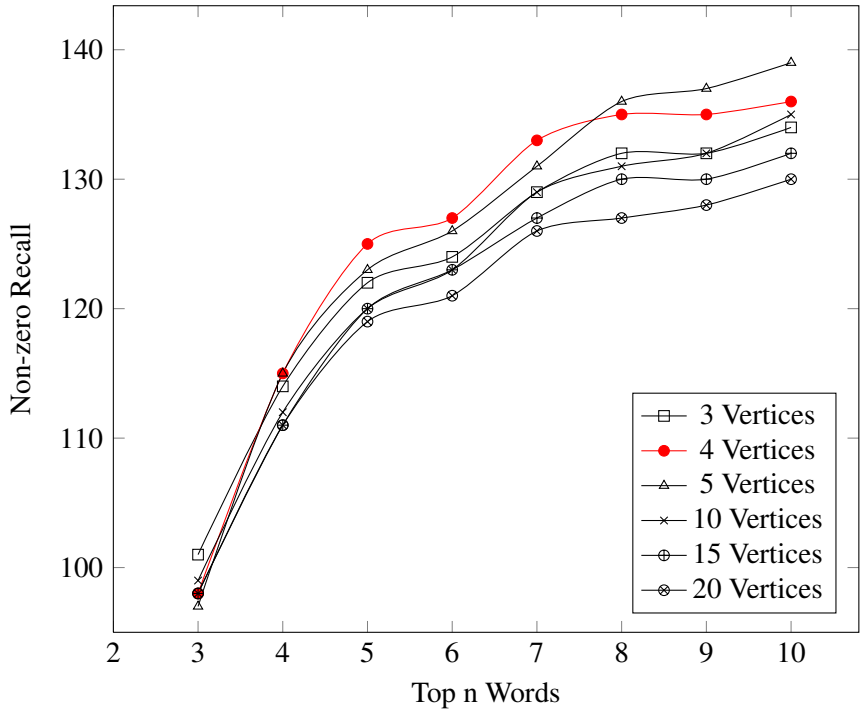


Figure 4.12: Effects of different number clique vertices on the non-zero recall values.

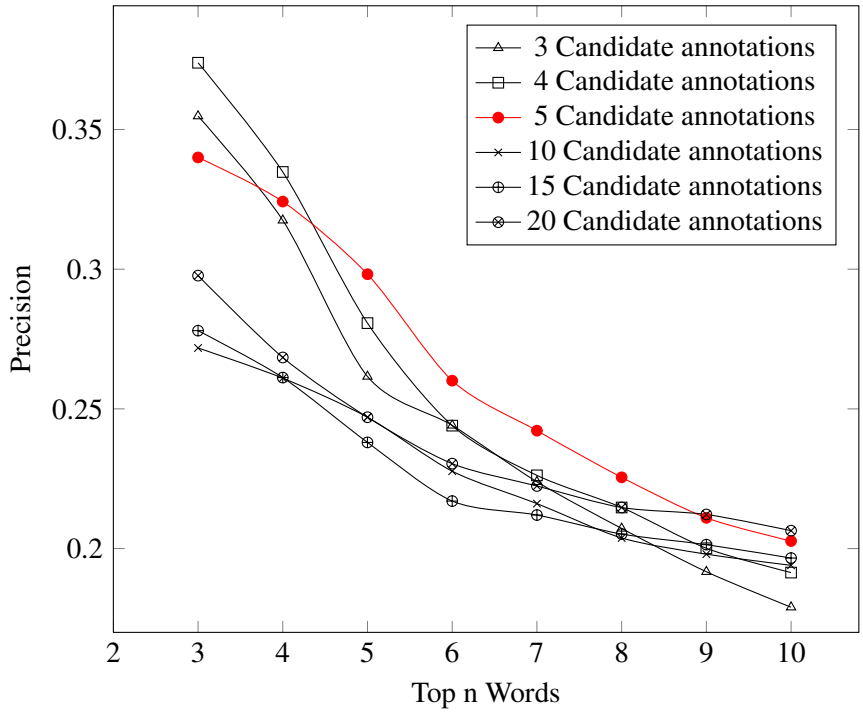


Figure 4.13: Effects of generating different number of candidate annotations on the precision values.

