Universidade Nova de Lisboa

Faculdade de Ciências e Tecnologia

Departamento de Informática

Dissertação de Mestrado

Automated Image Tagging through Tag Propagation

Miguel Marinhas da Silva (29727)

Orientador: Prof. Doutor João Miguel da Costa Magalhães

Trabalho apresentado no âmbito do Mestrado em Engenharia Informática, como requisito parcial Para obtenção do grau de Mestre em Engenharia Informática.

Lisboa

20 de Abril de 2011

Automated Image Tagging through Tag Propagation

Nº do aluno: 29727

Nome: Miguel Marinhas da Silva

Título da dissertação:

Automated Image Tagging through Tag Propagation

Palavras-Chave:

Anotação de multimédia

Aprendizagem automática

Classificação multimodal

Tags

Extracção de metadados

ImageCLEF

Keywords:

Multimedia annotation

Machine Learning

Multimodal classification

Tags

Metadata extraction

ImageCLEF

Agradecimentos

Esta tese representa não só o meu trabalho de mestrado, mas é sobretudo o culminar do meu percurso académico. A vida académica para mim foi como acordar no breu da noite, lentamente e com alguns tropeções fui conseguindo ligar luz atrás de luz, até que consegui descobrir o meu caminho.

Em primeiro lugar gostaria de agradecer ao meu orientador, Professor *João Magalhães*, que me aceitou como seu orientando, tendo eu a honra de continuar o trabalho desenvolvido na sua tese de doutoramento. Obrigado pela paciência que teve comigo, por me desafiar a melhorar, por me guiar no rumo certo, mas sobretudo pela oportunidade única que me concedeu de trabalhar num projecto com o qual tanto aprendi.

Gostaria de agradecer também à *Quidgest* que me permitiu continuar os meus estudos em detrimento do trabalho na empresa. Foram cruciais no meu crescimento pessoal e profissional. Obrigado *Carlos*, *Cristina* e *João Paulo*. Obrigado também a todos os colegas de trabalho durante estes anos.

Agradeço à *Fundação para a Ciência e Tecnologia* a bolsa concedida que me permitiu poder focar apenas no meu trabalho mesmo durante um período de crise ecónomica.

A todos os meus amigos e companheiros o meu apreço. Ao *David* e *Zé Miguel* em particular pela ajuda na revisão da tese e interessantes discussões sobre a vida e tecnologia. Ao *Filipe O.*, *Filipe A.*, e *João* pela amizade e permitirem-me momentos de distracção. Aos colegas de casa pela comida e conversa.

Por último mas não menos importante gostaria de agradecer à minha familía por me ter suportado em mais que uma maneira durante todo este tempo. Um especial abraço ao avô *Zé* e à avô *Cristina* duas pessoas fundamentais no meu crescimento.

Obrigado *Pai*, *Mãe*, *Joana* e *Teresa*, esta tese é também para vocês.

Venha o próximo.

Abstract

Today, more and more data is becoming available on the Web. In particular, we have recently witnessed an exponential increase of multimedia content within various content sharing websites. While this content is widely available, great challenges have arisen to effectively search and browse such vast amount of content. A solution to this problem is to annotate information, a task that without computer aid requires a large-scale human effort. The goal of this thesis is to automate the task of annotating multimedia information with machine learning algorithms.

We propose the development of a machine learning framework capable of doing automated image annotation in large-scale consumer photos. To this extent a study on state of art algorithms was conducted, which concluded with a baseline implementation of a *k*-nearest neighbor algorithm. This baseline was used to implement a more advanced algorithm capable of annotating images in the situations with limited training images and a large set of test images – thus, a semi-supervised approach.

Further studies were conducted on the feature spaces used to describe images towards a successful integration in the developed framework. We first analyzed the semantic gap between the visual feature spaces and concepts present in an image, and how to avoid or mitigate this gap. Moreover, we examined how users perceive images by performing a statistical analysis of the image tags inserted by users. A linguistic and statistical expansion of image tags was also implemented.

The developed framework withstands uneven data distributions that occur in consumer datasets, and scales accordingly, requiring few previously annotated data. The principal mechanism that allows easier scaling is the propagation of information between the annotated data and un-annotated data.

Sumário

O volume de informação disponível na *Web* é, nos nossos tempos, cada vez maior, em particular tem-se assistido recentemente a um crescimento exponencial de conteúdos multimédia. Embora estes conteúdos sejam facilmente disponibilizados aos utilizadores, existem grandes dificuldades na procura e pesquisa sobre este conjunto tão vasto de informação. A solução para este problema é a anotação da informação, uma tarefa que sem a assistência de automatismos computacionais requer um grande esforço de trabalho humano. O objectivo desta tese é a automação da anotação de informação multimédia utilizando algoritmos de aprendizagem máquina.

Propomos o desenvolvimento de um framework de aprendizagem máquina capazes de executar anotação automática em repositórios públicos de imagens, tais como *Flickr* ou *Picasa*. Para este efeito foi efectuado um estudo sobre o estado da arte de algoritmos de anotação automática de imagens, concluindo com uma implementação base do algoritmo de *k*-nearest neighbor. Esta implementação base serve o propósito de preparação do trabalho para a construção de um novo algoritmo capaz de anotar imagens com poucas imagens de treino e em grandes conjuntos de imagens de teste – isto é uma abordagem semi-supervisionada.

Foi também efectuado um estudo adicional sobre as características de imagens de forma à sua integração no framework desenvolvido. Analisámos em particular o fosso semântico entre as características que compõem a imagem, e como evitar ou mitigar este fosso. Foi também analisada a forma como os humanos interpretam as imagens efectuando uma análise estatística nas tags das imagens inseridas pelos utilizadores. O resultado desta análise foi a implementação de um framework de expansão linguística e estatística das tags das imagens.

O algoritmo desenvolvido lida com distribuições de dados com ruído como repositórios públicos de imagens, e escala de acordo com o volume de dados. Este algoritmo necessita de um conjunto menor de dados previamente anotados, sendo um dos mecanismos principais para conseguir obter esta escalabilidade a propagação de informação entre os dados.

Contents

1

Introduction

Librarians have always catalogued/annotated books and other documents as part of their job. Similarly, professional annotators have also annotated multimedia information (e.g., TV interviews, professional stock photos) as part of their jobs. The task of adding metadata to information consists in the association of an annotation to a document, i.e. a photo of a dog might have the annotation "dog" or "animal" associated to it. The goal is to make information available to users, either through search applications (pull) or recommender applications (push). With today's increasing amount of data in the World Wide Web, namely in multimedia contexts (e.g. YouTube[1], Flickr[2]), and the information search paradigm well established (e.g. Google[3]) there is a need to correctly organize and annotate multimedia information. A vast amount of un-annotated data is available while annotated data is scarce and requires, most of the times, expensive human effort to be annotated. Figure 1 depicts an example of search by keyword, namely "sky", with its accompanying returned results.

Automating the task of information annotation or cataloging is especially problematic for multimedia content such as images and video because most search engines discard visual (colors, textures) and audio features (pitch, rhythm) and only consider the textual part of content (such as filename or text surrounding the item). The additional effort to embed both text and visual data in a multimedia annotation task can produce

[1] http://www.youtube.com/
[2] http://www.flickr.com/
[3] http://www.google.com/

significant improvements. The result of this effort is beginning to show up in search engines, such as Google's content-type filters that detect visual features, although there is prevalence of textual features usage over visual features.

When viewing a photo, humans naturally extract concepts from it that can be related to the photos time, location, scene or event among others. Further concept detection can be made if a group of photos is available instead of just one photo, making it possible to establish comparisons and extract similar concepts that were otherwise hidden. This process is natural to humans but it is much more complex in machines. In the case of machines, annotation requires knowledge of the existing features in the photo, either supplied by humans in the form of tags (user annotations), or embedded in the photo (visual features) to infer the presence of concepts. The process of inferring the presence of concepts in multimedia information, through image annotation, is the core problem addressed by this thesis.

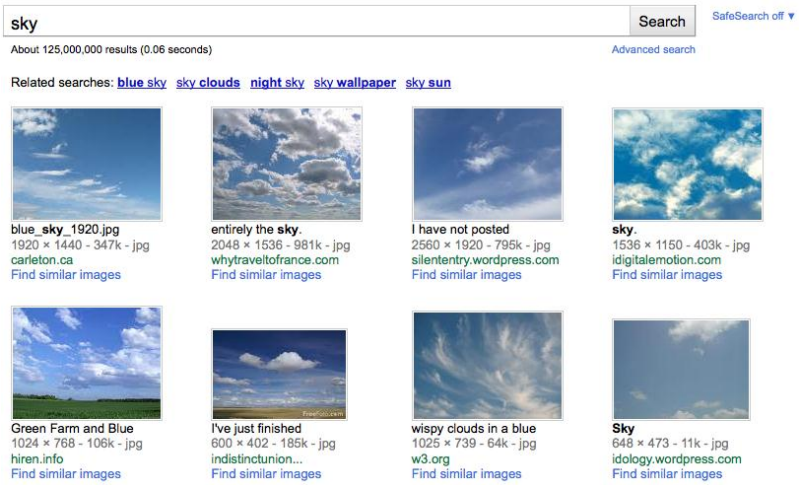

Figure 1 - Google Search by the keyword sky.

1.1 Image metadata

An image is a multimedia document that can be interpreted in many different ways. Depending on the task, an image can be represented by its set of visual features, textual features and annotations. These different interpretations of an image can be seen in Figure 2.

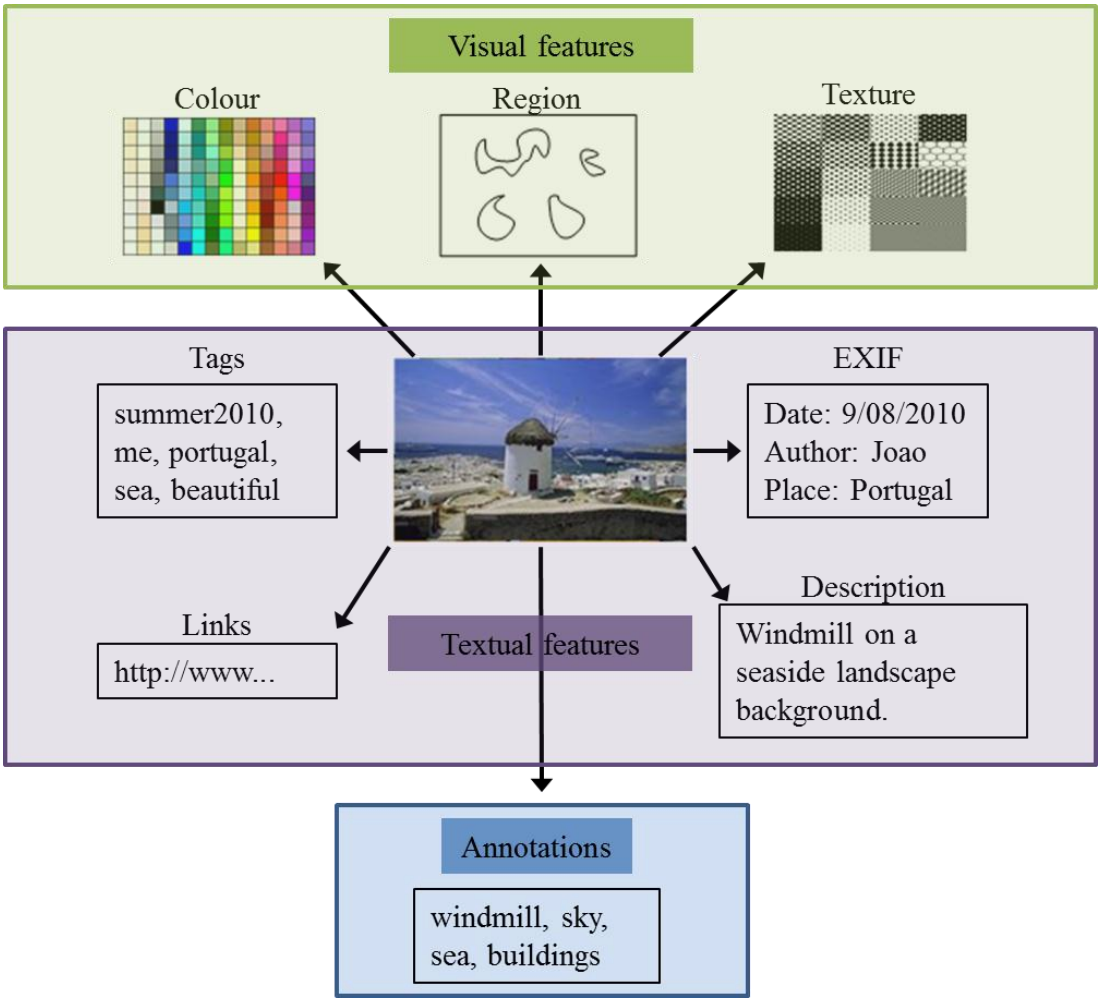

Figure 2 - The different features spaces of an image.

These different characteristics are called feature spaces and each one is defined by the method to compute the representation of an image on a given feature space. In this thesis we will characterize an image as a vector:

$$d = \{d_E, d_V, d_T, d_A\}$$

composed by a feature vector d_T describing the text part of the image, a feature vector d_V describing the visual part of the image, a feature vector d_E containing the capturing device metadata and the vector d_A containing annotation confidence scores. More specifically, the image vector d is composed by the feature vectors:

- d_E contains the EXIF metadata added by the capturing device, e.g., video camera, phone camera;
- d_T contains the tags added by the users. Usually it is a cleaned version of the user annotations (spell checked, stemmed etc.);
- d_V contains low-level visual features such as texture, colour or shape;

- d_A contains annotation confidence scores concerning the presence of the corresponding concept in that image.

The objective of this thesis is shown in Figure 2. Using the visual feature vector d_V (green rectangle) and textual feature vector d_T (purple rectangle) features we aim to annotate images concerning concepts presence using the vector d_A (blue rectangle).

1.1.1 Visual data

Visual data or low-level features corresponds to analysis made upon the visual information (pixels) contained in the image as seen in the green rectangle for visual features in Figure 2. The analysis done will be of two forms, color-based (HSV color moments) and texture-based (Gabor [1] and Tamura [2]) allowing to extract feature descriptors.

The feature vector d_V used to define the visual features is formally defined as:

$$\boldsymbol{d_V = \{d_{V1}, \ldots, d_{VT}\}}$$

where each $d_{V,t}$ corresponds to a given feature space.

1.1.2 Keywords and concepts

The term *keyword*, in our work, corresponds to the linguistic representation of a concept. The scope of these concepts can be as diverse as sports, people or art. Concepts are considered high-level features because they require a considerable level of knowledge and perception to understand the reality captured in a multimedia document. In our work we will use the concepts presence as the semantics of the multimedia document. Concepts may not be explicitly present in the multimedia information of the document and methods are required to compute the likelihood that the concept is actually present in the multimedia document.

1.1.3 User tags

An image tag is a keyword assigned to a multimedia item by a user to describe the item. In Figure 2 an example of tags is shown in the "Tags" box in the textual features (purple rectangle). Tags are keywords inserted by users, and as such suffer from user subjectiveness therefore having variable truth-value and relevance. Although the problems from such subjectivity could be amplified in large multimedia datasets, research has shown [3] it's common to obtain a consensus on the vocabulary used in tags on such datasets, even in the absence of a central controlled vocabulary.

The existence of user generated content makes its subjectiveness and relevance an issue, with a high probability of tags that can be considered noise. These tags will have to be filtered through the usage of spell-checkers, thesaurus or by using only tags with a high

term frequency. The filtering will reduce the number of words with low frequency in the tag space, thus reducing the long tail effect.

The set of resulting tags, also known as vocabulary, after the application of noise filters will be:

$$\boldsymbol{W_T = \{t_1, \dots, t_N\}}$$

Where W_T defines a lexicon of N tags used to tag multimedia documents. With this vocabulary we can construct the tag feature vector as depicted below:

$$\boldsymbol{d_T = (d_{t,1}, \dots, d_{t,N}) \in \{0, 1\}^N}$$

The d_T vector represents the tag information of N tags from the vocabulary W_T, where each component $d_{t,1}$ indicates the presence of tag i in the document *d*.

1.1.4 Annotations

An annotation in the context of multimedia is a keyword assigned to a multimedia item that describes it and is known to be truth (sometimes it is considered the ground-truth). An example of annotations in an image is shown in Figure 2 (blue rectangle). To describe the semantics of multimedia information we define the set

$$\boldsymbol{W_A = \{w_1, \dots, w_L\}}$$

as a vocabulary of L keywords which will be used as annotations. These keywords are linguistic representations of abstract or concrete concepts that we want to detect in multimedia documents. The vector d_A is formally defined as:

$$\boldsymbol{d_A = (d_{W,1}, \dots, d_{W,L}) \in [0, 1]^L}$$

where each component $d_{W,t}$ contains the annotation w_t confidence score concerning the presence of the corresponding concept in that particular document.

The semantic description of multimedia information, the vector d_A, is the core topic of this thesis.

1.2 Motivation: Image annotation

Automated image annotation has a varied range of applicability, from detecting violent or lewd content to medical imaging. The main motivation behind this thesis is to enable automated annotation in consumer multimedia (photos). This is especially useful when integrated with a multimedia hosting solution or in a multimedia retrieval system to complement searches and categorize multimedia accordingly. The machine learning

algorithms used for automated annotation are varied and produce different outcomes depending on the data and used features. Because of this variety of methods and heterogeneity of features and data, further study is needed in this field. Challenges such as *ImageCLEF*-Photo Annotation (part of the CLEF[4] challenges) promote the research and development of algorithms that can enhance the performance of automated image annotation. The photo annotation challenge is described as [4]:

> *"The visual concept and detection and annotation task is a multi-label classification challenge. It aims at the automatic annotation of a large number of consumer photos with multiple annotations."*

Following this premise, *ImageCLEF* propose to solve the problem with one or more approaches that mainly vary on the features used for the automated annotation:

- Automatic annotation with visual information only (low-level visual features)
- Automatic annotation with Flickr user tags (mid-level features)
- Multi-modal approaches considering visual information and/or Flickr user tags and/or EXIF information

Regardless of the approach used, the task remains the same: annotate the photos of the test set with a predefined set of annotations. These annotations indicate the concepts presence in the image. The concepts used are for example abstract categories like "Family & Friends" or "Partylife", the time day (i.e. "day", "night") or Animals (i.e. dog, bird). For an example of these concepts see Table 1. This list of concepts illustrates how some of the concepts are rather abstracts such as "architecture" and "transport", and others such as "baby", can have multiple meanings depending on context. This means that at least a simple form of semantic treatment will have to be made.

General topic	Subtopics
Sky	Clouds
Water	lake, river, sea
Animals	dog, cat, bird, fish, horse, insect
Plants	flowers, trees
People	portrait, boy/man, girl/woman, baby
man-build structures	architecture, building, house, city/urban, bridge, road/street

Table 1 - The ImageCLEF list of concepts.

[4] http://www.clef.org/

The need for different approaches derives from the inherent variation of performance in machine learning algorithms for different features. A specific algorithm can provide good results with a set of features but may perform poorly when those features change. It is because of this that different approaches need to be tested, to discover the best pairs of algorithm/features for each approach or a pair that can maintain an optimal level of performance in all approaches. Each approach only differs in the features used.
We will use two datasets in our research, the MIR-Flickr [4] and NUS-WIDE [5] dataset which will be discussed in the next chapter.

1.3 Objective

The goal of automated image annotation is to provide the multimedia items (in this case images) with text annotations, and consequently concepts, enabling a better semantic description of the item. An enriched set of semantic annotations about the item, makes the task of searching for those items significantly more accurate (precision metric) and complete (recall metric).

The objective of this thesis is to study automated multimedia annotation algorithms, and propose a novel graph based framework to propagate tags among images and consequently annotate images regarding concept presence.

1.4 Proposed Framework

For this thesis a framework was developed to further study annotation algorithms with the various features available. This framework can be depicted in the following diagram:

- **Feature spaces**: The framework allows the researcher to specify which features are used from the set of existing feature spaces. After a set of features have been chosen, this selection is passed to the machine learning algorithm.
- **Machine learning algorithm**: After specifying the features, a collection of training images is chosen to be used in the algorithm. This training set will be annotated with correct annotations (ground-truth) and will provide a prediction function/model. Afterwards the machine learning algorithm will annotate the test set using the prediction function.