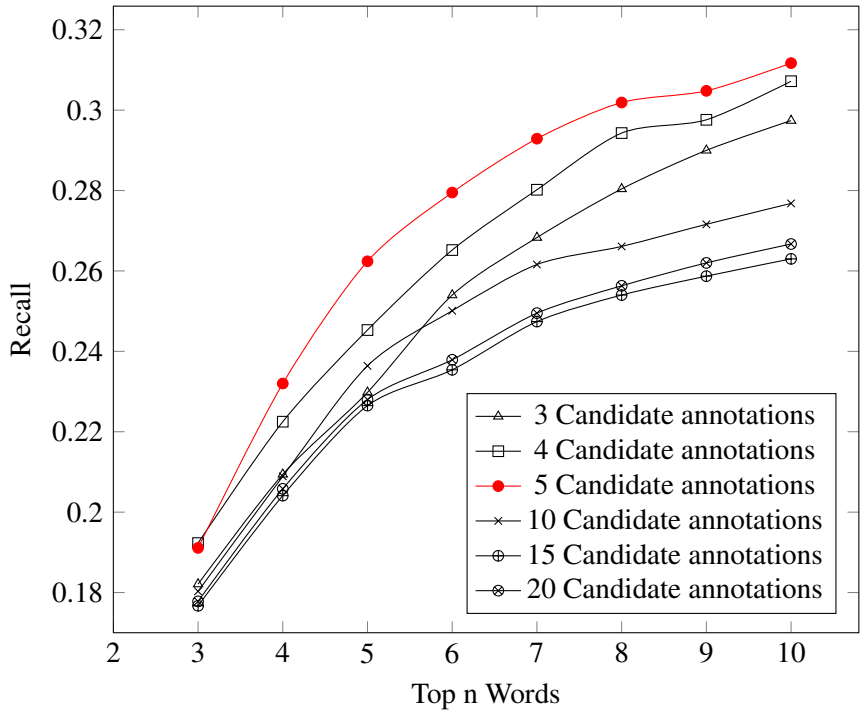


Figure 4.14: Effects of generating different number of candidate annotations on the recall values.

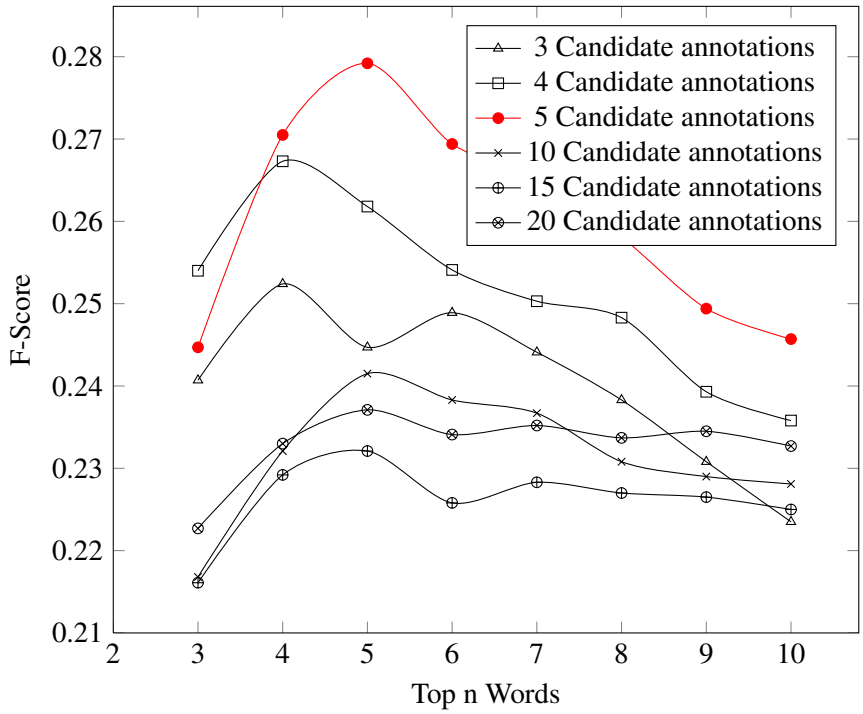


Figure 4.15: Effects of generating different number of candidate annotations on the f-score values.

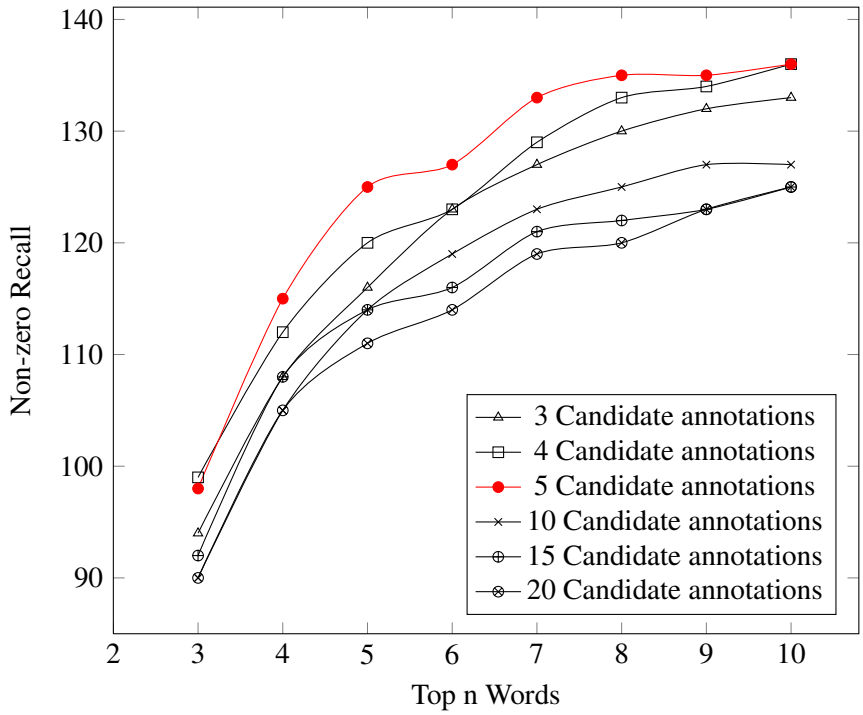


Figure 4.16: Effects of generating different number of candidate annotations on the non-zero recall values.

4.2 Results

Performance of the proposed refinement method is compared with HANOLISTIC to measure the improvement of automatic annotations. Figures 4.17, 4.18, 4.19 and 4.20 illustrate results of proposed refinement method and HANOLISTIC.