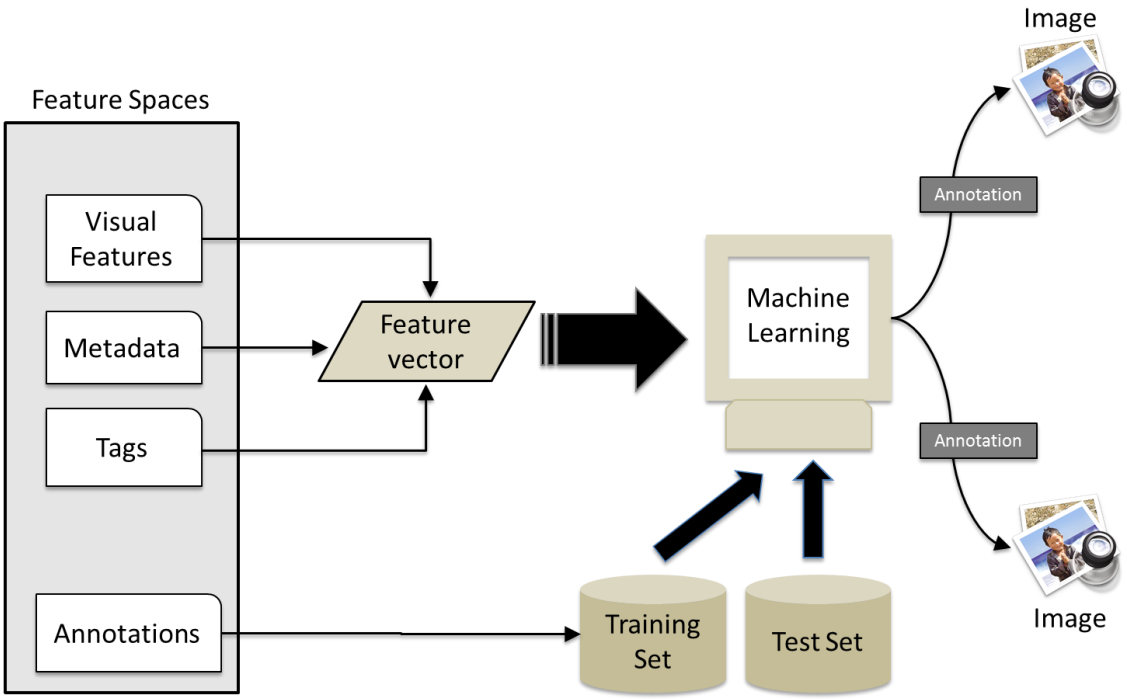

Figure 3 - Base image annotation framework.

1.5 Organization

The chapters in this thesis were arranged to be as much self-contained as possible with novel contributions in chapter 4 and 5. The organization of this thesis is as follows:

- **Chapter 2** – reviews the state of the art in image annotation algorithms and background information in feature extraction.
- **Chapter 3** – describes the framework developed and the baseline image annotation algorithms used in the framework for each feature space with the accompanying results.
- **Chapter 4** – describes an analysis made on user tag model and the improvements made upon it.
- **Chapter 5** – describes the improvements made to the baseline algorithm using the results and lessons learned in the previous chapter to create a multi-feature iterative algorithm.
- **Chapter 6** – concludes this thesis presenting the findings and contributions made.

2

Related Work

2.1 Introduction

This chapter surveys the related work considered to this thesis and is divided in the following form:

First we discuss visual and textual feature spaces. Regarding textual features we elaborate on the nature of image tags, motivations for tagging and their relevance. The next section discusses visual features and their relevance. An introduction to machine learning based approaches to image annotation algorithms is made in the following section, particularly graph-based approaches relevant to this thesis. The final section of this chapter will present a summary of the image datasets and metrics used throughout this thesis.

2.2 Textual feature descriptors/Social media tags

This section discusses social media tags relevance and is based in the work of Magalhaes [6] where we will study relevance as the central concept of Information Retrieval. It has been widely studied in different areas as the extensive review presented by Mizzaro [7] shows. Mizzaro claims that relevance is a complex concept involving different aspects: methodological foundations, different types of relevance, beyond-topical criteria adopted by users, modes of expression of the relevance judgment, dynamic nature of relevance, types of document representation, and agreement among different judges. In this discussion we leave some aspects aside and merge the remaining aspects into two practical facets that are important to the design of semantic-multimedia information retrieval: **types of relevance**; **incomplete and inconsistent relevance judgments**.

Several research areas have their own definition of relevance giving more emphasis to their specific objectives – IR aims at finding documents that *best* answers an information need, i.e. the most relevant documents for a particular user query. Information retrieval relies on datasets of documents whose relevance for a given query was judged by a human. Unfortunately, there is no universal definition of what a relevant document is: the notion of a relevant document is diffuse because the same document can have different meanings to different humans. This has been discussed by several researchers that noticed discrepancies between relevance judgments made by different annotators, see [8] and [9]. These discrepancies are more visible in large multimedia collections for two reasons: (1) multimedia information is not as concrete as textual information, thus more open to different interpretations and relevance judgments (types of relevance); (2) assessing the relevance of documents is an expensive task involving humans during long periods of time, thus collections with a large number of documents are only partially annotated: relevance judgments are incomplete and inconsistent.

2.2.1 Motivation for tagging

Although the benefits of tagging in the information retrieval domain are immense, there isn't a strong motivation for users to tag. Ames et al [10] explored motivations and incentives for tagging through the usage of photo tagging applications (e.g. ZoneTag. ESP game). In [10] it is hypothesized that multiple motivations are a determinant factor in users decision to annotate, especially social incentives. It is shown that incentives, such as point of capture tagging (tagging directly in the recorder devices) and tag suggestion improve significantly user motivation to tag.

2.2.2 Incomplete and inconsistent tagging

Another practical problem concerning relevance in very-large scale collections is the incompleteness and inconsistency of relevance judgments. In some situations the evaluation collection is so large that human assessors cannot judge all possible documents (incomplete relevance judgments), and sometimes different annotators give different relevance judgements to the same document (inconsistent relevance judgments). These trends have been extensively studied by Voorhees [8] and Buckley and Voorhees [11] who proposed a metric to reduce the effect of incomplete relevance judgments. More recently Aslam and Yilmaz, presented more stable metrics in [12, 13] to tackle the stability of measures under these conditions (incomplete and inconsistent relevance judgments).

One of the most important studies of human relevance judgments of multimedia information is the one presented by Volkmer, Thom, and Tahaghoghi [14]. They describe

and analyze the annotation efforts made by TRECVID participants that generated the relevance judgments of all training data for 39 concepts of the high-level feature extraction. To overcome the problems of incomplete and inconsistent relevance judgments the following rules were followed:

- Assessors annotated a subset of the documents with a subset of the concepts; this avoids the bias caused by having the same person annotating all data with the same concept.
- All documents must receive a relevance judgment from all annotators; this eliminates the problem of incomplete relevance judgments but increases inconsistency.
- Documents and concepts were assigned to annotators so that some documents received more than one relevance judgment for the same concept; this eliminates the inconsistency problem if a voting scheme is used to decide between relevant and non-relevant.

We stress the fact that this annotation effort was done on training data that is usually much larger than test data. So, the same problems of incomplete and inconsistent relevance judgments exist when systems are evaluated. This large scale effort was highly valuable for two reasons: it produced high-quality annotations of training data; and it gave important information on how humans judge multimedia information for particular queries, see [14] for more details.

2.2.3 Types of relevance

Systems are evaluated on collections of documents that were manually annotated by human assessors. According to the information domain, different definitions of relevance are more adequate than others. We have identified three types of relevance that are valuable to evaluate multimedia information retrieval:

- *Binary relevance*: under this model a document is either relevant or not. It makes the simple assumption that relevant documents contain the same amount of information value. This approximation results in robust systems that achieve similar accuracy across different query types, [15].
- *Multi-level relevance*: one knows that documents contain information with different importance for the same query, thus, a discrete model of relevance (e.g., relevant, highly-relevant, not-relevant) enables systems to rank documents by their relative importance. This type of relevance judgments allows assessors to rate documents with different levels of relevance for a particular topic.

- *Ranked relevance*: when documents are ordered according to a particular notion of similarity. An example of this type of relevance is when studying different image compression techniques users are asked to order compressed images by their quality in relation to the original.

The binary relevance model is a good reference to develop IR systems that serve a wide variety of non-specialized IR applications – the system is tuned with a set of relevance judgments that reflect the majority of human assessors' judgments. Voorhees [16] has showed empirically that systems based on binary relevance judgments are more robust and stable than the ones based on multi-level relevance judgments. This happens because in the second case, systems use a fine-grain model to create a rank with N groups corresponding to the different level of relevance. The ranking algorithm has the task of placing each one of the M documents in the correct group of relevance level. It is easy to see that this task is much more difficult and tuning such algorithms will easily lead to an overfitting situation that is less general, and therefore less robust and stable [16].

The relevance judgments of the ranked relevance model are actually a rank of documents that exemplify the human perception of a particular type of similarity, e.g., texture, colour. The similarity function expressed by the rank is the ranking algorithm that is approximate. For this reason, these systems (and the evaluation metrics) are more stable and less prone to overfit than multi-level relevance systems. A disadvantage of this ranked relevance is the exponentially increasing cost of generating the ranked relevance judgments.

2.2.4 The category of a tag

First it is important to explore how users tag a specific item, and what their motivations are. A parallel can be established with user search queries where search motivations are very diverse. In [17] and [18] a study was conducted on user search query patterns and found a paradigm shift between navigational searches (used to find websites) to more informational searches (i.e. information about a given topic) and resource searches (i.e. obtaining a resource). It was through this analysis on user search behavior that new information about search patterns arose. In social media tagging, understanding why users tag is the first step to better understand user tagging behavior. These behaviors, or patterns, can be organized into a taxonomy of user tagging motivations. Some of the most common behaviors are low-level tagging which describe colors or visual elements, but more complex behaviors arise from understanding tagging motivations. Tagging behavior patterns such as referencing something personal and mostly known to the user (i.e. me, art), collaborative tagging, where an image is annotated following a collaborative tagging

effort (i.e. a large group of users attributing the "*interesting*" tag to an image). A summary of tag patterns is shown in Table 2, where the taxonomy of tags is shown, exemplifying tagging behaviors. By understanding user tagging motivations we can better assess image tags relevance.

2.2.5 The subjectiveness of a tag

The subject of our study has a varied and complex nature. While it starts by being a simple keyword inserted by the user, according to his judgment, correctly defining a tag's type needs a multitude of research domains ranging from natural language processing to computer vision. In Table 2 the tag type taxonomy used in this chapter is shown along with examples.

Category	Examples
Ambiguous	Camera, explore
Collaborative	Abigfave, interestingness
Author	Me, art
High-level	Portrait, sports
Mid-level	Sky, ruby
Low-level	Blue,bw
EXIF-location	London, Fifth Avenue
EXIF	Canon, 2007

Table 2 - Tag type taxonomy with examples.

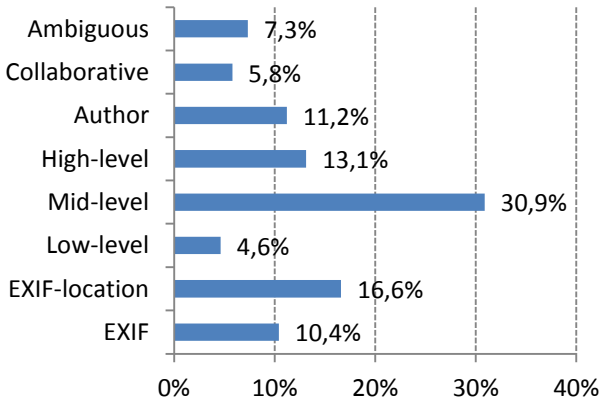

Figure 4 - Distribution of tag type taxonomy in MIR-Flickr.

The tag type taxonomy used in this chapter is based on the following categories:

1) *EXIF*: Metadata embedded in the image related to the device used to capture the image. An example of this is the information regarding the maker and model of the camera;

2) *Geo-tag/EXIF-Location*: Metadata available in certain devices that references a geographical location, usually where the photo was taken;
3) *Low-level*: pertains to the visual content of an image;
4) *Mid-level*: the most common features, including any generic keyword;
5) *High-level*: these correspond to the annotations, keywords with ground-truth certainty;
6) *Author*: these are directly related to the owner/user subjective assessment of the image;
7) *Collaborative*: these are related to the interpretation of a group of people, hence collaborative;
8) *Ambiguous*: in this type of features there can be multiple interpretations of the keyword depending on the viewer and context. There can also exist an overlapping between other types of tags. Ambiguous can be very prominent in a dataset and take various forms as detailed by Weinberger et al [21]:
    a) *Semantic*: the tag "bass" has two types of meanings;
    b) *Geographical*: the tag "Cambridge" can correspond to two different places;
    c) *Temporal*: the tag "worldcup" can correspond to multiple events (2006, 2010 for instance);
    d) *Language*: the tag "mist" means dung in German and fog in English;
    e) *Generalization*: while not being technically an ambiguity, the usage of a generalist tag can induce in error, the tag "Europe" for example can be overly common and won't introduce relevant information to the annotation algorithm;

2.3 Image feature descriptors

2.3.1 Hue Saturation Value color histogram moments

One of the most commonly used features is the color histograms, which represent the distribution of various color ranges in an image. In particular we will use the Hue, Saturation and Value (HSV) color space. Unlike the RGB color space, the HSV color space reflects the human perception of color similarity. This implies that two colors considered similar by humans will have a small distance in the HSV color space, i.e., color similarity is inversely proportional to distance this cylindrical-coordinate system.

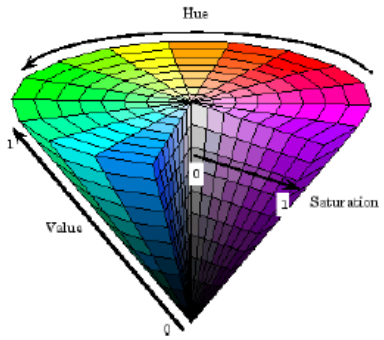

Figure 5 - HSV color space.

The dimensions of the HSV color space correspond to the Hue, Saturation and Value, depicted in Figure 5. If we consider each channel independently one can compute the marginal histograms. Figure 6 depicts an image in its original RGB format. When converted into the HSV color space and printed as an RGB picture one can visualize the color space transformation. The bottom graphs depict the histogram of each HSV dimension (Hue, Saturation and Value).

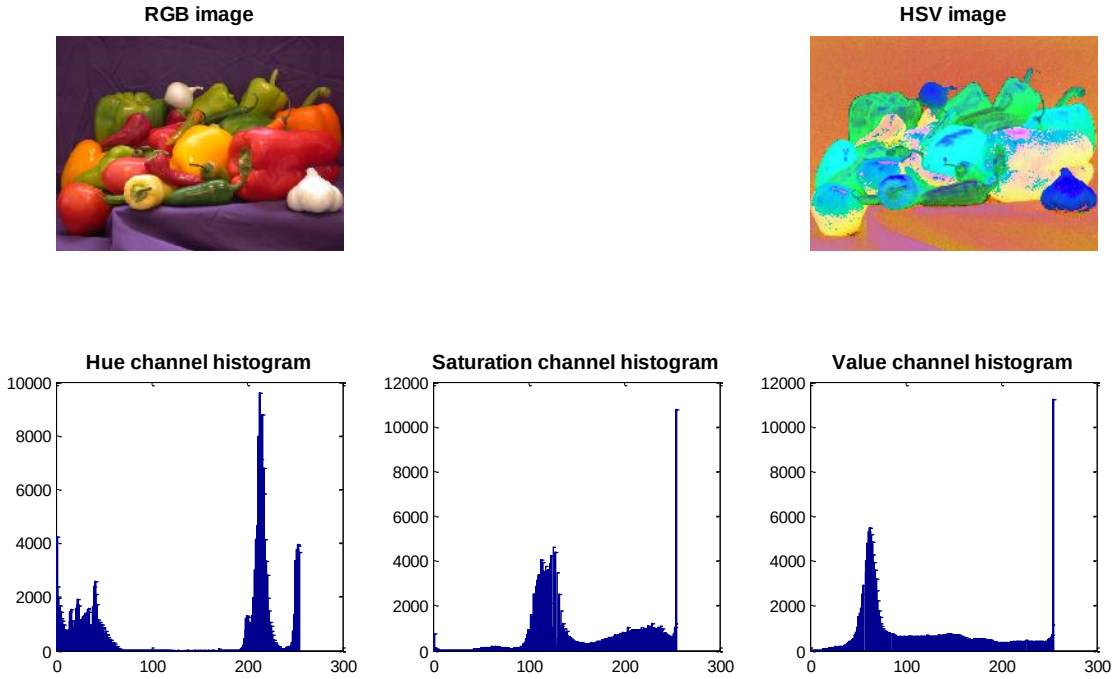

Figure 6 - Example of the "peppers.png" RGB image, the HSV image, and the marginal HSV color histograms.

Several descriptors can be computed from these histograms. Lower dimensional histograms can be computed, e.g., with 16 bins per histogram, and the moments of these histograms, e.g., mean, mode, variance. In this thesis we use the mean and the variance of

each color channel. Moreover, we divide the image in 3 by 3 blocks, giving 9 regions per image and compute mean and the variance of the histogram of each HSV color space dimension. Thus, the HSV color feature vector is:

$$\boldsymbol{d_{V1} = (d_{V1,1}, \ldots, d_{V1,54})}$$

which corresponds to 9 sub-images (3 by 3 tiles), 3 color channels (H, S and V) and 2 histogram moments (mean and variance), totaling 54 dimensions.

2.3.2 Tamura features

Cognitive vision experiments have been conducted by several scientists to understand how humans perceive textures. Based on a user experiment, Tamura et al. proposed the definition of six texture aspects that were commonly used by users to describe textures. These texture characteristics are coarseness, contrast, directionality, regularity or ruggedness. Users were then asked to rank a set of images according to their coarseness, then by their contrast, and so on. The resulting ranks for the top three characteristics are presented in Figure 7.

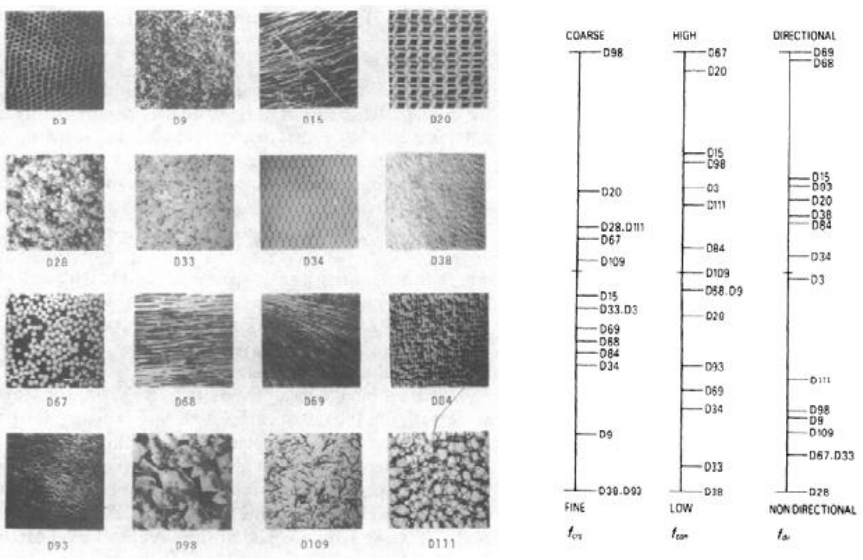

Figure 7 - Tamura features [19].

To emulate the human visual perception system, Tamura et al. proposed algorithms to reproduce these results. As a result these algorithms are now widely used to compute the coarseness, contrast and directionality characteristics of visual content.

We compute three Tamura texture features (coarseness, contrast and directionality) of all images as follows: images are divided into 9 tiles (3 by 3 rectangles) and the mean and the variance of each one of the three characteristics are computed. The result is a 27 dimensional feature vector:

$$\boldsymbol{d_{V2}} = \left(\boldsymbol{d_{V2,1}}, \ldots, \boldsymbol{d_{V2,27}}\right).$$

2.3.3 Gabor filter moments

A texture can be seen as a combination of different edges at different orientations and scales. This principle is applied by the Gabor filters that scan the image with Gaussian filters to decompose the original image into several images representing the different edges of the original image. These computed images represent the edges at a scale and direction corresponding to the configuration of the Gabor filter, Figure 8.

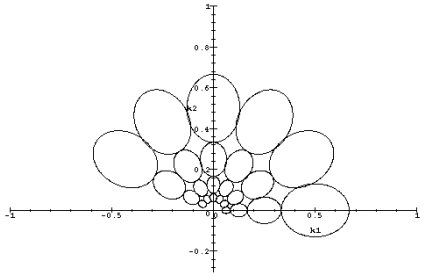

Figure 8 - Plot of the Gabor filters in the frequency plane [1].

Figure 9 represents the result of the application (i.e. convolution) of Gabor filters with six different directions (0º, 30º, 60º, 90º, 120º and 150º) and four different scales over the original image. As the filter scales increase, the edges become coarser. Also, note that as the direction of the filter moves away from the horizontal plane (0º), the detected edges have a corresponding direction.