As it can be seen in the Figure 4.17, HANOLISTIC provides higher precision values than proposed refinement method at 3 to 5 words. However, after 5 words, HANOLISTIC's precision values decrease sharply compared to the proposed refinement method's precision values. The reason of this behavior is that HANOLISTIC assigns high possibility value of existence to high frequent words and these words groups at the top of the annotation result of HANOLISTIC. This behavior also can be observed in Figures 4.18 and 4.20, where the vocabulary coverage of HANOLISTIC is limited and sharp increase after third word supports the HANOLISTIC behavior. Proposed refinement method provides more steady precision decrease than HANOLISTIC. This means that, proposed method spreads improvements over annotation result.

In Figure 4.18, proposed refinement method provides higher recall values than HANOLISTIC for all number of words. This recall statistic shows that refinement method improves vocabulary coverage of HANOLISTIC. This behavior also supported by non-zero recall result (Figure 4.20).

Figure 4.19 shows the improvement of HANOLISTIC in terms of F-score. As described in Section 2.3, F-score provides single performance value for the system by calculating the harmonic mean of precision and recall and annotation performance of system could be compared by using this value. As it can bee seen clearly in Figure 4.19, proposed refinement method not only improves HANOLISTIC F-score values but also provides more steady decrease after fifth word.

The main reason of improvement in recall and non-zero recall is that HANOLISTIC employs fuzzy k-nearest neighbor algorithm at *Level-0*. Therefore HANOLISTIC is sensitive to occurrence number of words. For a high frequent word in train set, HANOLISTIC tends to assign this word to all of the images. However, less frequent words could not get high possibility value of existence. Proposed refinement method reevaluates possibility values by using semantic similarities words and gives high possibility value of existence to less frequent words

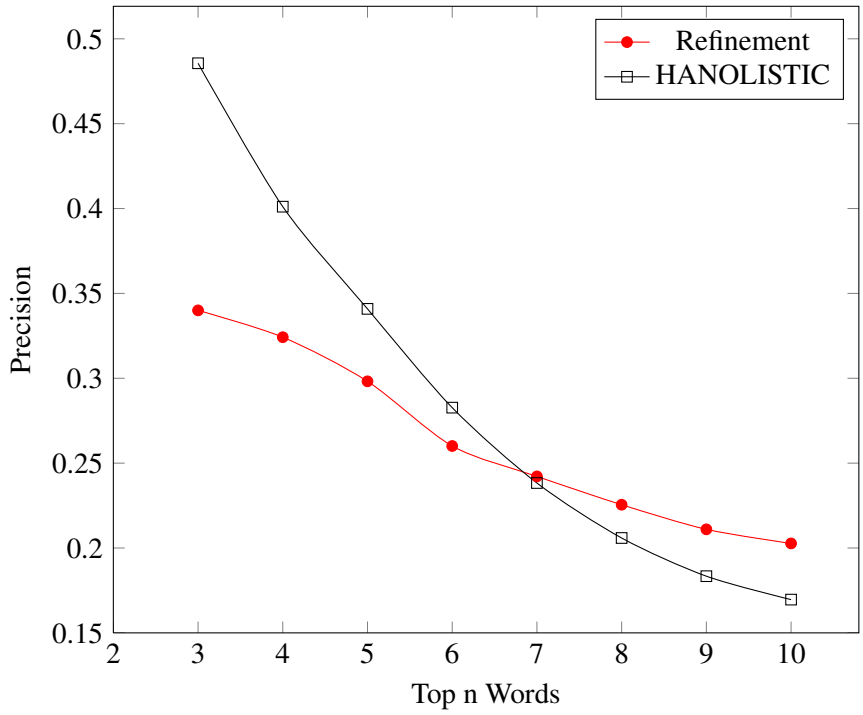


Figure 4.17: Precision values of HANOLISTIC and proposed refinement method.

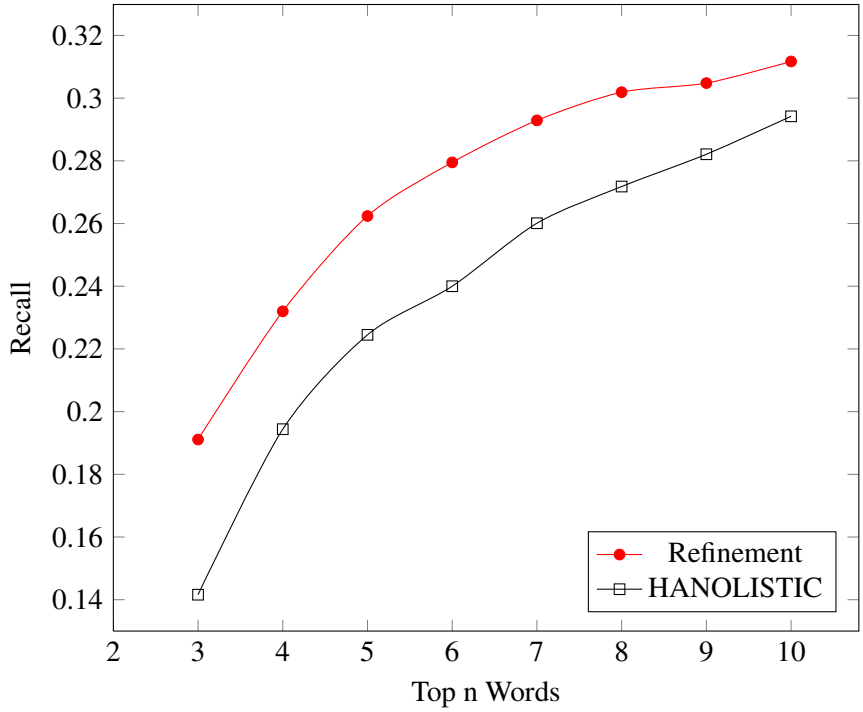


Figure 4.18: Recall values of HANOLISTIC and proposed refinement method.