For the experiments in this thesis we compute the Gabor filter descriptors proposed by Manjunmath et. al [1]. The visual feature vector

$$\boldsymbol{d_{V3}} = \left(\boldsymbol{d_{V3,1}}, \ldots, \boldsymbol{d_{V3,48}}\right),$$

is composed by the mean and variance of each Gabor filter. We consider 6 directions (0º, 30º, 60º, 90º, 120º and 150º) and 4 scales. This corresponds to Figure 8.

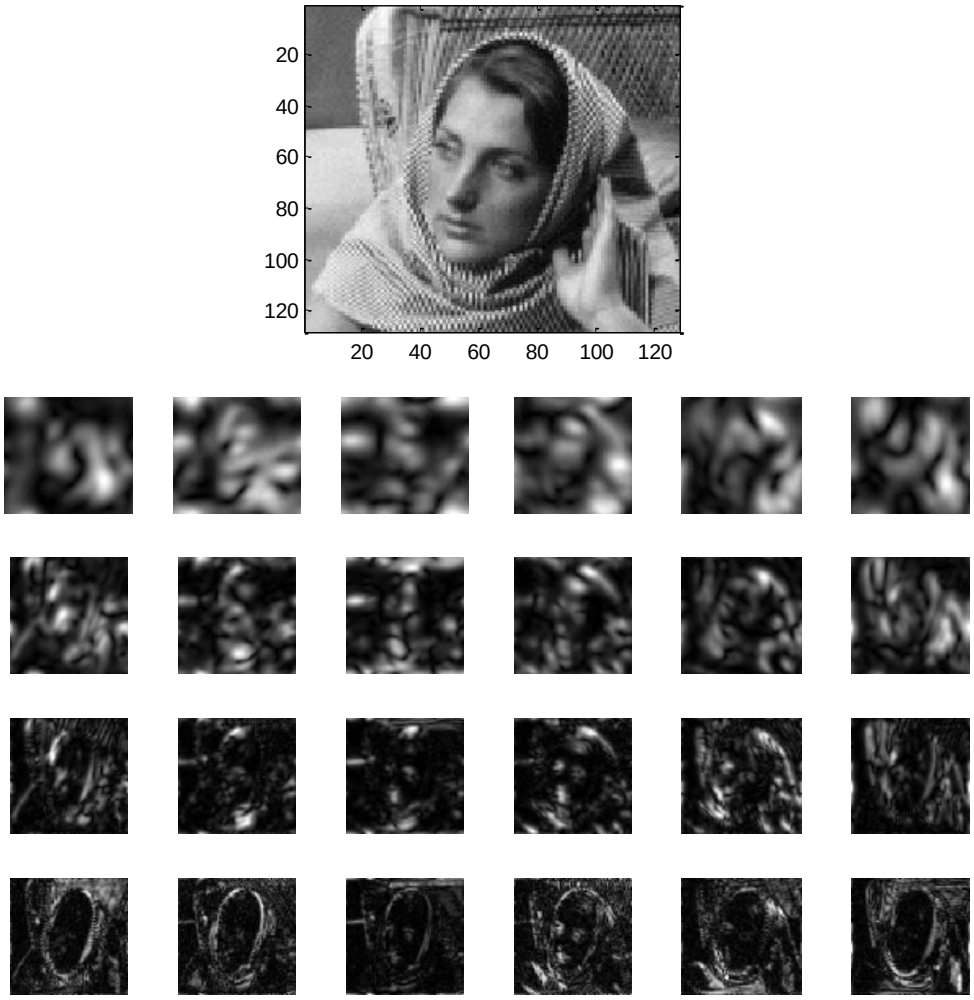

Figure 9 - Gabor filters output.

2.4 Machine learning based annotation

Machine learning is the machine equivalent of human learning. Where the Human race learns from absorbing data from their surrounding and processing it, Machine learning is a computer science discipline that studies algorithms for machine knowledge learning based on empirical data.

The process of automated image annotation belongs to the domain of machine-learning classification algorithms. Although many methods have been used, there isn't a perfect method for all types of data with varied features and data distributions. There are machine-learning methods that perform better with certain data that aren't as good with other types of data. In this thesis we will focus in supervised and semi-supervised learning algorithms:

- ***Supervised learning:*** This type of machine learning algorithms requires a set of annotated data to be used for annotation inference. From this set of data a

predictor function is estimated, in a phase that can be described as the learning phase. This predictor function will then be applied to the automated annotation of new un-annotated data.

- ***Semi-supervised learning:*** This type of machine learning technique uses both annotated and un-annotated data for the training/learning phase. As opposed to the Supervised Learning it is not required to have a full set of annotated data, which could be time consuming to annotate, but rather a small portion of annotated data with enough distinctive features that can be used to compare with the un-annotated data.

2.5 Annotation algorithms

Our research will be based around non-parametric annotation algorithms. These types of image annotation algorithms don't require a learning or training phase of the algorithm parameters. These types of algorithms, which are also known as lazy learners, use local prediction functions that are calculated whenever a new query is made about the annotation of a new image. As such they can handle a varied number of annotations, avoid overfitting of parameters, due to the local prediction, and require no learning or training phase which makes them a suitable candidate for dynamic data-sets of images. An example of this type of algorithm is the *k*-Nearest Neighbor algorithm.

2.5.1 Graph based Methods and Semantics

The usage of graphs to map images and features has become more commonly used recently due to its efficiency and scalability in solving machine learning problems and as such applicable in automated image annotation. These methods in conjunction with non-parametric annotation have the advantage of being domain independent and have generally a simple parameter tuning, which are strong points shared by general graph model method. The drawback of these methods is in the difficulty of differentiating between the relevance of different types of features/nodes in one graph.

Unlike other approaches, graph-based methods can model the relations between concepts making possible to form a relation between then, i.e. car – wheel, which enables the formation of an ontology that enables a faster propagation and annotation of those concepts. These types of promising techniques using graph-based methods are becoming prominently used in automated annotation [19-21].

2.5.2 Graph Modeling

Concept relationship can be modeled with graph-based methods by representing semantic relations using edges. Graphs can be modeled in two ways: directed graphical models and undirected graphical models. The major difference between those two types is the explicit imposition of concept causality in the Directed Graphical Models. In Figure 10 an example of undirected and directed graphs is given.

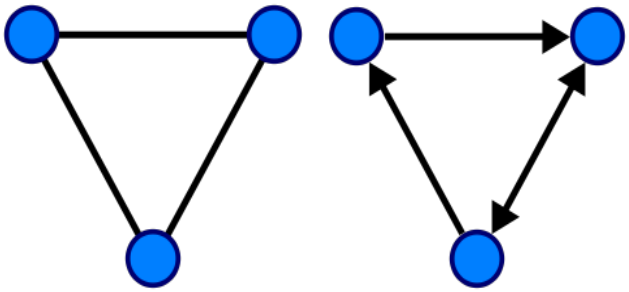

Figure 10 - Undirected (left) and directed graphs (right).

In [20] a study is conducted comparing these two types of graphs (directed and undirected) in a concept learning task. The authors in [20] present a comparison between those two types of graphical models that ascertain the potential of the undirected graphical models for the task of annotation that can detect similar semantic concepts. This conclusion is explained by avoiding the existing concept causality in directed graphs, therefore removing hidden dependencies in the graph and allowing for faster graph-model manipulation.

2.5.3 Learning with Local and Global Consistency

A vast amount of un-annotated data is available while annotated data is scarce, as such there is an added difficulty to be able to adapt algorithms to unbalanced data, which corresponds to never seen concepts or with few examples. Approaches that can combine small amounts of annotated data with un-annotated data and handle unbalanced amounts of annotated/un-annotated data can improve this problem – namely using semi-supervised learning. A further improvement on the *k*-NN algorithm that uses semi-supervised learning is developed on [19] where the key to its success is the prior assumption of consistency. This consistency is achieved in two parts:

- Local consistency – where nearby points are likely to have the same annotation, which is a natural consequence of *k*-NN algorithms.
- Global consistency - where points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same annotation.

This argument for a global consistency is also shared by various authors as referred in [19]. To implement this type of consistency a new algorithm is developed in [19] where each

point, or image in our case, will iteratively spread its annotation information to its neighbors until a global stable state is achieved. One of the advantages of this algorithm is its adaptability to include new parameters, like semantic distances, which would lead to more interesting annotation propagation.

2.5.4 Nearest Spanning Chain

Either using directed or undirected graph models the possibility to map concept relationships is a clear advantage in graph-based methods. When dealing with different feature spaces new mechanisms are needed to deal with these differences. In [21] a method to support various feature spaces (visual and textual) is created to deal with feature correlation. This method requires the construction of an adaptive graph that can support the multi-modality of features and also map their similarities. With this graph the annotations are propagated using semantic similarities from *WordNet* [22] and low-level visual features. This adaptive graph is a modification of the *k*-Nearest Neighbor algorithm with the graph organized using a *Minimum Spanning Tree* in which the authors call it the *Nearest Spanning Chain* (*NSC*). The *NSC* has good performance when compared to other annotation algorithms in controlled image data sets like the Corel Photo Database although their performance declines with "real world" image data sets with the increase of annotation errors as seen in [21].

2.6 Evaluation methods

2.6.1 Datasets

To further explore user photo tags a reference dataset from which we can draw sufficiently large samples is required. There are many different datasets with specific characteristics. Professional Stock-Photos (such as the Corel Database) are made of photographs taken by professional photographers and as such with controlled camera settings. The NUS-Wide [5] or MIR-Flickr [4] dataset which are consumer photo datasets with varied photo settings. The MIR-Flickr [4] and the NUS-WIDE [5] datasets will be used for the quantitative analysis developed in this thesis.

The MIR-Flickr dataset contains 25,000 images retrieved from the Flickr multimedia repository taking into account a high "*interestingness*" factor. This dataset supplies all original tags – user annotations – provided by the Flickr users and also annotations for all the images, which were obtained by majority voting using manual annotation with the Amazon Mechanical Turk tool. Annotations for 24 concepts are provided in this dataset. This is a significant dataset that encompasses various types of

photographs with a varied set of concepts between them. From this dataset the three previously mentioned feature descriptors (Gabor, Tamura, and Marginal HSV) were extracted to be used in our research. There are about 69,000 unique tags in the dataset and a cardinality of 9 tags per image. In Figure 11 the distribution between the various types is seen using a sample from the MIR-Flickr dataset

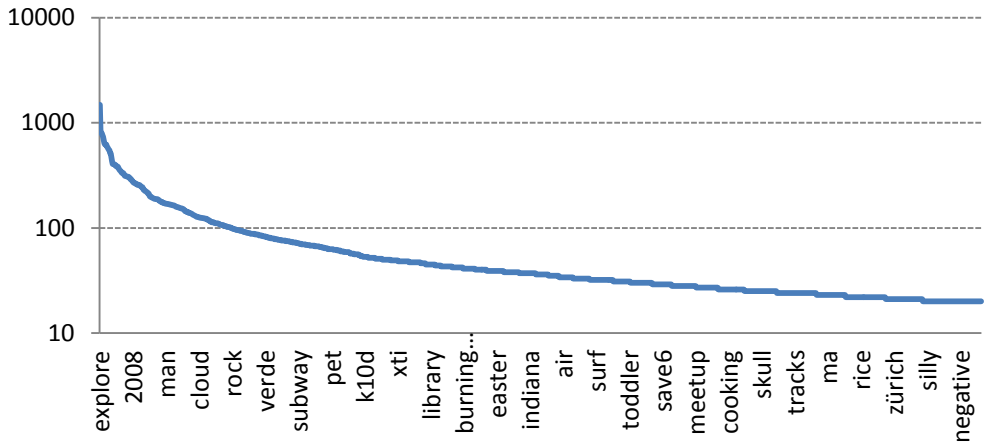

Figure 11 - Tag frequency distribution in MIR-Flickr dataset.

The second dataset used, NUS-Wide, contains 269,648 images retrieved from Flickr. Although a larger dataset, only 5018 unique tags are used throughout this dataset. Unlike the MIR-Flickr dataset, the tags have been given a slight treatment for tags that can be considered noise and there is a much lower cardinality of tags per image than MIR-Flickr. Annotations for 81 concepts are also provided for the entire dataset.

Since both datasets come from public repositories folksonomies are present with usage of a varied range of languages in user annotations (tags) and as such semantic problems like noise, varied synonyms or abbreviations need to be dealt accordingly.

2.6.2 Metrics

To conduct this study we will use the Precision and Recall metrics to assess the performance of the results:

		Correct result (concept presence)	
		Concept not present	Concept present
Obtained result	Tagged	false positive	true positive
	Not tagged	true negative	false negative

$$\boldsymbol{precision} = \frac{\boldsymbol{true\ positives}}{\boldsymbol{true\ positives} + \boldsymbol{false\ positives}}$$

$$\boldsymbol{recall} = \frac{\boldsymbol{true\ positives}}{\boldsymbol{true\ positives} + \boldsymbol{false\ negatives}}$$

The Accuracy metric will also be present in some cases, which measures the proportion of true results in the population, where a score of 100% will mean the predicted values are exactly the same as the expected values.

$$\boldsymbol{accuray} = \frac{\boldsymbol{true\ positives} + \boldsymbol{true\ negatives}}{\boldsymbol{true\ positives} + \boldsymbol{false}\ \textbf{positives} + \boldsymbol{false\ negatives} + \boldsymbol{true\ negatives}}$$

We will also use the F-measure (or F-score) metric which corresponds to the harmonic mean or weighted average of precision and recall. F-measure is given by the expression:

$$\boldsymbol{Fmeasure} = \mathbf{2} \times \frac{\boldsymbol{precision} \times \boldsymbol{recall}}{\boldsymbol{precision} + \boldsymbol{recall}}$$

The final metric used will be the root mean square error which is defined by:

$$\mathbf{RMSE}(\widehat{\boldsymbol{\theta}}) = \sqrt{(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})^{\mathbf{2}}}$$

The root mean square error or RMSE is used to quantify the difference between an estimator and the true value of the quantity being estimated. RMSE measures the root square average of the square of the error. The error is the amount by which the estimator differs from the quantity to be estimated.

2.7 Summary

In this chapter we have first researched the two types of feature spaces prominently used in this thesis, textual and visual features. Initially we discussed textual features, namely tag relevance as a central concept in information retrieval and its correlation with incomplete and inconsistent tagging that occurs in user tagging. We explored the nature of tags and tagging motivations and further researched how tags can be categorized, introducing the taxonomy of tags.

Research has also been made on the visual features used in this thesis (Gabor, Tamura and Marginal color moments), how they can be obtained and what information can extracted from them.

Second we have introduced machine learning annotation algorithms and studied annotation algorithms variations, in particular graph based methods. Their benefits come from improved efficiency, scalability and the possibility to map relationships between graph elements with concepts. Further research was made on graph-based algorithms,

namely using iterative algorithms [19] to spread information throughout the graph using a *k*-NN algorithm variation.

3

Feature-based image annotation

3.1 Introduction

Effective image annotation algorithms have to address the common problems found in machine learning classification tasks: namely overfitting and the need of a learning phase. To address these problems, non-parametric image annotation algorithms have become prominently used, in particular *k-Nearest Neighbor* algorithm (*k*-NN), Kernel density estimation (KDE) and other graph-based methods. The focus on this chapter will be on the *k*-NN algorithm, which is a non-parametric machine-learning annotation algorithm which will be used to annotate images based on the closest training examples. Given a set of images $I = \{i_1, \dots, i_l, i_{l+1}, \dots, i_n\} \subset \mathbb{R}^m$, each image represented in a common feature space, and an annotation set $W_A = (w_1, \dots, w_L)$, the first *l* images $i_i (i \leq l)$ are annotated with $w_i \in W_A$, i.e. the training set, and the remaining images $i_u = (l + 1 \leq u \leq n)$ are un-annotated, i.e. the test set. The goal is to predict the annotation of the un-annotated images, therefore performing image annotation. When a new image needs to be annotated an approximation function computes its *k* -nearest neighbors, where $Neighbors \in \{i_1, \dots, i_l\}$. This function is a majority voting performing binary annotations (in its naïve form) using its *k* nearest neighbors and is described by:

$$\boldsymbol{knn}\left(\boldsymbol{i_{u,w_j}}, \boldsymbol{k}\right) = \frac{\sum_{\boldsymbol{n=neighbors(i_u,1)}}^{\boldsymbol{neighbors(i_u,k)}} \boldsymbol{I_{n,w_j}}}{\boldsymbol{k}}$$

This formula reads as follows: for each new image i_u being annotated for keyword $w_j \in W_A$, we will average the occurrence of the annotation in the *k* most similar neighbors

of i_u. The choice of neighbors relies on the distance between the new image and its neighboring images in a given feature space. The *k* parameter determines how many neighbors will be used in that function. Choosing the best *k* parameter depends on the data, with larger values of *k* reducing the effect of noise in the image annotation algorithm, but making the boundaries between annotations less distinct. A choice for an optimal *k* value can be made using cross-validation to assert an optimal value for a given dataset.

This algorithm has an inherent graphical architecture with nodes and edges connecting the images with the surrounding neighbors which provides the possibility to establish a network of similar images or concepts. But as other machine learning algorithms it is not without disadvantages, namely the high variance in the presence of limited sampling, the inequality of importance between neighbors and being a computationally intensive algorithm.

An example of the *k*-NN annotation algorithm is given in the picture below:

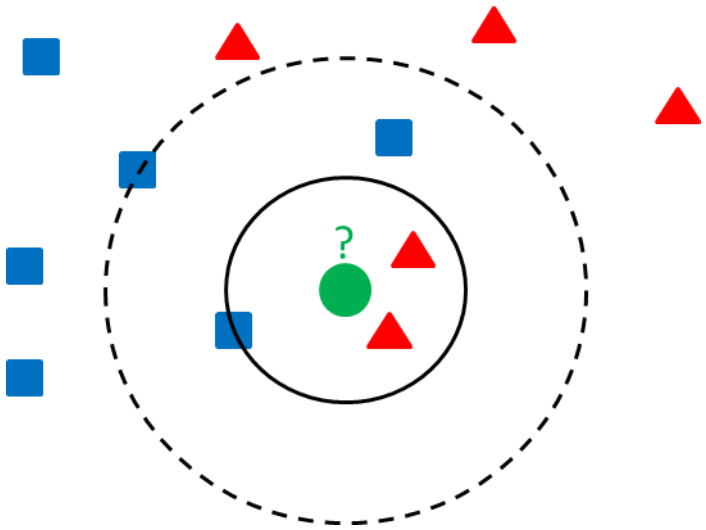

Figure 12 - NN Annotation algorithm with varying *k*.

The green circle is the new image to be annotated, the blue squares and red triangles are images with two distinct annotations. The annotation procedure goes as follows:

- If *k* =3 it means that we are in the inner circle defined by the third nearest neighbor. The neighborhood consists of two red triangles and one blue square, which by majority voting the new image belongs to the annotation of the red triangles.
- If *k* =5 it means that we are in the outer circle defined by the fifth neighbor. The neighborhood is now the two red triangles and three blue squares, which by majority voting the new image belongs to the annotation of the blue squares.

This example illustrates two problems of the *k*-NN algorithm:

- **High variance**: when in the presence of limited data, results exhibit a high-variance and sensitivity to the number of neighbors;
- **Neighbor distance**: the importance of each neighbor isn't proportional to its distance, for example for *k*=5 the new image belongs to the annotation of blue rectangles although the red triangles annotation is much closer to the new image.

This chapter implements a baseline *k*-NN algorithm which will be used to evaluate several variations of the nearest-neighbor algorithm using the various feature spaces. The first section describes the image representation and the feature spaces that compose it. The following section discusses the improvements made to the baseline implementation. The remaining sections consist in the experimental results obtained from the baseline and improved *k*-NN implementations.

3.2 Image feature vectors

To apply the *k*-NN algorithm we need to describe the image features used to represent images. There are several feature spaces than can be used to describe an image but in this chapter we will only focus on two types, tag/textual based features and visual-based features. Images will be represented using a vector space model containing the feature vectors for each feature space. A commonality between the two types of image vectors is the presence of the image annotations. These are represented by the vector d_A which contains annotation confidence scores. To describe the semantics of multimedia information we define the set

$$W_A = (w_1, \dots, w_L)$$

as a vocabulary of *L* keywords which will be used as annotations. These keywords are linguistic representations of abstract or concrete concepts that we want to detect in images. The vector d_A is formally defined as:

$$d_A = (d_{W,1}, \dots, d_{W,L}) \in [0, 1]^L$$

where each component $d_{W,t}$ contains the annotation w_t confidence score concerning the presence of the corresponding concept in that particular image.