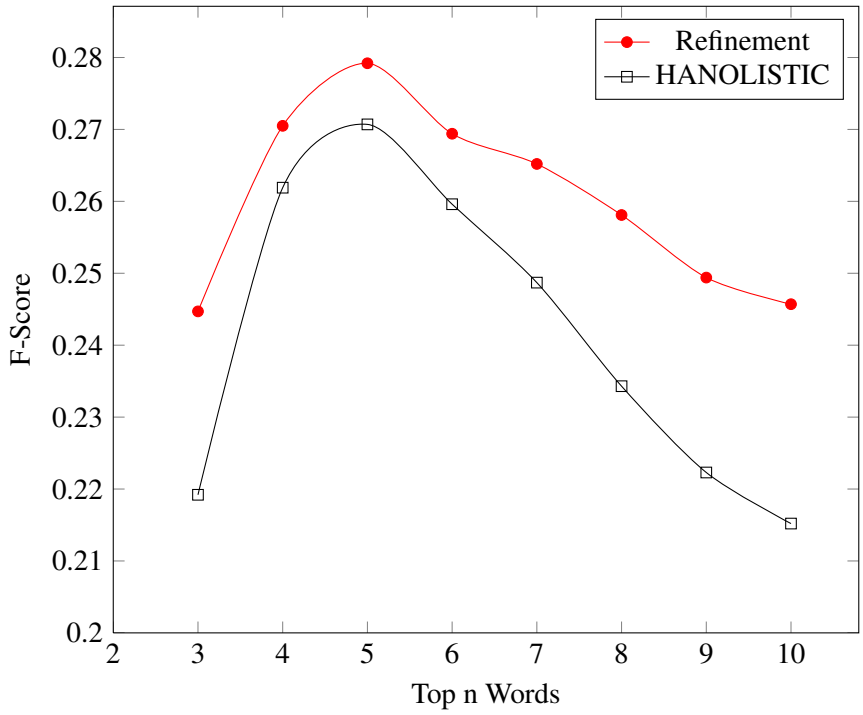


Figure 4.19: F-Score values of HANOLISTIC and proposed refinement method.

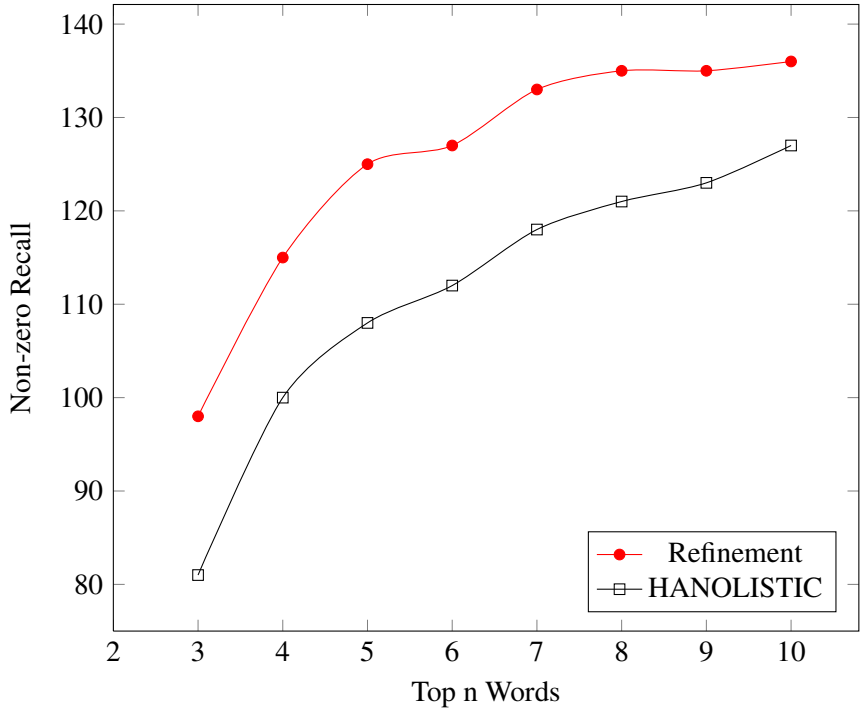


Figure 4.20: Non-zero recall values of HANOLISTIC and proposed refinement method.

Table 4.2: Vocabulary coverage of HANOLISTIC and Proposed Refinement Method

	HANOLISTIC		Refinement	
	# of Predict	Coverage	# of Predict	Coverage
High Frequent 25 words	25	% 100	25	% 100
Low Frequent 238 words	83	% 29	100	% 35

to be annotated. For example, *cactus* in Figure 4.21.c, *branch* in Figure 4.21.d and *meadow* in Figure 4.21.e are not annotated by HANOLISTIC because the frequency of occurrence these words is less than other words in the vocabulary. However, the proposed refinement method extracts these words by establishing the semantic relationship of the less frequent words to the other words and correctly assigns them in to the images.