3.2.1 Text-based feature vector

Images will be represented by two text-based feature vectors: the feature vector d_T containing the tags added by the users and d_A describing the semantics of the image,

$$\boldsymbol{d} = \{\boldsymbol{d_T}, \boldsymbol{d_A}\}.$$

The d_T vector is usually a cleaned version of the user tags (spell-checked, stemmed etc.). In this case, for simple noise reduction we will only use tags with a cardinality of at least twenty, therefore discarding tags that don't occur often in the image dataset. The set of tags, also known as vocabulary, will be:

$$\boldsymbol{W_T} = \{\boldsymbol{t_1}, \dots, \boldsymbol{t_N}\}$$

Where W_T defines a lexicon of *N* tags used to annotate images. With this vocabulary we can construct the tag feature vector formally defined as:

$$\boldsymbol{d_T} = (\boldsymbol{d_{t,1}}, \dots, \boldsymbol{d_{t,N}}) \in \{\boldsymbol{0}, \boldsymbol{1}\}^{\boldsymbol{N}}$$

The d_T vector represents the image tags, where each component $d_{t,i}$ indicates the presence or non-presence of tags in image *d*.

3.2.2 Visual-based feature vector

Previously we have explored textual features – unfortunately this type of information isn't always present in images. Low-level visual features capture the most important information encoded in the different color pixels composing the image. From this data we can gather information about the image using various techniques. In this chapter we will represent images in two ways: d_V feature descriptors concerning the visual part of the image, and d_A concerning the semantics of the image,

$$\boldsymbol{d} = (\boldsymbol{d_V}, \boldsymbol{d_A})$$

The feature vector d_V is formally defined as:

$$\boldsymbol{d_V} = \{\boldsymbol{d_{V1}}, \dots, \boldsymbol{d_{VT}}\}$$

where each $d_{V,t}$ corresponds to a given feature space. Three types of features will be computed from the images, two texture-based features (Gabor and Tamura) and one color based (HSV color moments). Thus, we shall have:

- $d_{V1} = (d_{V1,1}, \dots, d_{V1,n})$, for the HSV color moment descriptor, a *n*-dimensional vector;
- $d_{V2} = (d_{V1,1}, \dots, d_{V1,m})$, for the Gabor texture descriptor, a *m*-dimensional vector;

- $d_{V3} = (d_{V1,1}, \ldots, d_{V1,p})$, for the Tamura texture descriptor, a p-dimensional vector.

3.3 A *k*-NN framework

In this section we describe the implementation of a baseline *k*-NN algorithm. This is a non-parametric annotation algorithm that unlike others (Support Vector Machines) doesn't require a learning phase. This algorithm maintains in memory all the images and associated features, and when a new image requires annotation a new local approximation function is created according to the most similar neighbor images found.

We first established a baseline implementation using textual features. This baseline implementation was used to discover the optimal similarity score to be used in the *k*-NN framework. Below a diagram depicting the annotation workflow of the *k*-NN implementation is shown:

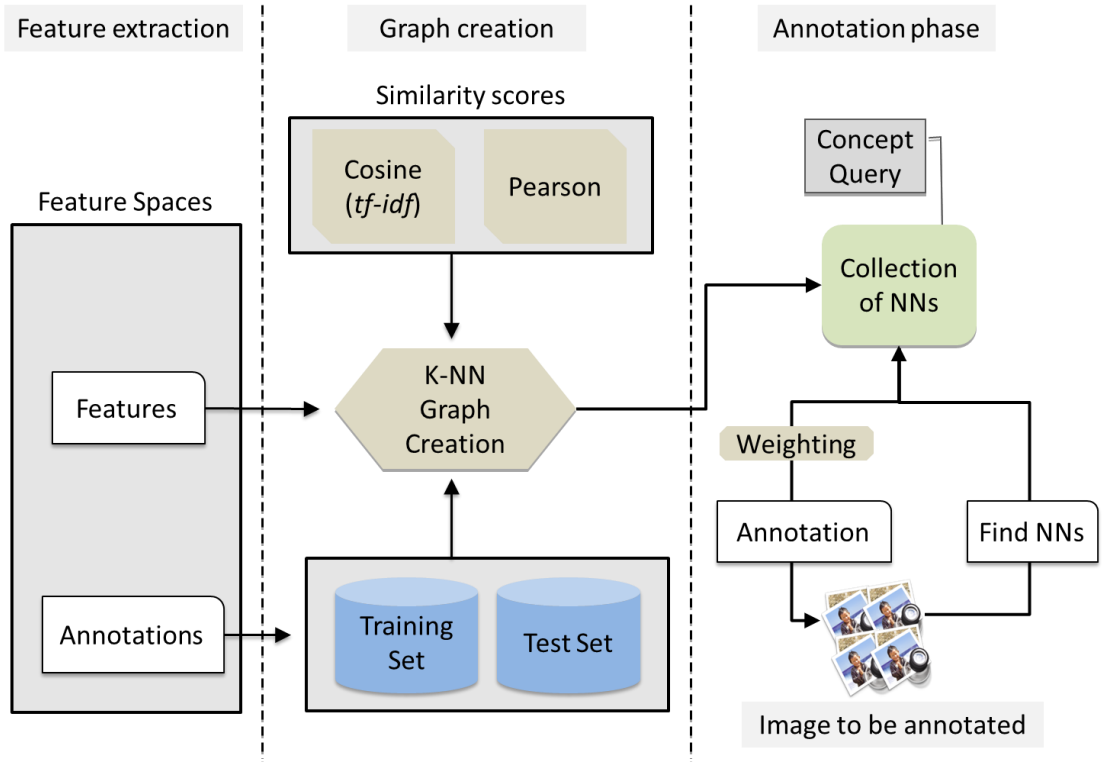

Figure 13 - *k*-Nearest Neighbor Annotation workflow.

The workflow is divided in three sections:

- **Feature extraction:** where different characteristics of images are processed/computed to be used in the creation of the feature model.

- **Graph creation:** the similarity scores between each test image and all training images are computed. Only the *k* nearest images are stored. The similarity scores are computed through the image tags.
- **Annotation phase:** where each image from the test set is annotated by an approximation function that uses its *k* nearest neighbors to vote for the presence of the concept to be annotated.

3.3.1 Similarity scores

One of the most important elements in the *k*-NN algorithm is the similarity score or distance used to find the nearest neighbors. This is a simple calculation used to discover the distance between two images and corresponds to the human notion of distance between two points in space.

In the case of two images, the Euclidean distance is given by the square root of the sum of the square of the difference between the each dimension of two feature vectors. Given two images with their feature vectors $p = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$ where p_i or q_i is equal to the dimension *i* of the image, the Euclidean distance is given by:

$$\boldsymbol{D_{Euclidean}(\mathbf{p}, \mathbf{q})} = \sqrt{\sum_{i=0}^{n} (p_i - q_i)^2}$$

This distance is mostly applicable to continuous variables, and performs worse than other distances when the data isn't well normalized or with an uneven distribution.

Other similarity scores yield better results in information retrieval, in particular the Pearson correlation and its variant, the Cosine similarity. These two distance metrics are invariant to scaling, and the Pearson correlation is also invariant to the addition of constants to its elements, disregarding the absolute value of the points being compared. The Pearson correlation is given by:

$$\mathbf{D_{Pearson}(p, q)} = \mathbf{1} - \frac{(\mathbf{p} - \overline{\boldsymbol{p}}) \cdot (\boldsymbol{q} - \overline{\boldsymbol{q}})}{\sqrt{\sum(\mathbf{p} - \overline{\boldsymbol{p}})^2} \sqrt{\sum(\boldsymbol{q} - \overline{\boldsymbol{q}})^2}}$$

The cosine similarity is a specific case of the Pearson correlation where the similarity between two vectors corresponds to the angle defined by them. The Cosine similarity is given by:

$$\mathbf{D_{Cosine}(p, q)} = \mathbf{cos(p \measuredangle q)} = \mathbf{1} - \frac{\mathbf{p} . \mathbf{q}}{\|\mathbf{p}\| \cdot \|\mathbf{q}\|}$$

These distances functions are used to search for the *k*-Nearest Neighbor of each vector. The result will be an undirected graph connecting each image to its neighbor using the distance functions.

3.3.2 Term weighting

Search engines commonly use term weighting to select the most relevant terms of a given document. Term weighting techniques allows giving emphasis to relevant terms. One of the most effective weighting technique is the term frequency-inverse document frequency (*tf-idf*) weighting. This technique gives more emphasis on terms that have a high frequency in the document but a low frequency throughout the dataset, which tends to make the weights significantly smaller on common terms. The *tf-idf* weighting is given by the expression:

$$(\boldsymbol{tf-idf})_{\boldsymbol{i,j}} = \frac{\boldsymbol{n_{i,j}}}{\sum_{\boldsymbol{k}} \boldsymbol{n_{k,j}}} \times \mathbf{log} \frac{|\boldsymbol{D}|}{|\{\boldsymbol{d}: \boldsymbol{t_i} \in \boldsymbol{d}\}|}$$

The weighting is composed by two parts, first the term frequency where a keyword importance increases proportionally with the number of times the same keyword appears in the document. And the second part, the inverse document frequency, which measures the frequency of the keyword in the collection of documents or images (it assesses how rare a keyword is).

This weighting is commonly used together with the cosine similarity in vector space models to determine the similarity between two documents. The cosine similarity yields results very close to the Pearson correlation, thus, allowing a viable alternative when used in conjunction with *tf-idf* weighting. Another advantage comes from using this improvement with the linguistic expansion and correction. Upon expanding the keywords of an image (in this case the user tags), and if duplicate keywords are added through linguistic expansion, a higher term frequency will occur for the given duplicate keyword. Therefore it will distinguish the importance of the keyword in the image and appropriately uses its new weight to calculate the similarity to other images.

3.3.3 Neighbors weighting

As previously detailed, one of the problems in the naïve implementations of the *k*-NN algorithm is the inequality of neighbors. A prime example of this is using neighbors that are too far away with having the same amount of "voting power" as other nearer neighbors. To solve this problem, the notion of weighted neighbors was introduced to

differentiate images according to the distance of the "voters" to the new data. Thus we will introduce the notion of weighted mean instead of simple voting:

$$\boldsymbol{weigthed_{knn}} = \frac{\boldsymbol{w_1 n_1 + w_2 n_2 + \cdots + w_n n_n}}{\boldsymbol{w_1 + w_2 + \cdots + w_n}}$$

This formula is a weighed mean composed by the weight of the neighbor, calculated according to the Gaussian function, which will be explained below, and the value for the neighbor (if it contains the annotation or not).

We will convert the distance to weight according to the distance to the image being annotated. The function used to convert distance to weight is the Gaussian function:

$$\boldsymbol{Gaussian_{(distance)}} = \boldsymbol{e}^{\frac{-\boldsymbol{distance^2}}{\boldsymbol{2\sigma^2}}}$$

The weight in this function is maximum (the value 1) when the distance is 0, declining as the distance increases, although it never reaches zero, allowing to always make a prediction. The standard deviation, σ, parameter can take various values each one slightly modifying the shape of the Gaussian function.

A new heuristic will be used to calculate the optimal value of the standard deviation. This new heuristic relies on the *k* parameter, averaging the distance from all the worse nearest neighbors from the test set. It has the advantage of being dynamically adjusted according to the data. The heuristic is given by the expression:

$$\boldsymbol{\sigma} = \sqrt{\frac{\sum \boldsymbol{distance(i_u, k)}}{\boldsymbol{N}}}, \boldsymbol{N = number\ of\ elements\ in\ test\ set}$$

The $distance(i_u, k)$ corresponds to the distance of the worst nearest neighbor for a given i_u image from the test set.

3.3.4 Parameter estimation by cross validation

When estimating the parameters one needs to avoid overfitting situations where results present high variance and low bias. Underfitting is the other extreme of parameter estimation where results present low variance and high bias. To determine the optimal set of parameters in our framework we shall apply cross-validation. Cross-validation is a model selection technique where data is randomly divided into training and validation sets and several trials are executed to find the ideal parameter values. This will allow assessing the algorithm's annotation error on the validation set for different parameter values. Several trials are repeated for different parameter values and different data splits.

3.4 Tag-based image annotation

3.4.1 Experiment protocol and data

In this section we have performed a series of experiments to establish a baseline *k*-NN annotation algorithm. Tests were performed using all annotations (24 concepts) from the MIR-Flickr. We have two types of experiment protocols:

- Cross-validation protocol: used to determine the best parameter/similarity score as in the case of choosing the similarity score and the best standard deviation parameter for neighbor weighting. For our cross-validation tests we randomly select a group of 10,000 images from the MIR-Flickr dataset, from which we will use 10% as the test data. This process is repeated 10 times to avoid bias. Only textual features were used in our cross-validation tests.
- Annotation protocol: used to execute the automated annotation *k*-NN algorithm. For the annotation protocol we have chosen 20,000 images and their features (which vary in each section) from the image dataset from which 5% will be used as test data and the remainder as training set.

3.4.2 Similarity Scores

We compared the performance of different similarity scores on a baseline implementation of the *k*-NN algorithm using the cross-validation protocol on textual features. Test results were assessed using the root mean square error for the three distances with a varying number of neighbors (the *k* parameter). In Figure 14 we can see the results for the various similarity scores using the varying *k* parameter.

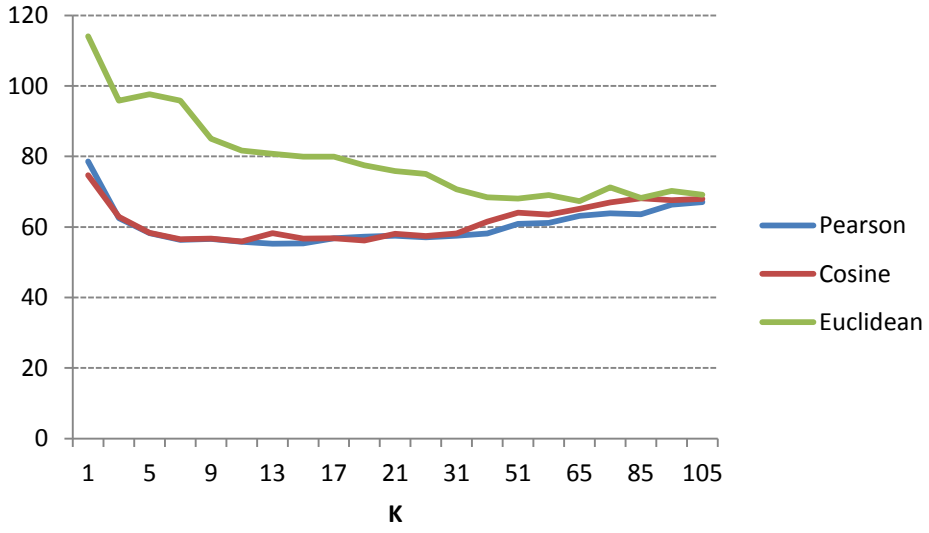

Figure 14 - RMSE from the various similarity scores with a varying *k*.

A closer analysis with $k = 3$ is detailed in the table below:

Distance	Precision	Recall	Accuracy	RMSE
Euclidean	27.96%	22.59%	78.82%	95.80
Pearson	55.23%	35.53%	85.43%	62.49
Cosine	54.12%	36.48%	85.25%	62.89

Table 3 - *k*-NN annotation algorithm with *k*=3.

The conclusions from the cross-validation are two-fold:

1. Experiments showed that the optimal number of neighbors is in the interval $k \in [7,15]$ for the Pearson and Cosine similarity scores. Another conclusion that can be drawn from Figure 14 is that as *k* increases all similarity scores tend to converge. This can be explained by the occurrence of an increasingly higher number of *k* neighbors which decreases the number of relevant neighbors, thus over-generalizing the annotation algorithm.
2. Second, as Figure 14 shows, the Pearson and Cosine distances obtain significantly better results when compared to the Euclidean distance. Based on the presented results we have chosen the **Pearson** correlation distance for most of the work developed throughout this thesis.

3.4.3 Term weighing

The figure below shows tests made with the baseline *k*-NN implementation with the Pearson similarity score versus the *k*-NN baseline implementation with the Cosine and *tf-idf* addition. The tests were conducted with the annotation protocol described previously using only textual features.

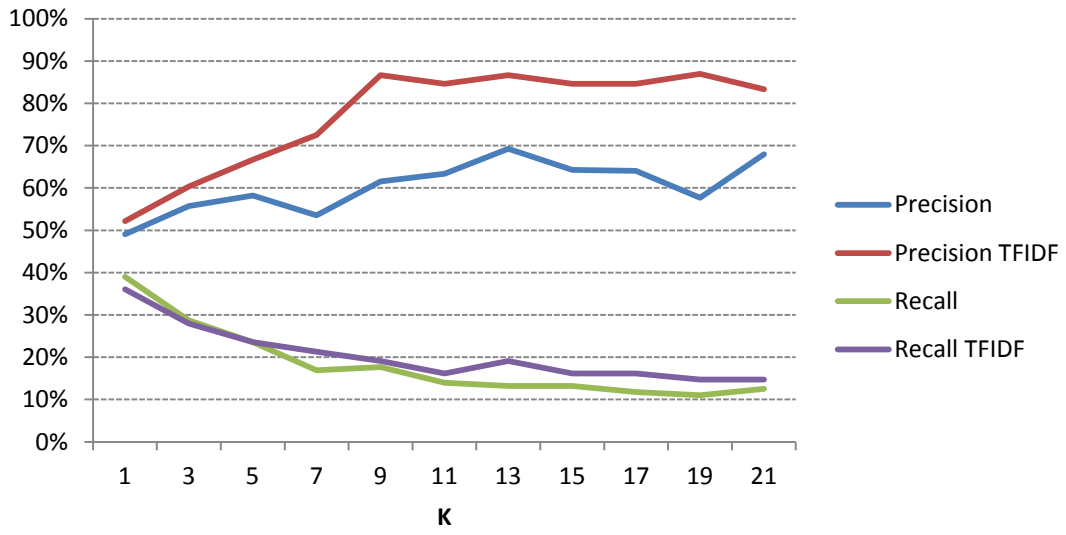

Figure 15 - Baseline Pearson implementation versus Cosine *tf-idf*.

From the analysis of the chart we can conclude that the usage of the Cosine similarity score with *tf-idf* yields better results. A major leap in performance occurs in precision while recall only improves marginally over the baseline implementation. We can infer the validity of this improvement.

Further study could be conducted to assess the performance with each individual annotation.

3.4.4 Neighbors weighting

To validate the correctness of this heuristic a simpler cross-validation was made to obtain the optimal standard deviation (sigma parameter) for a fixed dataset which closely matches the values obtained by the new heuristic. For this test we used the cross-validation protocol described previously with textual features and the Pearson correlation as similarity score. Figure 16 depicts this validation where the sigma obtained by the heuristic is compared with other sigma values throughout a varying *k*.

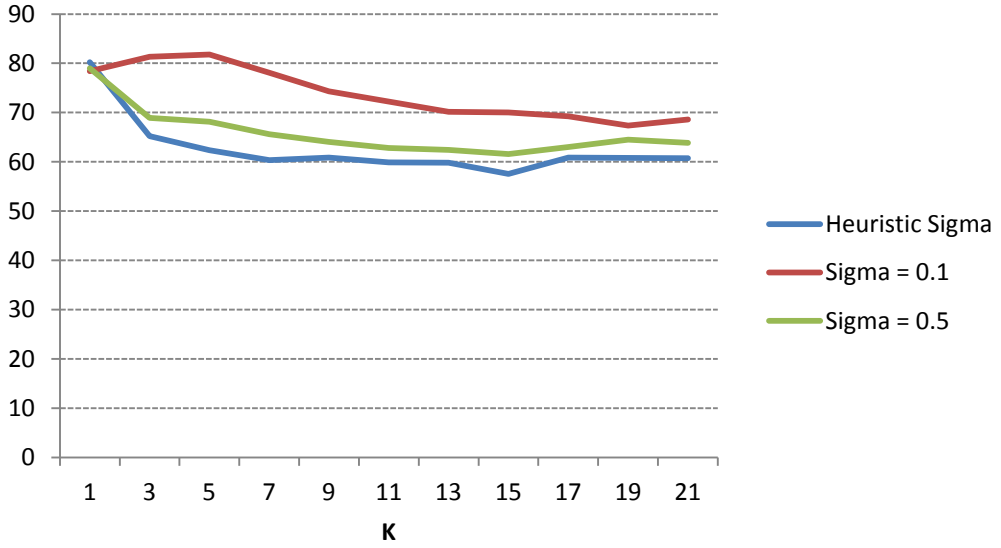

Figure 16 - RMSE with sigma comparison.

We can infer from the chart that the dynamic sigma obtained by the heuristic performs better than a fixed value for sigma, therefore this heuristic is valid and can yield good results. After estimating the sigma parameter further experimentation was made to explore the results of the weighted nearest neighbors' implementation versus the baseline using the annotation protocol (with textual features Pearson correlation). This comparison is depicted in the figure below:

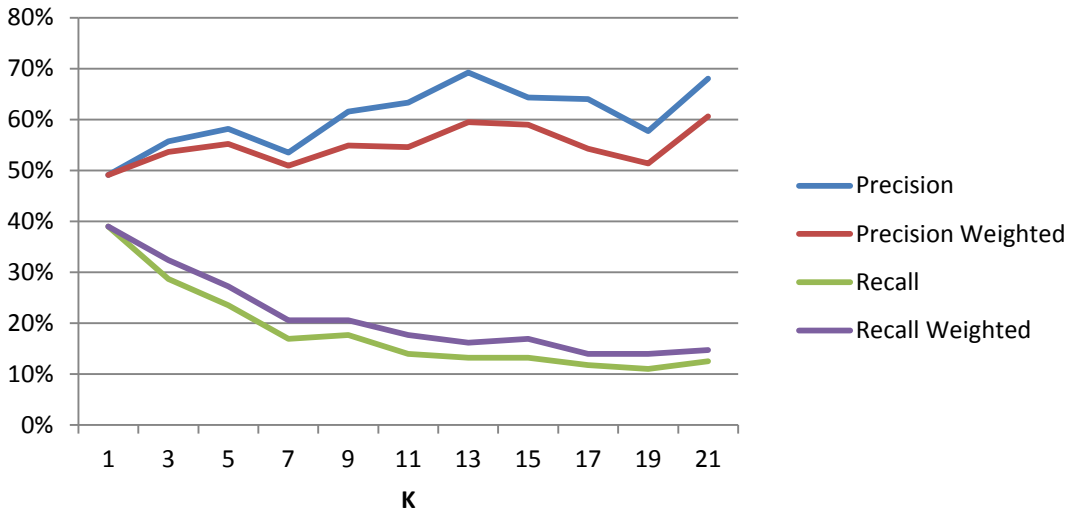

Figure 17 - Comparison baseline versus weighted *k*-NN.

From this chart we can conclude that precision is better in the baseline implementation when compared to the weighted implementation. Regarding recall, the weighted algorithm results in a slightly better recall mitigating the low recall usually associated with human annotation. In theory the weighted neighbor improvement should yield better results, across all metrics, because it solves the problem of attributing the same weight to neighbors that can be at different distances. In practice, and given the amount of noise in our image dataset, it is normal to expect lower precision rates as the weight importance of the nearer neighbors is augmented which could lead to more false positives with noisy data.

3.4.5 Discussion

As seen in the previous tests, the improvements made to the baseline *k*-NN implementation can contribute to a higher performance annotation algorithm and solve some of the problems the naïve version of *k*-NN has. We offer a comparison of these implementations below using the results previously obtained:

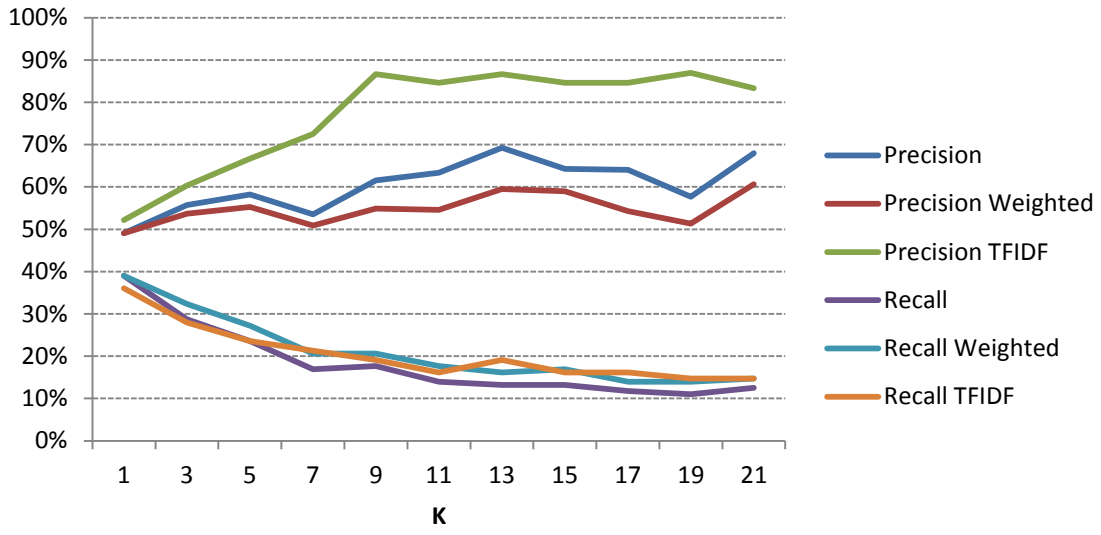

Figure 18 - *k*-NN comparison of improvements.

In regards to precision we can clearly see an advantage in using the term frequency over the remainder alternatives. In terms of recall the difference between the implementations is less distinct despite showing the baseline implementation having the worst recall. Further testing might find a mixture of the improvements yielding better results than the individual parameter tuning.

3.5 Visual-based image annotation

3.5.1 HSV Color moments

In this section we explored a *k*-NN implementation using the marginal HSV visual feature where our goal was to establish a baseline result for this feature descriptor. We performed this experiment using the annotation protocol with the baseline *k*-NN implementation and the Pearson correlation as the similarity score. For further comparison we added a *k*-NN implementation that randomly annotates images. Below are the results for this experiment:

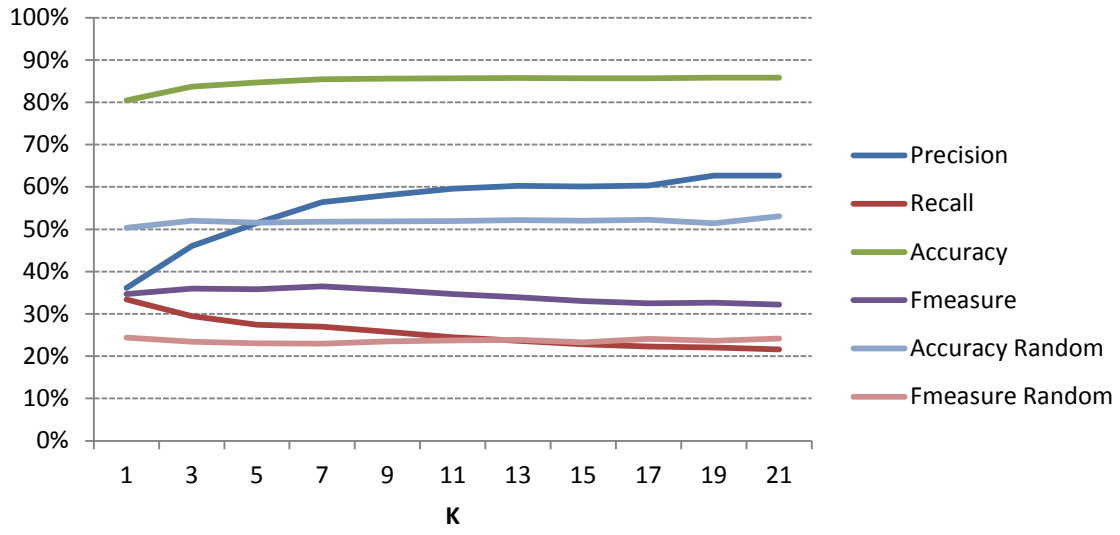

Figure 19 - Baseline *k*-NN using Marginal HSV features.

An analysis of this chart shows a slight increase in performance from *k*=1 to *k*=7 in the F-measure, which can be explained by a rapid increase in precision. The explanation is found in a decreasing number of false positives with the growth of the *k* parameter. With this increase in the number of neighbors more information is inserted in the *k*-NN approximation function which eliminates bias that comes from performing annotation with a small pool of neighbors. From this chart we can also assess that recall steadily decreases in performance proportionally to the increase of the *k* parameter. This decline corresponds to the increase in the number of false negatives and can also be explained by the same reason precision increases. The decrease in bias that comes from the increase in the number of neighbors, over-generalizes the approximation function lowering true positives/increasing false negatives. We can also show that this feature descriptor is useful in *k*-NN for annotation purposes as its performance is better when compared to random annotation.

3.5.2 Tamura features

In this section we explored a k-NN implementation using the Tamura feature descriptor where our goal was to establish a baseline result for this feature descriptor. We performed this experiment using the annotation protocol with the baseline *k*-NN implementation and the Pearson correlation as the similarity score. For further comparison we added a *k*-NN implementation that randomly annotates images. Below are the results for this experiment:

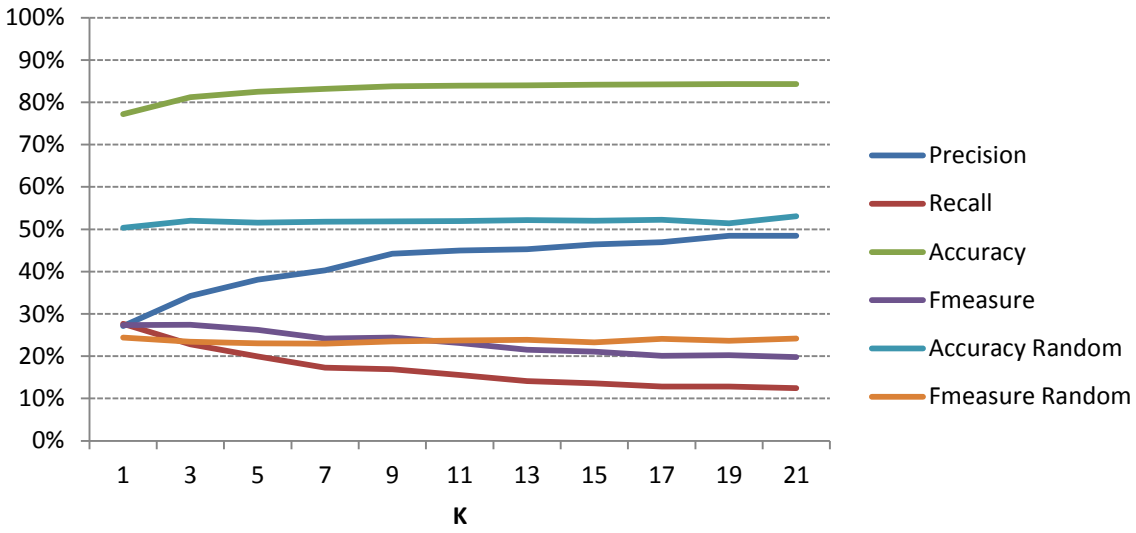

Figure 20 - Baseline *k*-NN using Tamura features.

An analysis of this chart shows a worse overall performance when compared with the marginal HSV color moments. There is a constant decline in the F-measure, as with the marginal HSV feature descriptor, explained by the same reason with the increase in false negatives accompanied by the growth of the *k* parameter. This increase in the number of neighbors over-generalizes the approximation function discarding most fringe cases. Precision increases as a side effect to this generalization by reducing the number of false positives, but with the cost of also lowering the number of true positives. When compared with random annotation despite having a greater accuracy, the F-Measure is, for most of the time, lower than the random annotation. For this reason and when in comparison with the marginal HSV color moments we can conclude this feature descriptor isn't as useful as the marginal HSV color moments and could be discarded.

3.5.3 Gabor filter moments

In this section we tested the Gabor visual feature in the *k*-NN implementation where our goal was to establish a baseline result for this feature descriptor. We performed this experiment using the annotation protocol with the baseline *k*-NN implementation and the Pearson correlation as the similarity score. As referenced we represented the image using two feature spaces, the visual feature descriptor (Gabor) and the image annotations. For further comparison we added a *k*-NN implementation that randomly annotates images. Below are the results for this experiment:

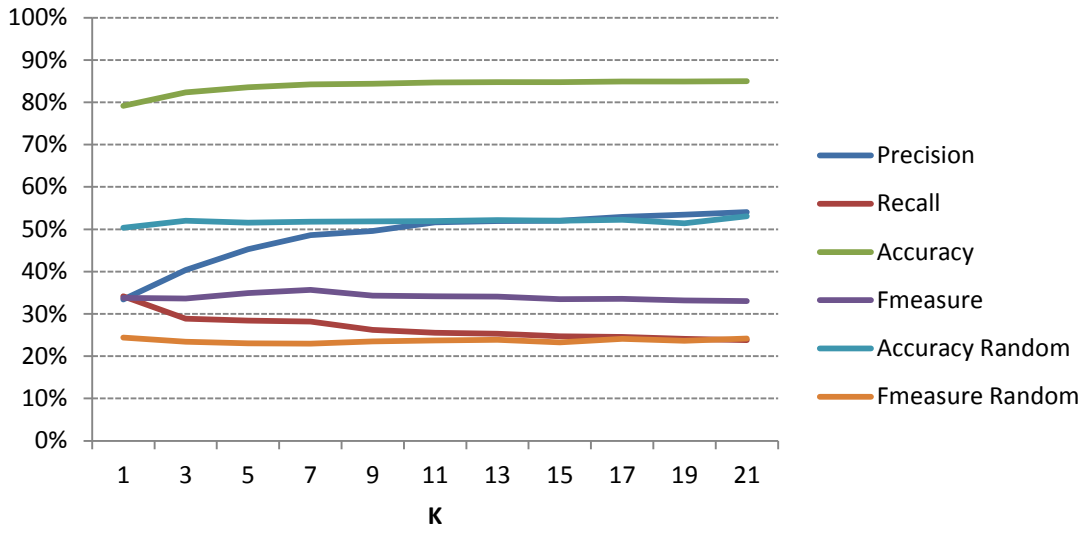

Figure 21 - Baseline k-NN using Gabor features.

As with the marginal HSV feature space, analysis of this chart also shows a slight increase in performance from k=1 to k=7 in the F-measure, which can be explained by a rapid increase in precision. The explanation is the same as in the previous feature descriptors, where the increase in the k parameter generalizes the approximation function. From this chart we can also observe that recall steadily decreases in performance, albeit at a slower rate than other feature spaces, proportionally to the increase of the k parameter. When compared to random annotation the k-NN algorithm using Gabor features clearly outperforms random annotation. Therefore we can conclude there is useful information to be added to automated annotation using Gabor features. Finally, by comparison with the other features, the Gabor filter moments are clearly superior to the Tamura features and on par with the marginal HSV color moments.

3.5.4 Per-Annotation analysis

In this section we analyzed annotation performance for each k-NN implementation and assessed the results. This analysis was executed over the previous k-NN implementations for the three visual feature descriptors, taking into account the 24 concepts that compose the annotation vocabulary used to describe an image. The k parameter, number of neighbors, was chosen according to the k with the best performance in the interval from $k \in [1,21]$. The F-measure metric, which combines precision and recall, was used to compare the different implementations.

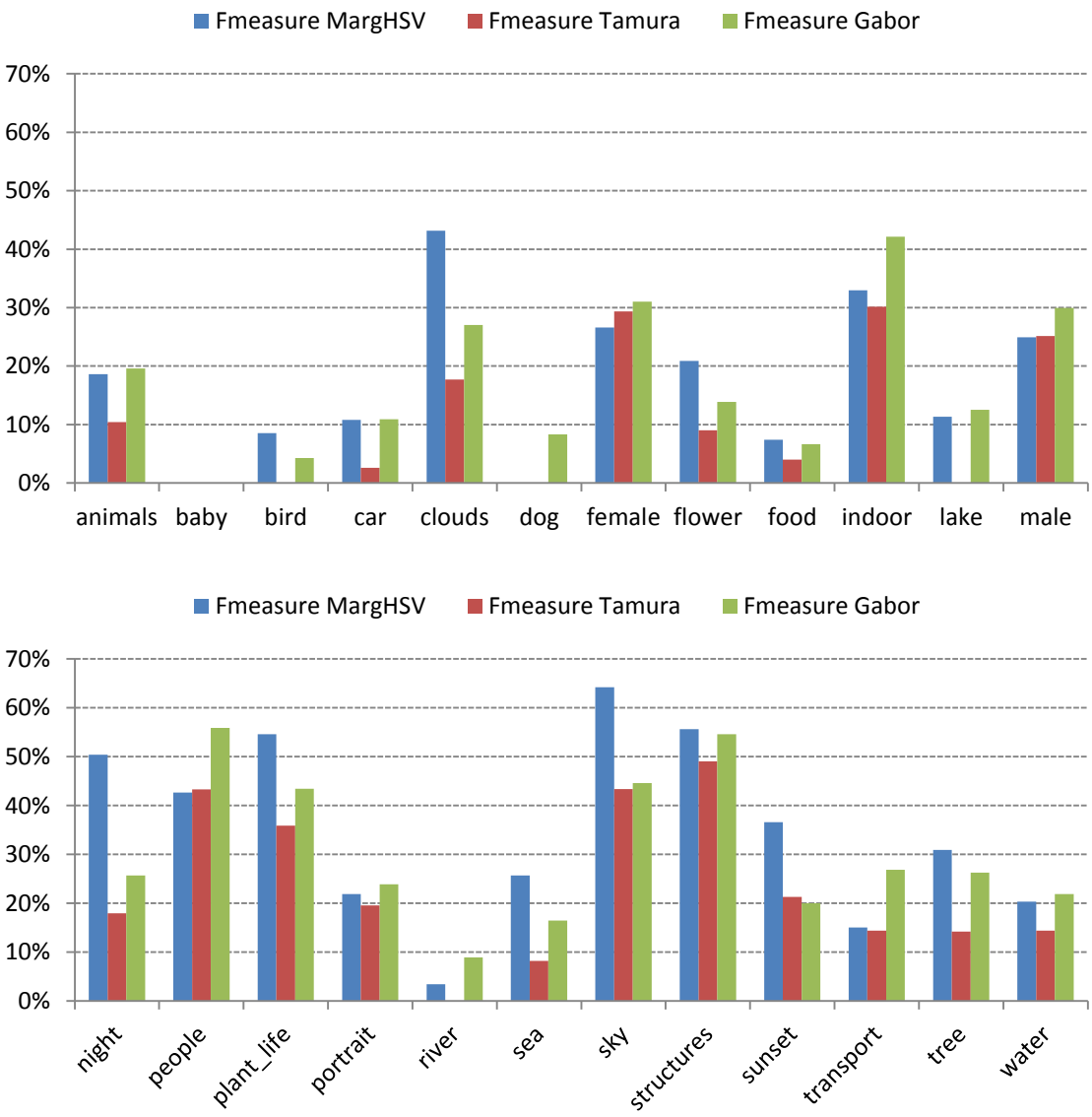

Figure 22 - *k*-NN Annotation comparison for the various feature descriptors.

The annotations *baby, bird, car, dog, food, lake* and *river* have a very low performance in terms of precision and recall (which translates to a lower F-measure) in every feature descriptor implementation. For some of these annotations, there is a low number of images in the dataset that contains the annotation, such is the case of baby, bird, dog or food. Other explanation for the poor results is the close similarity between concepts like river, sea, water, lake or animal, bird, dog. Within these groups of concepts there is a possible annotation bias towards a concept with more annotated samples in the dataset therefore with a higher possibility of having better similar neighbors. For some of these concepts, the visual feature descriptors don't help much: in concepts like *baby, car* or *food* an image can take many forms resulting in no explicit visual pattern or visual patterns too complex to be detected.

Another conclusion comes from the *k*-NN implementation for the marginal HSV feature descriptor where annotations like *clouds, night, sky, plant_life, sky* and *structures* obtained good performance when compared to the other visual feature descriptors. This is an expected conclusion as the marginal HSV feature descriptor deals with the color information in the images and the concepts corresponding to the annotations mentioned above are all characterized by distinctive colors (i.e. blue for clouds and sky, green for plant_life). There are also benefits from most of these annotations having a significant amount of training images which can yield a better approximation function.

An analysis on the Tamura features results, finds the best performing annotations to be *female, male, people, portrait*. This feature descriptor is based on the human visual perception of texture, from which the above annotations have much in common as the texture on human faces and human body is very similar throughout the image dataset.

To conclude, the Gabor *k*-NN implementation excels in annotations like *indoor, female, male, people, structures*. This feature descriptor is the application of the Gabor filter for edge detection, from which orientation and scale can be extracted in the form of textures to be compared. The annotations in which the Gabor implementation excels all have distinctive patterns, for instance straight lines for the *indoor* and *structures* annotation, and the same type of line patterns for *people*, *female* and *male*.

Through the study of each feature descriptor *k*-NN implementation and subsequent annotation analysis we could infer which descriptor performs best for each annotation and corresponding concept. Furthermore almost every feature descriptor gives some contribution for the annotation algorithm, provided there is a significant amount of training images.

3.6 Summary

The *k*-NN is an apparently simple algorithm that can be mutated according to the needs of the problem domain. In this chapter we studied several improvements over a baseline implementation of the *k*-NN that can solve some of its flaws. Three studies were conducted in different settings, namely similarity scores, neighbor weighting and usage of term frequency-inverse document frequency. First we established the Pearson correlation as the similarity score used for the baseline implementation. We also studied improvements over the baseline algorithm where the weighted neighbors achieved a marginal improvement over the baseline implementation. The cosine similarity score – *tf-idf*

combination allowed also a slight increase in performance over the previous implementations but is limited to textual features.

To conclude we studied visual features using the baseline implementation. This study has shown visual features can hold relevant information to automated annotation systems and in particular the three feature descriptors studied. From each feature descriptor we could evaluate their strengths and weaknesses and how they can be used to future improvements on automated annotation, namely better performance of Marginal HSV color moments and Gabor versus the Tamura features.