As described in Table 4.1, 25 high frequent words occur in % 50 of training set and remaining 238 words which also occur in test set cover % 45 of training set. In Tables 4.2, 4.3, 4.4, these word groups are labeled as ”High Frequent 25 words” and ”Low frequent 238 words” respectively. Table 4.2 compares the coverage result of HANOLISTIC and the suggested refinement method. HANOLISTIC and proposed refinement method cover all high frequent 25 words, but HANOLISTIC covers only 29 percent of less frequent 238 words, while proposed refinement method cover 35 percent of these words.

Some of the precision values of the suggested refinement method is lower than the raw annotation results of HANOLISTIC. The major reason of the decrease in precision values is that proposed refinement method is highly dependent on the performance of the raw annotator. If raw annotator proposes too many unrelated, noisy words in annotation, proposed method finds an optimal solution for these noisy words. For example, in Figure 4.22.c, HANOLISTIC proposes ”people”, ”water”, ”street”, ”tables” and ”scotland”. However the image is manually annotated with ”building”, ”palace” and ”people”. ”people” is the correct word in annotation result of HANOLISTIC. According to annotation of HANOLISTIC, proposed refinement method constructs relationship between ”people”, ”tables” and ”street” and removes ”water” and ”scotland” from annotation. Instead of ”water” and ”scotland”, proposed refinement method adds ”restaurant” and ”food” to annotation. The reason is that, ”people”, ”table” and ”street” words have higher degree of relationship with ”food” and ”restaurant” than ”water” and ”scotland”. However, the word *restaurant* does not occur in test set. This kind of false positive word predictions decrease the precision result of proposed refinement method.

Table 4.3: Precision comparison of HANOLISTIC and Proposed Refinement Method

	HANOLISTIC		Refinement	
	# of Predict	Precision	# of Predict	Precision
High Frequent 25 words	25	0.34	25	0.38
Low Frequent 238 words	149	0.34	173	0.28

Table 4.4: Number of false prediction of HANOLISTIC and Proposed Refinement Method

	HANOLISTIC		Refinement	
	# of Predict	# of False Predict	# of Predict	# of False Predict
High Frequent 25 words	25	0	25	0
Low Frequent 238 words	149	66	173	73

Table 4.3 shows that proposed refinement method has higher precision values for more frequent words. However, for low frequent words, the precision value is decreased. The main reason of this decrease is the increase of false positive word predictions. As Table 4.4 represents, because of proposed system gives higher rank to low frequent words, in case of wrong raw annotation, system tends to assign less frequent words to images and these false predictions decrease precision value.

beach, kauai, people, water
Raw : water, field, beach, tulip, people
Refined : beach, water, sand, people, tree
(a)

boats, people, water
Raw : water, people, buildings, rocks, sky
Refined : water, boats, people, buildings, crab
(b)

blooms, cactus, flowers, needles
Raw : flowers, grass, petals, needles, blooms
Refined : blooms, flowers, cactus, needles
(c)

birds, branch, nest
Raw : birds, sky, nest, grass, buildings
Refined : nest, birds, branch, tree, grass
(d)

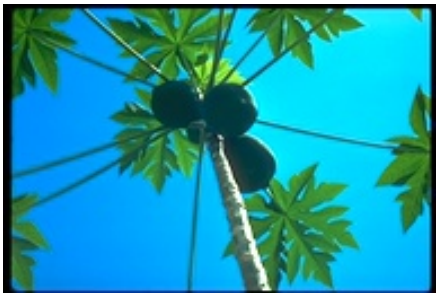
bear, grass, grizzly, meadow
Raw : bear, grass, tree, horses, grizzly
Refined : grizzly, bear, meadow, grass, water
(e)

jet, plane, sky
Raw : plane, jet, sky, clouds, bear
Refined : jet, plane, sky, clouds
(f)

oahu, people, waves
Raw : water, people, forest, waves, sky
Refined : waves, water, people, oahu, sky
(g)

close-up, leaf, plants
Raw : leaf, plants, grass, tree, birds
Refined : leaf, plants, close-up, flowers, stems
(h)

Figure 4.21: Positive refinement samples



branch, leaf, sky
Raw : people, water, sky, pool, swimmers
Refined : swimmers, people, pool, water
(a)

frost, ice, plants
Raw : water, sky, rocks, people, tree
Refined : water, sand, sky, dunes, rodent
(b)

buildings, palace, people
Raw : people, water, street, tables, scotland
Refined : tables, people, restaurant, street, food
(c)

desert, rocks, sand
Raw : tree, water, sky, sun, city
Refined : sun, sky, tree, water, lake
(d)

mountain, rocks
Raw : grass, bear, polar, cubs, wall
Refined : bear, polar, cubs, grass, tree
(e)

giraffe, grass, sky, tree
Raw : sky, water, sand, tree, plane
Refined : plane, jet, sky, runway, clouds
(f)

bear, grass, polar, snow
Raw : water, field, sky, grass, scotland
Refined : scotland, water, cottage, mountain
(g)

albatross, flight, sky
Raw : sky, flight, birds, plane, jet
Refined : jet, plane, sky
(h)

Figure 4.22: Negative refinement samples

CHAPTER 5

CONCLUSION AND FUTURE DIRECTIONS

Annotating an image with content related words is a very useful and challenging task. It is still an unsolved problem and an active research topic in computer science. However, performance of available automatic image annotation techniques are far behind for the practical usage and newly proposed method could not make further improvement on annotation performance. The major reason is the image annotation methods map low level visual features to high level semantic words. Since low level visual features do not represent any semantic meaning, performance of annotation system are limited. Image annotation refinement methods aim to improve annotation performance by using additional semantic similarity information between words. Semantic similarities between words could be calculated by using the external word databases or extracted from the image dataset itself.