4

User tags model

4.1 Introduction

The act of tagging images, also called annotating, is a way of succinctly describing the content of images. The tags of an image describe how the user perceives that image– each tag is a linguistic concretization (possibly with typos) of an abstract concept of that particular user. Social media users, tag content with every keyword that they wish. This generates uncontrolled vocabularies nowadays called folksonomies. Their advantages are obvious from the multitude of social-media Web applications that apply it successfully. Marlow et al. [23] proposed a taxonomy to help in the analysis, design and evaluation of these applications, hence, confirming the variety of Web 2.0 applications. The direct application of uncontrolled vocabularies offer a good solution to the problem of multimedia annotation but it is not a solution that delivers full accuracy and completeness. Thus, understanding how users annotate as a whole becomes a critical task to exploit the full potential of uncontrolled vocabularies, [24].

Inspecting the tags of the image in Figure 23, one can easily spot some correct tags, incorrect tags and in some cases mutually exclusive tags. Abbreviations, words out of context or word concatenation are common in tags and provide rich and sometimes ambiguous information. Later, if users wish to find those images or related images, they can submit a query with the corresponding tags. As such tag accuracies are crucial to enable better information retrieval systems.

Blue, nyc, paris, park, people, portrait, rome, sea, sky, street, tree, urban, vacation, sunset, boston, bw, live, nature, kids, home, color, baby, beach, yellow…
http://www.flickr.com/photos/escariao/2187632618/

Figure 23 - Flickr image and its corresponding tags.

The community-control nature of folksonomies provides an adequate setting for the creation of new terms and jargon specific to that community of users. This leads to situations where only the users belonging to that community are fluent in the new jargon. Such situation is commonly called idiosyncrasies in information retrieval: group of users might give different meanings to common words or might create new words.

Idiosyncrasies pose many challenges to an information retrieval system when searching content tagged by users. First, it is never possible to know the correct meaning that a user gives to a keyword, e.g., the keyword football means different sports for different cultures. Second, the user might dishonestly annotate a document with a popular keyword to attract other users (spam). Third, users might have different criteria to annotate documents, e.g., some users might rigorously annotate all keywords while others might skip the obvious ones. Information retrieval systems have been dealing with idiosyncrasies in user queries for many years. Spellchecking, word synonyms, and word co-occurrences are the basis of many techniques to tackle this issue. Queries and tags are both created by users, meaning they share many characteristics and solutions. In particular, the usage of *WordNet* not only reduces noise in user queries, but also finds use in query expansion techniques. Following a similar reasoning, Sigurbjornsson et al [9] have also studied the mapping of Flickr tags onto *WordNet* semantic categories.

Social media tagging produces a vast amount of weakly annotated data. These human relevance judgments are a tremendously valuable data resource. Many image annotation and retrieval algorithms explore these tags as another form of input information about images. These algorithms incorporate this extra information source by simply counting tag occurrences. We propose to go one step further and propose to quantify the predicted level of accuracy of a tag.

In this chapter we propose to quantify the "weak" in "weakly-annotated data". We propose a user tagging model to quantify the accuracy of tags and groups of tags. Such

model has applications in automated annotation tasks, retrieval tasks and related problems. This work has links to tag ranking, where for a given image (or media) and its tags, an algorithm analyses the content and ranks its tags by importance. Our proposal is to model the tag accuracy over a set of data and predict the likelihood of that tag being correct in new data. For example, a given tag might have ambiguous meanings (e.g. face, row) or be popular among spammers (e.g. baby, girl). Thus, instead of using a binary annotation for representing the presence of a tag in an image, an image annotation algorithm can use probabilistic annotations linked to the confidence of a tag correctness.

To compute the user tag model we devised a framework to process tags with linguistic normalization and tag expansion techniques. The aim of the framework is two-fold: (1) improve tags accuracy by tackling tag ambiguity and expanding tags with related words; (2) enable a quantitative analysis of tags accuracy. The quantitative analysis of tags accuracy is then performed by comparing the user tags to ground-truth. This leads to values of precision and recall for each tag as well as for groups of tags. Ground-truth is available for two image datasets MIR-Flickr [4] and NUSWIDE [5]. To increase tags accuracy, the framework not only corrects typos in tags but also enhances them with tags related to the ones inserted by users.

The tag noise reduction process uses both linguistic and statistical techniques to clean user typos and idiosyncrasies. Noise reduction can be accomplished using linguistic corrections using tools such as Hunspell's spellchecking and WordNet lemma's to identify potential noise tags. This will provide expansion of the user tags based on its semantic and linguistic representation. Statistics based techniques will also be applied by creating clusters with the data from the user tag model (this corresponds to a tag co-occurrence model). For each tag only the top k results from its cluster, excluding the tag itself will be used in the user tag model expansion.

The work related to this chapter was presented in section 4.1. In the following section we offer an general description of the framework. Section 4.3 discusses the accuracy of tags without any cleaning process (raw tags). Section 4.4 and 4.5 discusses the linguistic and statistical correction techniques respectively. Finally, in section 4.6 we discuss the overall evaluation and discuss the results.

4.2 User tags model

In social media, users tag content by searching the document for the presence of a concept and annotate it with the corresponding keyword if it is present. The most distinguishable

characteristic of user tags is their subjective nature. This introduces several ambiguities rooted on the user's understanding of the keywords and criterion to decide the keyword presence in the content. Noisy tags can arise from spelling mistakes, abbreviations, purposefully erroneous tags or name usage making these tags unique. These noisy tags account for the long tail effect in the tag distribution of public consumer photo datasets like the MIR-Flickr or NUS-WIDE.

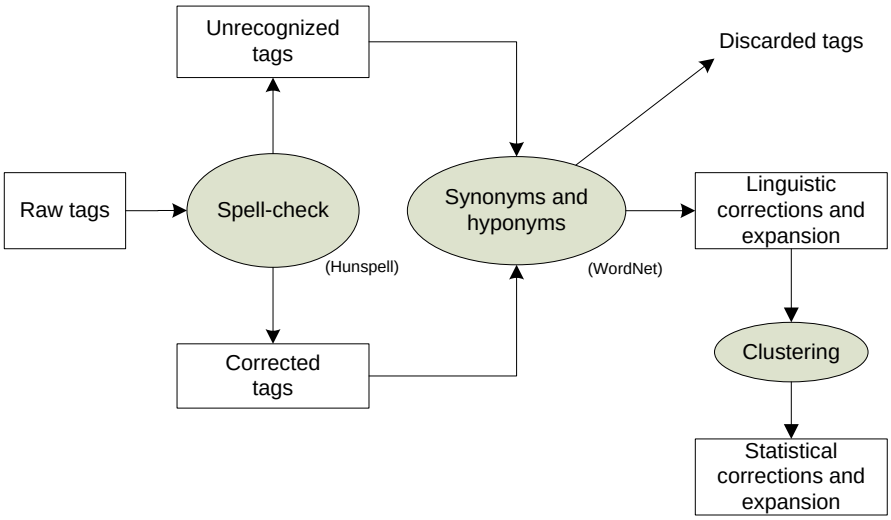

Figure 24 - User tag cleaning and enhancement framework.

To enhance user tags and answer the above challenges we propose a two-phase framework for linguistic normalization and expansion – the diagram in Figure 24 depicts the framework. The tags processing framework is divided into two parts:

- In the first phase a linguistic normalization is accomplished through spellchecking, stemming and keyword expansion using synonyms and hyponyms. This phase clears some of the noise in user tags, and expands tags with keywords similar to the ones assigned by the users.
- The second phase consists in a statistical expansion of the image tags by creating cluster of keywords based on existing tags, thus exploring tags co-occurrences.

Initially we evaluate the unprocessed tags and verify their accuracy, after that we apply the processing methods to clean and enhance the initial tags and verify the obtained accuracy.

4.2.1 Data

To study user photo tags we used the MIR-Flickr [4] dataset with 25,000 images and the NUS-Wide [5] dataset with 279,000 images. In terms of user tags, the NUS-WIDE dataset has 1,000 tags and the MIR-Flickr dataset has 69,000 tags. Both datasets were crawled from Flickr, a public image repository with a wide range of languages, resulting in extremely

heterogeneous user tags, e.g. different tagging languages for the same image and abbreviations. The NUS-WIDE dataset has ground-truth for 81 annotations and the MIR-Flickr dataset has ground-truth for 24 annotations.

4.2.2 Tags and Annotations

The user tagging model is composed by two types of textual features, the annotations and the tags, which are both keywords used to characterize an image. In both Flickr datasets (NUSWIDE and MIR-Flickr) we have ground-truth annotations and user tags creating a large folksonomy. Given the set of *M* user tags $W_T = \{t_1, \dots, t_M\}$, and the set of M annotations $W_A = \{a_1, \dots, a_M\}$, media documents can be represented as a tag feature vector and an annotation feature vector as depicted below:

$$\boldsymbol{d_T} = (\boldsymbol{d_{T,1}}, \dots, \boldsymbol{d_{T,M}}) \in \{\mathbf{0},\mathbf{1}\}^{\boldsymbol{M}} \qquad \boldsymbol{d_A} = (\boldsymbol{d_{A,1}}, \dots, \boldsymbol{d_{A,N}}) \in \{\mathbf{0},\mathbf{1}\}^{\boldsymbol{N}}$$

The d_T vector represents the tag information of N tags from the vocabulary W_T, where each component $d_{t,i}$ indicates the presence of tag i in the document d. The same applies to the annotations vector.

Annotations are created by curators of media documents who received specific training on how to identify concepts in information, how to clarify ambiguities regarding the meaning of keywords, and have the sole intention of correctly annotating content. Also, in most cases, annotations are obtained by a redundant voting scheme intended to remove disagreement between annotators. Thus, it constitutes an extra method of checking the validity of data annotations. Formally, we define:

- Annotations are intended to describe media content and correspond to the ground-truth knowledge concerning the given media. These annotations are considered high-level features because they require a considerable level of knowledge and perception to understand the reality captured in an image.

Social media tags are the creation of Web users motivated by many different reasons []. Tags have a subjective truth-value which can generate erroneous or incorrect information that must be considered as noise. Formally, we define:

- User tags are Web user inserted keywords to describe an image. Since tags are also keywords, the same keyword can be used as a tag and as an annotation.

The key difference between these two types of keywords lies in the ground-truth value of annotations that do not exist for user tags. We validate the correctness of a tag by matching it into an annotation. A tag is considered to be correct if it can be mapped into an annotation through some of the tag processing techniques.

4.3 Raw tags

Most commercial image collections have annotations with 100% accuracy produced by curators. In contrast, the social media tags are done by any user interested in creating and publishing online content. In real scenarios with social media users one would expect to have tags with accuracies below 100%. To verify this assumption we examined two datasets (MIR-Flickr and NUS-WIDE) containing Flickr images and measured the accuracy of several annotations. As expected, we concluded that the error is not random.

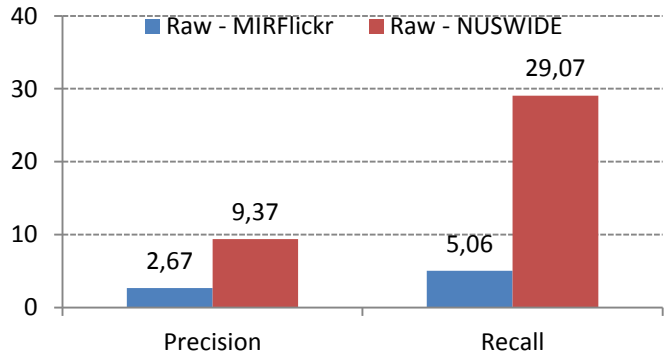

Figure 25 - Baseline model.

Figure 25 shows the precision and recall metrics on the MIR-Flickr and NUSWIDE datasets. Upon establishing the user tag model, it is necessary to establish a baseline model for all tags present in the datasets for further comparison. This model is an analysis of the errors in user tagging is made on both datasets when compared to the annotations supplied by professional annotators.

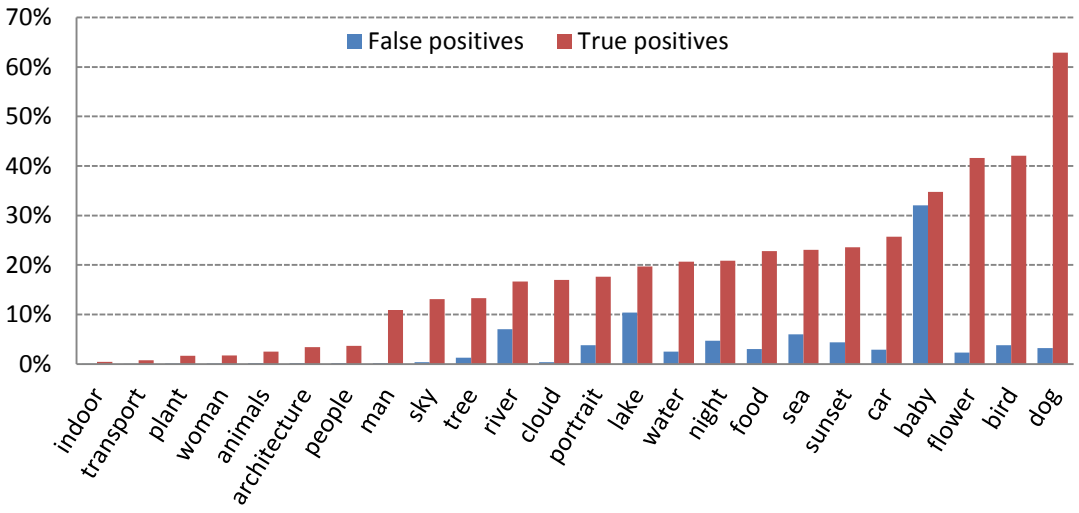

Figure 26 - Accuracy of real user manual annotations for the MIR-Flickr dataset.

On Figure 26 we can see the baseline model results using all the tags and annotations present in both MIR Flickr and NUSWide datasets. The difference between MIRFlickr and

the NUSWide datasets can be explained by the tags present in both of them. While the NUSWide has only a fixed set of 1,000 tags with a more controlled vocabulary, the MIRFlickr tag set is composed by about 70,000 tags. This tag set has a very high noise rate with many misspelled tags, foreign language terms or word concatenated tags, thus lower precision and recall rates.

4.4 Linguistic tag corrections and expansions

Our first approach to remove noisy tags will be through the analysis of tag frequencies across the entire dataset – tags occurring less than a fixed number of times are removed to ensure the non-uniqueness of the tag. This approach lacks the semantic validation of linguistic knowledge, which can only be solved using dictionaries [17]. Through the usage of a dictionary to check the tag as a valid word we can further improve our tag set by discarding tags that can't be found in a dictionary and don't occur prominently in our dataset.

4.4.1 Spellchecking

Spellchecking was implemented with the *Hunspell* spellchecker which performs morphological analysis to enable the cross-referencing between tags and real words. By spellchecking each tag and setting aside tags that don't belong to the English dictionary we began to clean the tag set and removed noisy keywords.

Its stemming capabilities can also increase the tag set by providing the root form of the word. By stemming tags we can increase the words matching rate and find better synonyms and hyponyms (described next).

4.4.2 Semantic similarities

For this step of the framework the *WordNet* [22] lexical database of English words was used. This database encompasses a collection of nouns, verbs, adjectives and adverbs grouped into sets of cognitive synonyms and hyponyms, each expressing a distinct concept. Words are interlinked through conceptual and lexical relations. The resulting network of meaningfully related words and concepts can be accessed to discover similarities. In this framework semantic similarities will be explored are as follows:

- *Synonyms*: different words for the same meaning
- *Hyponyms/Hypernyms*: define the "is-a" relation. A common example is the hyponym *dog* whose semantics is included in the hypernym *animal* establishing the hierarchical relation between them.

- *Meronyms/Holonymy*: these two semantic similarities define the "part-of" relation, the hierarchical inferior concept must always be a part of the hierarchical superior concept. A common example is the meronym *finger* and the relation to the holonym *hand*.

Although the size of this database is considerable, it becomes limited when comparing to the concept space of the World Wide Web. Nevertheless this tool has been proven very useful [25, 26] to successfully obtain a better performance in the automated annotation domain. It is also proven in [9] that at least 51.8% of Flickr tags can be mapped onto WordNet semantic categories.

4.5 Statistical tag corrections and expansions

In the presence of a wide set of tags it is common to find groups of keywords that occur together, e.g., the co-occurrence of the keywords 'sky' and 'clouds' or 'portrait' and 'woman'. Finding structure, or groups, within data is the subject of clustering and is used frequently in data-mining applications. Clustering is an example of unsupervised learning where the learning algorithm learns the structure of un-annotated data. These methods can enrich the set of media tags and can disambiguate some of their tags.

In the case of the social media tags, we can discover groups of highly frequent co-occurring tags. These groups not only allow finding co-occurring keywords but also synonyms, hypernyms and even the same keywords in different languages.

Dealing with tag ambiguity improves the quality of user tags. Word ambiguities is a challenge that has been tackled before through the statistical analysis of a word context Kilian et al [27], is an example of a probabilistic framework to find ambiguities and allowing the user to decide how to disambiguate. While semi-automatic methods are useful to solve ambiguities, an automated method is required in our case.

We applied the k-Means clustering algorithm to create the clusters of keywords. The discovery of groups can be used to indicate that keywords within a group have a higher probability of association between them. The statistical expansion method proposed relies on this probability. By finding the k nearest keywords from any given keyword in a cluster we can add these k keywords to enrich our user tag model. This method relies on the truthfulness of the data to create distinct clusters of tags. In general this phenomena is too rare to cause any bias: the majority of tags are correct and incomplete, but not incorrect (there are few false positives). Thus, the contribution of clustering will be

twofold: (1) it will provide new information relating a tag to other tags in the same cluster; (2) it will help disambiguate some of the keywords.

A common method to disambiguate is to suggest the most common co-occurring tags allowing the addition of new information. For example, given the tag "ski", another common tag is "snow". This method can sometimes add redundant information, which an example can be found in the tag "Eiffel" where the tag "Paris" would be the most common co-occurring tag but wouldn't add any new information, while a tag like "sunny" or "night" would be far more valuable.

4.6 Evaluation

An evaluation was conducted to assess the techniques presented in the previous sections. Different combinations of techniques were also evaluated: the following table summarizes the tested combinations of linguistic and statistical tag processing techniques. The statistical technique used k= 40 and only the three highest ranked tags were chosen from each cluster to enrich the user tag model for each tag.

Spell-check	Stemming	Synonyms	Hyponyms	Clustering
-	-	-	-	-
✓	-	-	-	-
-	✓	-	-	-
-	-	✓	-	-
-	-	-	✓	-
✓	✓	✓	✓	-
-	-	-	-	✓
-	✓	✓	✓	✓
✓	✓	✓	✓	✓

Table 4 - Summary of test conditions.

4.6.1 Results

Table 4 presents the test results for the different combination of techniques (Figure 27 and Figure 28 illustrate the same results graphically). The usage of Hunspell [28] to correct spelling mistakes greatly reduces the number of false positives improving precision in image tags. In conjunction with stemming, false positives can also be further reduced. Stemming alone increased the matching rate among words which is the reason behind a slight increase in both recall and precision. Spellchecking and stemming did not improve

results significantly in the NUSWIDE dataset, which might be linked to a crawling strategy based in query words to find data in Flickr.

	MIRFlickr		NUSWIDE	
	Prec	**Recall**	**Prec**	**Recall**
Raw	2.67	5.06	9.37	29.07
Spell-check	5.26	5.06	9.51	29.07
Stemming	3.04	5.76	9.81	30.21
Synonyms	2.88	5.46	9.98	30.70
Hyponyms	3.69	6.96	12.25	36.56
Clustering	5.85	10.76	18.06	49.55
Spell-check + Synonyms + Hyponyms + Stemming	8.31	7.92	13.27	38.66
Cluster + Synonyms + Hyponyms + Stemming	6.94	12.70	21.25	56.11
Cluster + Spell-check + Synonyms + Hyponyms + Stemming	11.97	11.24	21.53	56.06

Table 5 - Test results for both datasets.

A small improvement in precision can be obtained by adding tag synonyms to the tagged images. However, hyponyms are the cause of the best single linguistic improvement that can be made to raw tags in both datasets. Finally, a combination of all linguistic techniques is significantly better than any other single technique.

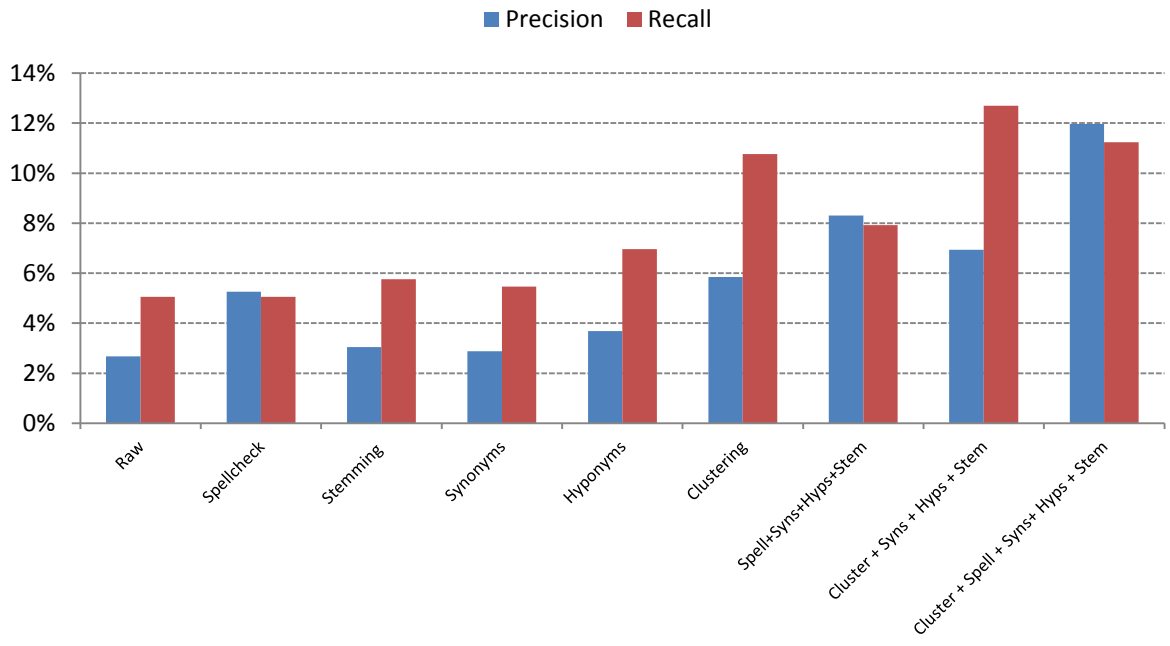

Figure 27 - Results for the MIR-Flickr dataset.

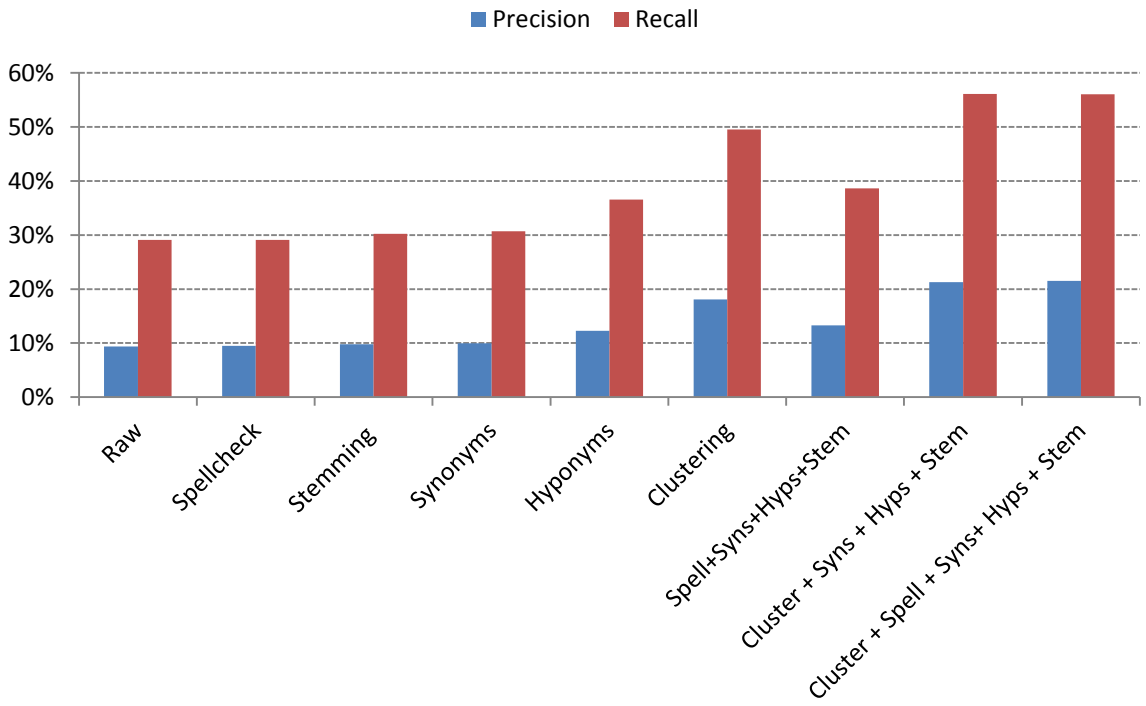

Figure 28 - Results for and NUSWIDE dataset.

When statistical expansion is introduced results change significantly. The introduction of statistically correlated tags can improve more than the usage of synonyms and hyponyms. This is in line with previous knowledge in information retrieval: words from the same words cluster (or with other type of statistical correlation) have strong linguistic associations and in some cases can be used to disambiguate other words.

The combination of linguistic and statistical methods (spellchecking to reduce false positives, and expansion through clustering, synonyms and hypernyms for true positives) achieved the best results by far. In the NUSWIDE dataset precision improved 129% over raw tags and recall improved 92% over raw tags. In the MIRFlickr dataset, improvements over raw tags are even more noticeable: precision increased by 348% and recall increased 120% (note that these figures include spell-checking and stemming which are standard IR techniques).

4.6.2 Discussions

The first methods use linguistic correction and expansion which in itself significantly improve tags precision and recall. While in general linguistic techniques are particularly relevant, in some cases there might be a slight reduction in precision is due to the insertion of ambiguity through synonymous, hyponyms (categories), polysemes (same spelling, related meanings) and homonyms (same spelling, different meanings).

The second method uses statistical expansion discovering co-occurring tags, which not only solves the tag ambiguity problem but can also provide new information about the tagged images. Consequently, we found out that statistical techniques are more robust than linguistic techniques – linguistic techniques look at isolated tags and statistical techniques look at the context where the tag occurs (the co-occurring tags).

By combining both type of techniques and since the performance of each one doesn't affect the other, the improvements made on the tags accuracy are quite relevant. Results showed how this framework can improve precision by adding new information, i.e., expanding the user tag set. This message is quite relevant as it summarizes how the accuracy of social media tags can become extremely useful for higher quality user tag models.

4.7 Conclusions

This chapter illustrated how a significant improvement in tags accuracies can be achieved, therefore increasing the value of social media tags. Tags have a low recall and precision which can be significantly enhanced with the detailed framework to complete and correct existing tags. Other improvements can be made to this framework by using a word segmentation algorithm that can account for hidden words in some of the tags (e.g. *newyorkcentralpark*). Another improvement can be made by implementing the disambiguation detection methods akin to Weinberger et al [27].

Finally, we believe the proposed framework can provide a richer and higher quality tag set enabling better automated media annotation algorithms.

5

Knowledge-based image annotation

5.1 Introduction

The title knowledge-based image annotation is motivated by the information learnt in Chapter 4 where we developed a framework for linguistic correction and expansion of the user tag model. It was confirmed that the developed framework provides higher quality data which can now be used in automated annotation algorithms. In this chapter we test this hypothesis by adding the cleaning and expansion framework to image annotation algorithms, such as the *k*-NN implementation discussed in Chapter 3. Other contribution from this chapter stems from the development of a methodology to use a multi-feature space in annotation algorithms. This methodology is an evolution from the single feature (visual or textual) annotation algorithms studied in Chapter 3. Furthermore, we answer the deficiencies in *k*-NN annotation algorithms by developing an iterative annotation algorithm based on the Learning with Local and Global Consistency algorithm [19]. The new algorithm, while having root in *k*-NN, can suppress its deficiencies achieving better performance. Through the combination of learnt knowledge about user tag model and *k*-NN improvements, we propose a multi-feature annotation algorithm that can explore a large number of un-annotated data present in most real-world image annotation problems. The outline of this chapter is detailed in the following figure:

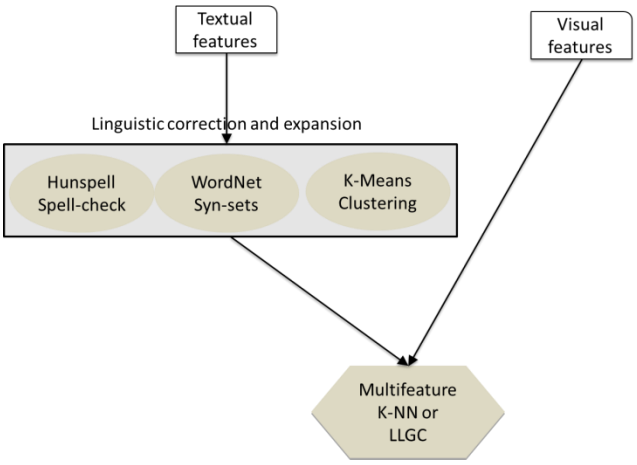

Figure 29 - Knowledge based multi-feature algorithms.

First, we explore the knowledge enhancements to the user tag model developed in Chapter 4. The following section, will explore the different approaches to multi-feature (feature-fusion) with *k*-NN algorithms and their variants. Section 3 will present a new algorithm, evolved from the *k*-NN algorithm and based in [19]. These first three sections conclude the theoretical presentation of the algorithms. Section 4 presents the experiments conducted to evaluate the proposed framework. Finally, we summarize the most important observations and contributions of this chapter.

5.2 Knowledge and feature fusion

5.2.1 Knowledge sources

In Chapter 4 we analyzed textual features from images datasets. From this analysis we concluded that not only a significant amount of noise can be reduced, but that also some of the data from textual features can be enhanced. We studied tag relevance and created a framework to correct and expand image tags. The framework is divided in three phases:

- **Linguistic correction**: where spellchecking on the textual features removed noisy words. The first phase has two components:
    - *Hunspell* is the primary tool for linguistic correction, and consists in simple spellchecking from which we can assert the existence of a certain keyword/tag in a dictionary.
    - *WordNet* is used as a backup tool for linguistic correction since it includes some keywords not found by *Hunspell* such as acronyms or other abbreviations.
- **Linguistic expansion**: the second phase of the framework expands the existing image tags. After the linguistic correction phase, the framework uses the cleaned

version of the keywords to search for their synonyms, hyponyms and stems to be added as additional information. The main tool used in the phase is *WordNet*.

- **Statistical expansion**: the last phase of this framework also expands the dataset although using a clustering strategy. We group co-occurring keywords into clusters using the *k*-means clustering algorithm, and use the most relevant keywords from each cluster to be added to the textual features for each image.

In Chapter 4 we concluded that the usage of this framework can yield a significant quality improvement on image tags and as such we will apply it to our annotation algorithm. Further testing will be made in the Evaluation section, applying this framework to image annotation algorithms. Further information on the framework can be found in Chapter 4.

5.2.2 Feature-fusion

After the tests conducted in Chapter 3, we can assert the utility of the various feature spaces studied in automated image annotation, namely the three visual descriptors and textual features. Despite their utility, each feature space has its flaws, i.e. lack of textual features or no correlation between visual features and textual features. These flaws can be mitigated if different feature spaces are used together, i.e. lack of textual features can be mitigated by visual features. A solution that combines feature spaces is an answer to these types of problems, and as such, we will present a methodology to combine feature spaces to enhance the annotation algorithm.

We will represent the image using a vector space model composed by two or more types of feature spaces, the visual feature descriptor (which corresponds to the Marginal HSV, Gabor and Tamura feature descriptors), the textual features and the image annotations:

$$\boldsymbol{d} = (\boldsymbol{d_V}, \boldsymbol{d_T}, \boldsymbol{d_A})$$

where a feature vector $\boldsymbol{d_T}$ describes the text part of the image, a feature vector $\boldsymbol{d_V}$ describes the visual part of the image, and the vector $\boldsymbol{d_A}$ containing annotation confidence scores. More specifically, we have:

- The feature vector $\boldsymbol{d_T}$ contains the tags added by the users. This tags will be the clean and expanded version obtained from the use of the framework developed in Chapter 4;
- The feature vector $\boldsymbol{d_V}$ contains low-level visual features such as texture, colour or shape;

- The feature vector d_A contains annotation confidence scores concerning the presence of the corresponding concept in that image;

With this new definition of the image representation we can combine the various feature spaces. This combination is called feature fusion where we will use multiple feature spaces to be used in the automated annotation algorithm instead of using just one feature space. Feature fusion will allow, as previously said, mitigation of the deficiencies in some of the feature spaces to increase the overall annotation algorithm performance. We will use two different variations of feature fusion to combine the nearest neighbors for each feature space:

- ***Avg K*** variation or linear average- by simple average of the total *k* neighbors per feature space i.e. in a test with all feature spaces and *k*=3 we will average 9 nearest neighbors, 3 per each feature space. This will allow information from each feature space to be inserted to the *k*-NN's approximation function.
- ***Top K*** variation or greedy selection- average of the best *k* neighbors from all feature descriptors i.e. in a test with all feature descriptors and *k*=3 we will select the top 3 nearest neighbors from the pool of 9 nearest neighbors. This variation will, optimally, select only the information from the best feature spaces, according to the similarity score, to be inserted to the *k*-NN's approximation function.

The two types of feature fusion presented will be the subject of an experimental evaluation in this chapter. The goal is to understand the best type of feature combination to be used in the annotation algorithm.

5.3 Local and global consistency

In Chapter 3 we analyzed the *k*-NN algorithm and its usage as an image annotation algorithm, and although some of its flaws can be solved by adding improvements there are still drawbacks. One of such drawbacks concerns the amount of annotated information available. The *k*-NN algorithm requires significant amount of information to have good performance which doesn't happen in most real world applications. In these scenarios un-annotated samples are far easier to obtain thus posing a problem for evolving annotation algorithms. The root of the problem lies in the *k*-NN structure as a supervised learning algorithm using only annotated training samples to infer its approximation function. The key to an evolving annotation algorithm is the shift from supervised learning to semi-supervised learning, which is learning not only from annotated data but also from un-

annotated data therefore requiring less data. By requiring less data the algorithm isn't susceptible to high variance in the presence of limited samples as in the *k*-NN algorithm.

Supervised learning algorithms, namely the *k*-NN, explore the local consistency in a set, using the neighboring points (images in our case) to infer the annotation. In semi-supervised algorithms, not only the assumption of local consistency is needed, but a new notion of global consistency (also called cluster assumption) is introduced. Both of the consistencies can be described as:

- Local consistency – where nearby images are likely to have the same annotation, which is a natural consequence of *k*-NN algorithms.
- Global consistency - where images on the same structure (typically referred to as a cluster or a manifold) are likely to have the same annotation.

We can understand the need for both types of consistencies by analyzing Figure 30, where we find the input data for an automated annotation task. In this figure we can see two annotations (red and blue) where the larger red and blue points are annotated data and the remainder points un-annotated data. Supervised algorithms, like the *k*-NN, base their annotation methodology on local consistency, that is to say on the closest neighbors and don't take into account global data distribution patterns. As we can see in Figure 30 there is a clear pattern in the data distribution as the blue and red annotations form a two moons pattern. This pattern can be found using global consistency. Figure 31 shows the connection graph between the various points in the data set and the output of the automated annotation task using the local and global consistency algorithm (*LLGC*) where the two moons pattern is taken into account in annotation.

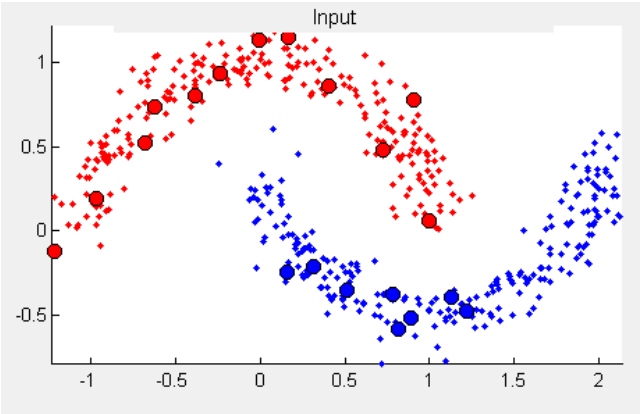

Figure 30 – Automated annotation on the two moons pattern.

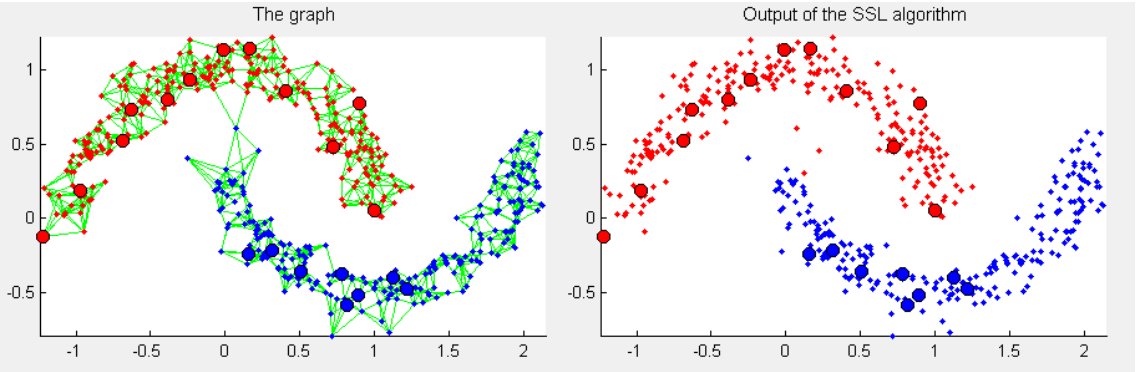

Figure 31 - Annotation using the LLGC algorithm (left) distances graph (right) automated annotation output.

In this new algorithm we are learning from annotated and un-annotated sources of information using an iterative process where the new information obtained from local and global consistency will be propagated throughout the dataset. The amount of propagated information from annotated and un-annotated data will have to be controlled by parameters previously estimated. The parameter estimation leads to the drawback of this algorithm, where an initial parameter tuning phase requiring a significant amount of time is needed to obtain optimal parameters. This algorithm is based on the work in [19].

5.3.1 Algorithm

Given a set of images $X = \{x_1, \dots, x_l, x_{l+1}, \dots, x_n\} \in R^m$ and an annotation set $W_A = (w_1, \dots, w_L)$, the first *l* images $x_i (i \leq l)$ are annotated as $y_i \in W_A$, i.e. the training set, and the remaining images $x_u = (l+1 \leq u \leq n)$ are un-annotated, i.e. the test set. Each image *i* is represented by a vector x_i corresponding to a given feature space. The goal is to predict the annotation of the un-annotated images, therefore performing image annotation.

The algorithm is as follows:

1. Form the affinity matrix W defined by:

$$\boldsymbol{W_{ij}} = \begin{cases} \exp\left(-\dfrac{\boldsymbol{sim.score}}{\boldsymbol{2\sigma^2}}\right), \boldsymbol{i \neq j} \\ \boldsymbol{0, i = j} \end{cases}$$

   where $W_{ii} = 0$ (to avoid self-reinforcement) and "*sim.score*" is a similarity measure (e.g. Manhattan distance or Pearson correlation) between the i and j feature vectors.

2. Construct the matrix

$$\boldsymbol{S = D^{-1/2} W D^{-1/2}}$$

where D is a diagonal matrix with its (i,i)-element equal to the sum of the *i*-th row of *W*. This step translates into a symmetric normalization of the W matrix.

3. Iterate

$$\boldsymbol{F(t+1) = \alpha SF(t) + (1-\alpha)Y}$$

until convergence. During each iteration of this step each image receives the information from its neighbors, the first term $\alpha SF(t)$, and also retains its initial information, the second term *Y*. The parameter $\alpha \in [0,1]$ specifies how much information is received from the neighbors versus the initial annotations. In the first step, the algorithm, equivalent to *F(0)*, is equal to the initial annotations *Y*.

4. Let *F** denote the limit of the sequence $\{F(t)\}$. Annotate each image x_i as a annotation

$$\boldsymbol{y_i = \arg max_{j \leq c} F^*_{ij}}$$

which means the annotation of each un-annotated image corresponds to the annotation from which it has received the most information during the iteration process. Since we are performing binary annotation, the possible outcomes are the presence or absence of the new annotation.

5.4 Evaluation

5.4.1 Data and experiment protocol

For our experiments we chose 20,000 images from the MIR-Flickr image dataset to be used in our analysis from which we chose varied distributions of training and test images. We performed automated annotation for all the 24 concepts in the MIR-Flickr datasets collecting the precision, recall, accuracy and f-measure metrics.

5.4.2 Experiment 1: Raw tags versus Expanded tags

The previous chapter dealt with an analysis of the user tag model, which allowed making linguistic corrections and expansions to the textual features. We can now confirm if the conclusions from the previous analysis are extensible to performance improvements in annotation algorithms. In this experiment we study the behavior of the *k*-NN using only textual features coupled with the improved text features from Chapter 4. The choice of the *k* parameter is done according to the best performing *k* for each experiment. We will use the standard evaluation protocol described previously using a distribution of 95% training images.