In this study we propose a novel image annotation refinement method. We introduce two new approaches for image annotation refinement problem. First approach is about representing semantic similarities between words. We extract semantic similarities between words from the image dataset. We assume that if two words occur in the same image, there is a relation between these two words. Frequency of co-occurrence is used to calculate degree of relationship between words. For example, let us assume that we have an image dataset which contains outdoor images. In this dataset, word ”mountain” occurs in every image which contains word ”lake”. By using this information, if an newly seen image contains word ”lake”, we infer that the image also contains word ”mountain”. As it can be seen from the example, degree of relationship between words is highly dependent to distribution of words in dataset. For example, if only one image contains word ”lake” and word ”boat”, relationship between ”lake” and ”boat” could not be inferred. In our approach, we introduce a fuzzy framework to represent relationship between words. By using fuzzy framework, we get ability to represent

relationship between words without effecting word distribution in the dataset. If we go back the previous example, we infer strong relation between ”lake” and ”boat”, because of every image contains ”boat” also contains ”lake”.

Second approach is about refining raw annotation result of an automatic image annotator. We assume that every word in raw annotation result could be correct annotation for the input image. Thus, we introduce a candidate annotation generation process for a given word. For a given word, we construct a relational graph and by processing this relational graph we generate a candidate solution. By using this process we generate as many candidate annotations as number of words in raw annotation. All generated candidate annotations represent a local optimal solution for the given word. By using these candidate annotations we can search solution space and find optimal refinement solution.

Experiments in Section 4 show that proposed refinement method improves raw annotator performance. However, experiments also indicate two weakness of proposed method. First weakness is that candidate annotation generation and selection process are expensive operations and bring extra complexity to annotation process. In the scope of this thesis, we do not implement optimized solution for generation and selection process. Instead of this, we restrict parameters of generation and selection process. Experiment show that these restrictions do not affect on performance but for large datasets runtime complexity become a main problem of proposed refinement system.

Second weakness of proposed refinement system is that performance of proposed refinement system is highly dependent to the raw annotator performance. If the raw annotator annotates an image with too many noisy words, proposed refinement system provides a local optimal solution for this raw annotation result and could not refine raw annotation result. In the scope of this thesis, we focused on raw annotation result of automatic image annotator and do not interfere annotation process. To improve refinement process, annotation process of raw annotator and refinement process of proposed refinement method could be combined.

5.1 Future Directions

Proposed refinement method could be improved as follows:

- Proposed refinement method uses annotation results of an annotation system. This provides abstraction between annotation and refinement systems and makes easier to design a refinement system. However, getting some low level information from image annotation system, like position or color of object, could improve the refinement results.

- When calculating relationship between words, co-occurrence statistics are used and occurrence of a word is connected to other word occurrence. For example, we connect *bird* and *fly* and if we find *bird* we assume that there could be *fly*. Occurrence of the third or forth word may be used to find the degree of relationship among more than two words. For example, may be *bird* and *fly* connected by *sky* and if we observe *bird* and *sky* we could ask *fly* into the annotation result.

REFERENCES

[1] C. Fellbaum, *WordNet: An Electronic Lexical Database*. The MIT Press, May 1998.

[2] R. Zhang, Z. M. Zhang, M. Li, W. ying Ma, and H.-J. Zhang, "A probabilistic semantic model for image annotation and multi-modal image retrieval," *Multimedia Systems*, vol. 12, pp. 27–33, 2006.

[3] T. Hofmann, "Probabilistic latent semantic analysis," in *In Proc. of Uncertainty in Artificial Intelligence, UAI'99*, 1999, pp. 289–296.

[4] C. Cusano, G. Ciocca, and R. Schettini, "Image annotation using SVM," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 5304, 2003, pp. 330–338.

[5] E. Akbas and F. Yarman Vural, "Automatic image annotation by ensemble of visual descriptors," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, june 2007, pp. 1 –8.

[6] D. Putthividhya, H. Attias, and S. Nagarajan, "Supervised topic model for automatic image annotation," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, march 2010, pp. 1894 –1897.

[7] O. Karadag and F. Vural, "Hanolistic: A hierarchical automatic image annotation system using holistic approach," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, 2009, pp. 16 –21.

[8] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, ser. SIGIR '03. New York, NY, USA: ACM, 2003, pp. 119–126.

[9] V. Lavrenko, R. Manmatha, and J. Jeon, "A model for learning the semantics of pictures," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[10] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple bernoulli relevance models for image and video annotation," in *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, ser. CVPR'04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 1002–1009.

[11] F. Monay and D. Gatica-Perez, "On image auto-annotation with latent space models," in *Proceedings of the eleventh ACM international conference on Multimedia*, ser. MULTIMEDIA '03. New York, NY, USA: ACM, 2003, pp. 275–278.

[12] ——, "Modeling semantic aspects for cross-media image indexing," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 10, pp. 1802 –1817, 2007.

[13] ——, "Plsa-based image auto-annotation: constraining the latent space," in *Proceedings of the 12th annual ACM international conference on Multimedia*, ser. MULTIMEDIA '04. New York, NY, USA: ACM, 2004, pp. 348–351.

[14] Y. Jin, L. Khan, L. Wang, and M. Awad, "Image annotations by combining multiple evidence & wordnet," in *Proceedings of the 13th annual ACM international conference on Multimedia*, ser. MULTIMEDIA '05. New York, NY, USA: ACM, 2005, pp. 706–715.

[15] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang, "Image annotation refinement using random walk with restarts," in *Proceedings of the 14th annual ACM international conference on Multimedia*, ser. MULTIMEDIA '06. New York, NY, USA: ACM, 2006, pp. 647–650.

[16] Y. Wang and S. Gong, "Refining image annotation using contextual relations between words," in *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*. New York, NY, USA: ACM, 2007, pp. 425–432.

[17] R. Cilibrasi and P. Vitanyi, "The google similarity distance," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 3, pp. 370 –383, march 2007.

[18] K. H. Lee, *First Course On Fuzzy Theory And Applications.*, ser. Advances in Intelligent and Soft Computing. Berlin: Springer-Verlag, 2005, vol. 27.

[19] P. R. J. Östergård, "A new algorithm for the maximum-weight clique problem," *Electronic Notes in Discrete Mathematics*, vol. 3, pp. 153 – 156, 1999, 6th Twente Workshop on Graphs and Combinatorial Optimization.

[20] Özge Öztimur, "Hanolistic: A hierarchical automatic image annotation system using holistic approach," Master's thesis, Middle East Technical University, January 2008.

[21] E. Balas, V. Chvátal, and J. Nešetřil, "On the maximum weight clique problem," *Math. Oper. Res.*, vol. 12, pp. 522–535, August 1987.

APPENDIX A

WORD FREQUENCIES IN TRAINING AND TEST SETS

Table A.1: Word frequencies in training and test sets

		Num. of Occurence in	
Index	Word	Training Set	Test Set
1	city	67	10
2	mountain	307	38
3	sky	883	105
4	sun	101	10
5	water	1004	116
6	clouds	254	26
7	tree	854	93
8	bay	9	0
9	lake	14	1
10	sea	43	2
11	beach	177	18
12	boats	155	15
13	people	670	74
14	branch	78	2
15	leaf	136	12
16	grass	446	51
17	plain	4	0
18	palm	28	3
19	horizon	59	4
20	shell	3	0
21	hills	113	18
22	waves	45	4
23	birds	179	17
24	land	15	1
25	dog	8	0

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
26	bridge	123	15
27	ships	21	3
28	buildings	408	54
29	fence	35	2
30	island	31	2
31	storm	3	0
32	peaks	4	1
33	jet	147	19
34	plane	199	25
35	runway	29	1
36	basket	5	1
37	flight	30	1
38	flag	21	2
39	helicopter	4	0
40	boeing	1	0
41	prop	21	1
42	f-16	12	1
43	tails	12	1
44	smoke	44	10
45	formation	24	2
46	bear	198	22
47	polar	122	13
48	snow	267	31
49	tundra	33	9
50	ice	99	12
51	head	71	2
52	black	34	2
53	reflection	55	9
54	ground	60	4
55	forest	71	11
56	fall	4	0
57	river	48	4
58	field	198	17
59	flowers	269	27
60	stream	8	0
61	meadow	15	3
62	rocks	228	22

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
63	hillside	21	3
64	shrubs	10	3
65	close-up	112	10
66	grizzly	30	7
67	cubs	15	1
68	drum	5	0
69	log	18	2
70	hut	25	6
71	sunset	76	7
72	display	32	1
73	plants	129	15
74	pool	77	11
75	coral	89	9
76	fan	9	1
77	anemone	13	1
78	fish	27	6
79	ocean	96	9
80	diver	1	0
81	sunrise	9	1
82	face	19	2
83	sand	184	19
84	rainbow	7	0
85	farms	21	2
86	reefs	56	5
87	vegetation	9	1
88	house	124	19
89	village	35	7
90	carvings	6	0
91	path	16	1
92	wood	24	4
93	dress	7	1
94	coast	84	5
95	sailboats	9	0
96	cat	96	11
97	tiger	91	10
98	bengal	21	6
99	fox	71	9

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
100	kit	6	1
101	run	7	0
102	shadows	32	3
103	winter	5	0
104	autumn	15	0
105	cliff	21	0
106	bush	36	1
107	rockface	5	0
108	pair	5	0
109	den	6	1
110	coyote	37	2
111	light	37	6
112	arctic	18	3
113	shore	51	8
114	town	48	9
115	road	47	4
116	chapel	9	0
117	moon	2	0
118	harbor	30	4
119	windmills	14	2
120	restaurant	27	4
121	wall	98	13
122	skyline	53	6
123	window	79	8
124	clothes	13	1
125	shops	59	4
126	street	203	26
127	cafe	2	1
128	tables	17	2
129	nets	5	1
130	crafts	3	1
131	roofs	34	2
132	ruins	107	12
133	stone	212	21
134	cars	134	17
135	castle	49	6
136	courtyard	37	2

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
137	statue	106	11
138	stairs	16	2
139	costume	17	3
140	sponges	4	0
141	sign	37	2
142	palace	56	4
143	paintings	9	0
144	sheep	7	2
145	valley	122	11
146	balcony	6	1
147	post	10	2
148	gate	20	2
149	plaza	9	1
150	festival	3	1
151	temple	94	11
152	sculpture	76	11
153	museum	23	3
154	hotel	17	2
155	art	6	1
156	fountain	17	1
157	market	40	2
158	door	35	2
159	mural	3	0
160	garden	117	10
161	star	4	0
162	butterfly	3	1
163	angelfish	2	0
164	lion	8	3
165	cave	4	2
166	crab	10	1
167	grouper	0	1
168	pagoda	12	0
169	buddha	26	1
170	decoration	6	1
171	monastery	14	2
172	landscape	35	4
173	detail	7	1

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
174	writing	13	0
175	sails	1	1
176	food	24	2
177	room	7	0
178	entrance	10	1
179	fruit	12	2
180	night	17	2
181	perch	2	0
182	cow	11	4
183	figures	16	0
184	facade	14	0
185	chairs	6	0
186	guard	11	0
187	pond	9	0
188	church	42	6
189	park	55	2
190	barn	8	2
191	arch	57	4
192	hats	21	2
193	cathedral	7	2
194	ceremony	13	1
195	crowd	6	0
196	glass	8	1
197	shrine	20	0
198	model	1	0
199	pillar	49	10
200	carpet	3	0
201	monument	7	3
202	floor	3	0
203	vines	9	1
204	cottage	13	3
205	poppies	12	0
206	lawn	16	2
207	tower	40	7
208	vegetables	4	0
209	bench	9	0
210	rose	4	0

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
211	tulip	28	3
212	canal	7	1
213	cheese	6	0
214	railing	1	0
215	dock	4	1
216	horses	103	12
217	petals	59	4
218	umbrella	15	0
219	column	10	2
220	waterfalls	9	0
221	elephant	13	3
222	monks	13	3
223	pattern	5	0
224	interior	7	3
225	vendor	3	1
226	silhouette	6	1
227	architecture	8	1
228	blossoms	5	0
229	athlete	11	3
230	parade	3	0
231	ladder	1	0
232	sidewalk	2	2
233	store	5	1
234	steps	11	0
235	relief	10	1
236	fog	8	0
237	frost	74	7
238	frozen	20	4
239	rapids	1	0
240	crystals	15	1
241	spider	2	0
242	needles	5	1
243	stick	8	0
244	mist	7	2
245	doorway	3	1
246	vineyard	7	1
247	pottery	4	0

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
248	pots	12	3
249	military	1	0
250	designs	5	0
251	mushrooms	1	0
252	terrace	6	1
253	tent	2	0
254	bulls	18	5
255	giant	16	0
256	tortoise	17	0
257	wings	8	0
258	albatross	12	1
259	booby	15	5
260	nest	71	7
261	hawk	1	0
262	iguana	13	3
263	lizard	21	1
264	marine	13	3
265	penguin	10	0
266	deer	47	4
267	white-tailed	26	2
268	horns	11	1
269	slope	9	1
270	mule	14	2
271	fawn	4	0
272	antlers	28	4
273	elk	32	6
274	caribou	18	4
275	herd	25	4
276	moose	18	1
277	clearing	2	0
278	mare	69	9
279	foals	77	9
280	orchid	1	1
281	lily	5	0
282	stems	29	2
283	row	10	0
284	chrysanthemums	4	0

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
285	blooms	8	1
286	cactus	6	2
287	saguaro	1	0
288	giraffe	16	1
289	zebra	37	4
290	tusks	14	1
291	hands	2	0
292	train	94	11
293	desert	55	11
294	dunes	28	1
295	canyon	18	3
296	lighthouse	5	2
297	mast	1	0
298	seals	15	0
299	texture	3	0
300	dust	3	0
301	pepper	1	0
302	swimmers	85	8
303	pyramid	35	3
304	mosque	8	1
305	sphinx	14	1
306	truck	5	1
307	fly	11	1
308	trunk	4	2
309	baby	18	1
310	eagle	4	0
311	lynx	7	1
312	rodent	30	4
313	squirrel	8	2
314	goat	9	2
315	marsh	5	1
316	wolf	9	0
317	pack	3	0
318	dall	3	0
319	porcupine	5	1
320	whales	4	1
321	rabbit	4	0

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
322	tracks	103	11
323	crops	2	0
324	animals	4	0
325	moss	0	1
326	trail	3	0
327	locomotive	53	9
328	railroad	63	8
329	vehicle	2	1
330	aerial	0	1
331	range	5	0
332	insect	1	0
333	man	28	1
334	woman	28	1
335	rice	2	0
336	prayer	11	0
337	glacier	1	0
338	harvest	1	0
339	girl	12	3
340	indian	22	3
341	pole	5	0
342	dance	5	1
343	african	7	1
344	shirt	3	0
345	buddhist	18	3
346	tomb	2	0
347	outside	6	1
348	shade	1	0
349	formula	27	4
350	turn	28	3
351	straightaway	18	0
352	prototype	21	4
353	steel	16	0
354	scotland	89	11
355	ceiling	1	0
356	furniture	1	0
357	lichen	1	0
358	pups	7	0

Continued on Next Page...

Table A.1 – Continued

		Num. of Occurence in	
Index	Word	Training Set	Test Set
359	antelope	10	2
360	pebbles	3	0
361	remains	1	0
362	leopard	1	0
363	jeep	1	0
364	calf	2	1
365	reptile	13	1
366	snake	3	1
367	cougar	1	1
368	oahu	22	1
369	kauai	26	4
370	maui	15	2
371	school	2	0
372	canoe	1	0
373	race	1	0
374	hawaii	16	3