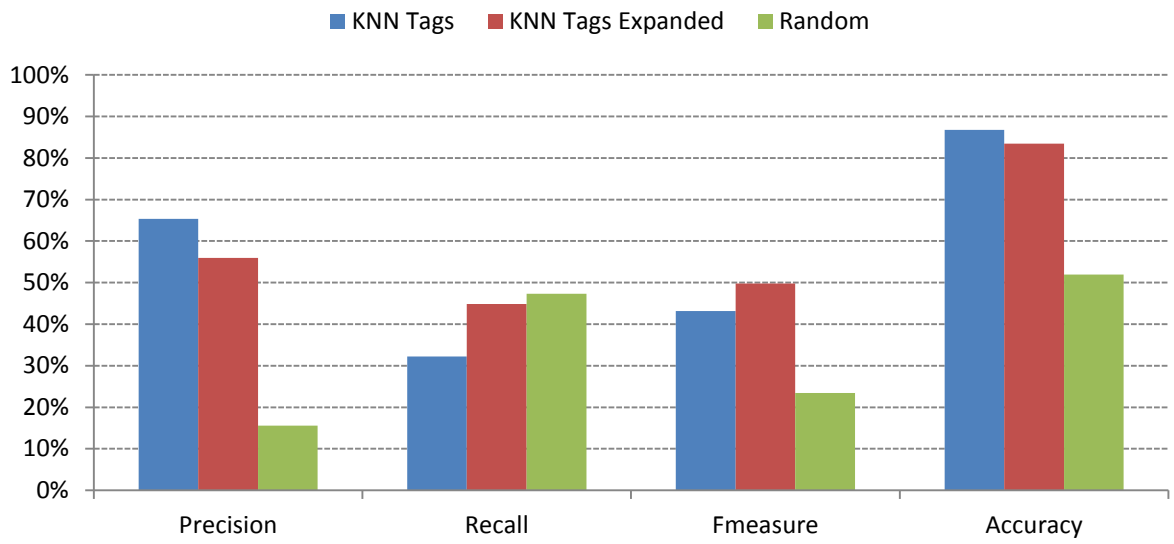

Figure 32 - Baseline *k*-NN versus knowledge based *k*-NN.

The first conclusion from the chart, and also the most important one, is the conclusion that the linguistic expansion and correction framework can improve automated annotation algorithms. Analyzing in detail, the recall metric in *k*-NN improves with the expanded user tag model which means a lower amount of false negatives, implying the expanded tag model lead to more annotations. A side-effect from more positives is a small degradation in precision. The balance of precision and recall is translated in the f-measure metric with a significantly higher performance of the algorithm with expanded user tag model when compared to the baseline implementation. Further testing with different datasets with a varied noise level could also provide more insight into the extent of the benefit from the cleaning and expansion framework.

5.4.3 Experiment 2: Single feature LLGC

In this experiment we considered single feature implementations (either visual or textual) to test the new local and global consistency algorithm when compared to the baseline *k*-NN single feature implementation. This test was executed with the standard annotation protocol using 95% training images. We used the Manhattan distance for visual features in the LLGC algorithm and Pearson correlation for textual features in the LLGC. We only used the best performing versions of the *k*-NN algorithm for each feature. Below are the charts detailing our results for single feature LLGC annotation:

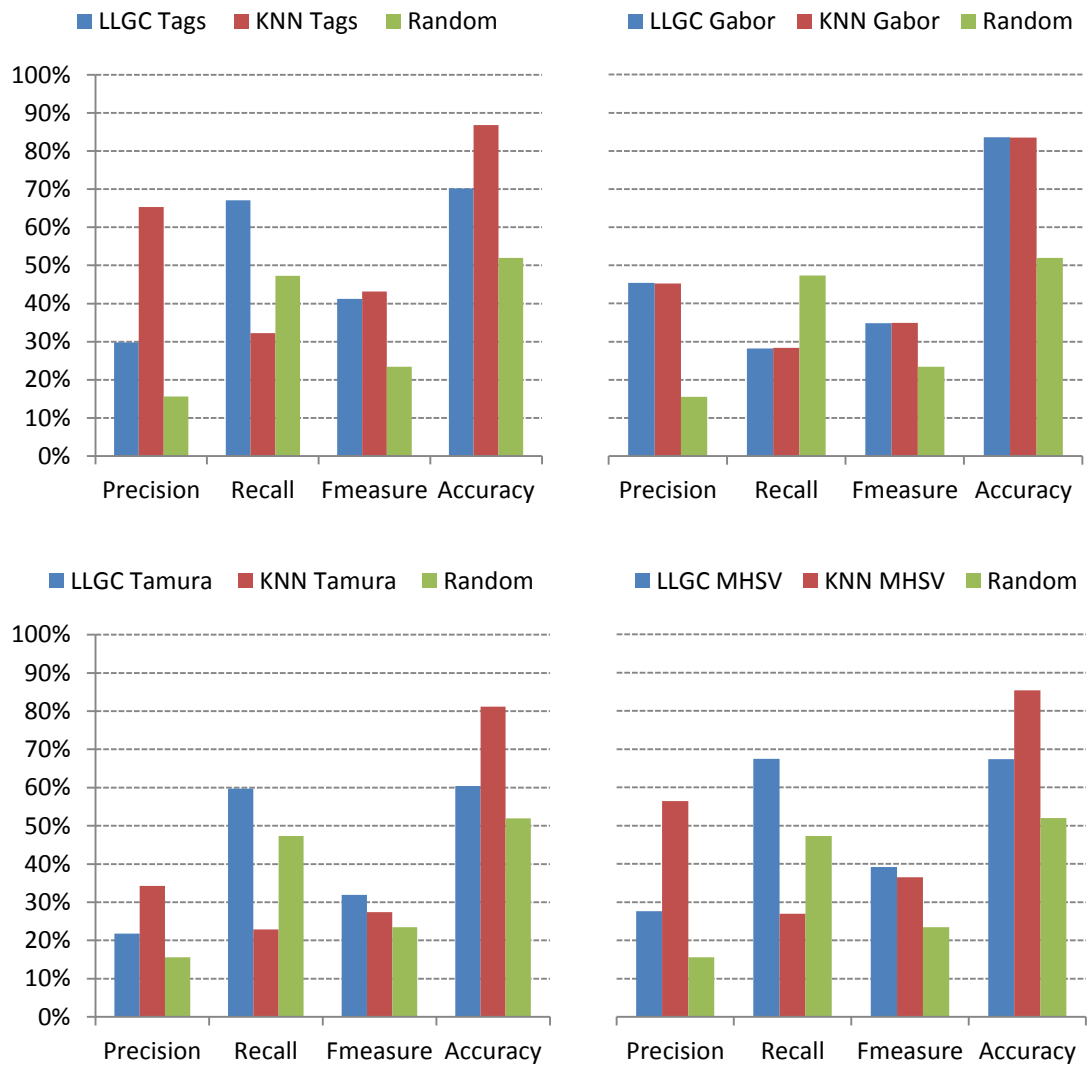

Figure 33 - Results for single feature LLGC algorithm versus *k*-NN.

From the charts we infer the following facts:

- Confirmation of the usefulness of all visual features using the LLGC algorithm. The usefulness can be explained by comparison with random annotation where the LLGC algorithm clearly outperforms random annotation for each feature space, which means the LLGC is more robust than the *k*-NN.
- The LLGC algorithm using visual features performs better than the *k*-NN algorithm for visual features. Despite having a lower precision than the *k*-NN algorithm the LLGC algorithm using visual features has a greater recall and f-measure metric. A higher recall percentage means more images were annotated, although this amount is offset by a lower precision percentage which means much more images were incorrectly annotated.

- Regarding textual tags, the LLGC algorithm is slightly worse versus the *k*-NN algorithm although the recall metric is far better than the *k*-NN algorithm. This can be explained by the greater number of false positives, an error of over-annotation. The solution for this problem could be in a more extensive phase of parameter tuning for the LLGC algorithm.

5.4.4 Experiment 3: Multi-feature *k*-NN – Top-*k* vs Avg-*k*

In this experiment we considered visual feature descriptors (Gabor and Marginal HSV) and the textual features we have previously studied combining them in the *k*-NN implementation. We will execute tests for the two multi-feature implementation variations (*Avg K* and *Top K*) for the standard annotation algorithm using 95% training images.

Below a figure showing the results for the *Avg K* and *Top K* variations of the *k*-NN multi-feature implementation is shown:

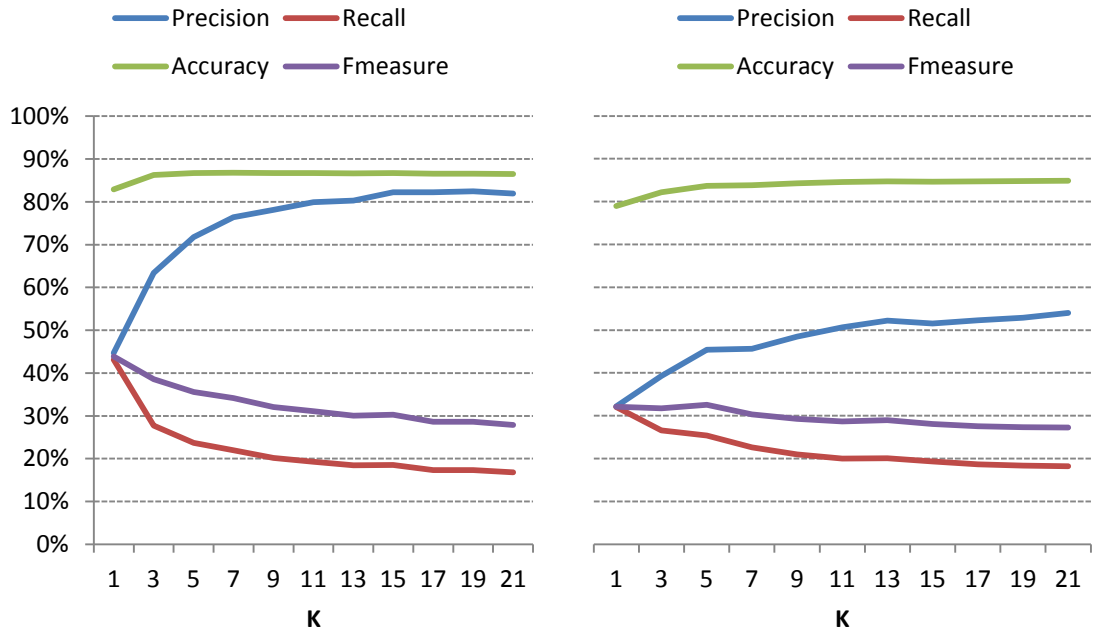

Figure 34 - Left - *k*-NN with *Avg K* variation, Right - Baseline *k*-NN with *Top K* variation.

In the case of visual and textual *k*-NN multi-feature implementation there is a similar behavior when compared to the visual-only multi-feature implementation. Both variations show a decrease in recall proportional to the increase of the *k* parameter although slower in the *Top K* variation. Regarding precision, both increase proportionally to the increase of the *k* parameter albeit the *Avg K* increases at a much faster rate. F-measure also shows a slight increase in performance from *k*=1 to *k*=5 in the *Top K* variation, which can be explained by a rapid increase in precision and a slower decrease in recall.

Based on previous single-feature *k*-NN annotation patterns, the difference in results between algorithm variations can be explained by the number of neighboring images used to annotate. In the *Avg K* variation, *k* times 3 (for each feature space) neighboring images are used while the Top K variation only uses *k* neighbors. With this increase in the number of neighbors more information is inserted into the approximation function (*Avg K*) used to annotate which eliminates the bias that comes from annotating with a small pool of neighbors. This increase in number of neighbors (*k* parameter) comes at a cost, a worse recall, since the number of true positives decreases faster with the increase of *k* than in the *Top K* variation.

Conclusions about the behavior of the algorithm are very similar to previous single-feature *k*-NN implementations. The difference between implementations lies in a significant improvement over all metrics used, which means the multi-feature implementation adds useful information to the annotation algorithm successfully improving its performance. Therefore we can conclude there is a usable link between low level data, the visual features and the textual features contributing to an increase in performance in automated annotation algorithms resulting in better semantic information to describe the images.

5.4.5 Experiment 4: Knowledge and feature fusion

After the initial single implementation of the iterative algorithm with promising results, we now explore a multi-feature implementation using the learned knowledge. For the multi-feature implementation we will alter the iterative equation to allow two feature spaces, visual and textual:

$$\boldsymbol{F(t+1) = \alpha Y + \beta S_V F(t) + \delta S_T F(t)}$$

The algorithm can now account for visual and textual features, and will be compared to the *k*-NN multi-feature implementation. For our final round of tests we will use all knowledge from the expanded user tag model with the addition of the visual features Marginal HSV and Gabor (normalizing each feature space into the affinity matrix). Our baseline will be the multi-feature *k*-NN implementation discussed in section 5.2.4 with the *Avg K* variation. For our experiment we will use the previously detailed evaluation protocol with two different distributions 95% and 25% training images.

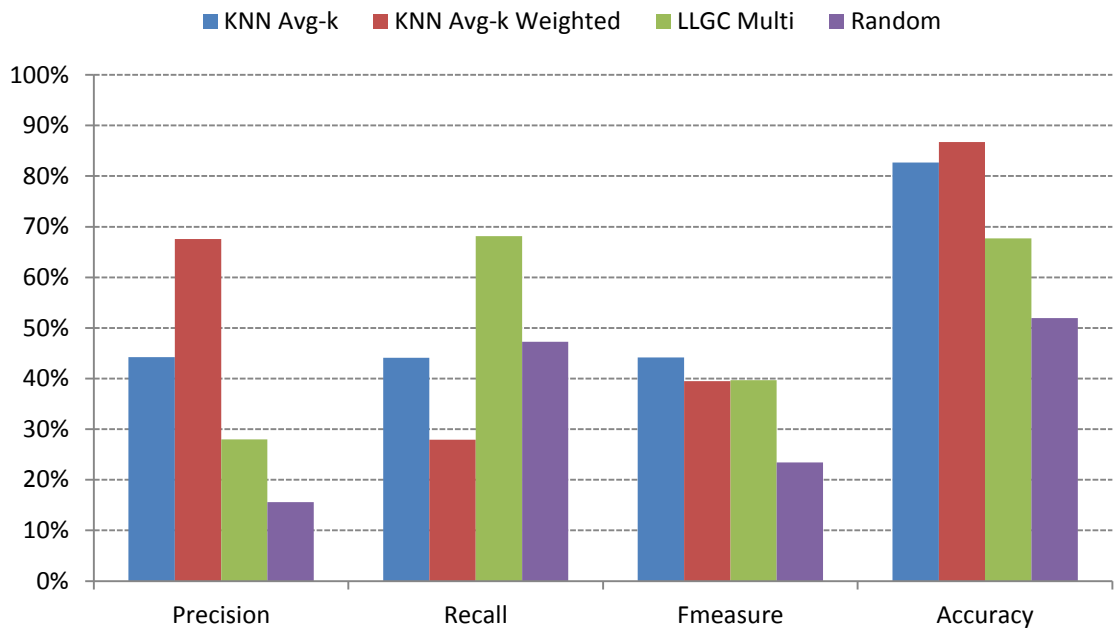

Figure 35 – Comparison of k-NN and LLGC implementations with 95% training images.

The above chart shows an important conclusion, multi-feature weighted implementations have good performance. Therefore we can conclude that the mixture of feature spaces doesn't degrade significantly the performance of the annotation algorithm. This is one step towards reducing the semantic gap between low-level features and textual features.

A deeper analysis on the chart shows the *k*-NN has slight advantage in this experiment. Regarding each individual metric:

- **Precision**: the *k*-NN implementations perform better, with the weighted variation clearly outperforming all other implementations. Precision in the LLGC can be increased with better parameter estimation to find parameters that can reduce the high number of false positives.
- **Recall**: the clear victor is the LLGC implementation where the recall value is high meaning there is a very good ratio of correct annotation between all concepts and images. The *k*-NN has a lower than random recall which means it isn't as robust as the LLGC algorithm and can be prone to instability.
- **F-Measure**: the baseline multi-feature *k*-NN has the best results with a small margin due to a good balance between precision and recall. It has reached the precision-recall break-even point which means it is a stable algorithm that doesn't emphasize one metric over the other.

- **Accuracy**: the *k*-NN implementations have significantly higher accuracy than the LLGC due to low number of false positives despite the high number of true positives of the LLGC algorithm.

Although the k-NN implementations, especially the baseline *k*-NN Avg-*k*, show a slight advantage, the LLGC algorithm still has a good performance and is also more robust than the *k*-NN with all its metrics better than random annotation. As previously said, extended parameter estimation could improve the LLGC results especially reducing the number of false positives, thus increasing precision.

For our second experiment we will use the same feature spaces but with only 25% training images, thus reducing significantly the amount of initial annotated information.

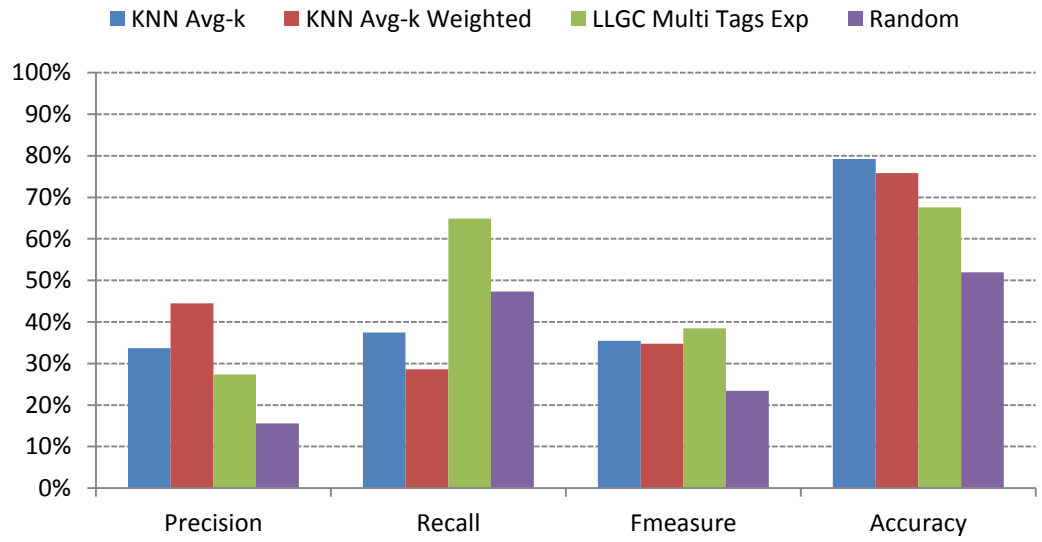

Figure 36 - Comparison of k-NN and LLGC implementations with 25% training images.

This experiment results have a similar overall pattern when compared with the previous experiment in the various metrics except for one difference, the best performing algorithm. The LLGC algorithm with less data (25%) can perform almost as good as the previous experiment while the *k*-NN implementations degrade in performance with the decrease in the number of training samples. This proves the easy adaptability of the LLGC algorithm to situations where un-annotated data is more abundant than annotated data. Realistically public image repositories will have even lower percentages of annotated data thus making the LLGC algorithm a good choice for these scenarios. It is also a more robust algorithm consistently better across all metrics when compared to random annotation. Further testing with even lower data distributions should prove the superiority of the LLGC algorithm in

automated annotation. As previously said, further parameter estimation for LLGC, where the goal is to decrease the number of false positives, can yield better results.

5.5 Summary

In this chapter we have explored the usage of linguistic and statistical knowledge from Chapter 4 by introducing the expanded user tag model. This enhancement brought a significant increase in performance over the baseline implementation.

A new algorithm (LLGC) was developed to learn from annotated data (as the *k*-NN) and un-annotated data through propagation of information. This new algorithm is based in two principles of consistency, the local consistency which is the same as the *k*-NN (neighbor locality), and global consistency where data distribution patterns, through clustering, are used to find more neighbors.

We have also researched multi-feature *k*-NN and LLGC implementations validating the usage of a multi-modal approach using all feature spaces establishing a link between textual and visual features, which could lead to a mitigation of the semantic gap.

Finally we used the knowledge from the enhanced user tag models, proposed *k*-NN improvements and the multi-feature approaches to compare each implementation establishing an improved annotation algorithm with all feature spaces. This algorithm can now handle uneven data distributions that occur in public image datasets, as the lack of quality in one feature space is usually compensated by the other feature spaces. As a result it is also less dependent on previously annotated data. The LLGC algorithm is also more robust than the *k*-NN across all evaluation metrics when compared to random annotation.

6
Conclusion

6.1 Achievements

The aim of this thesis was to research automated image annotation, accomplished using the *k*-NN algorithm and the LLGC algorithm. The first part of this thesis focused on the description of the feature spaces and the implementation of a baseline *k*-NN algorithm. In the second part we focused in extracting more knowledge from the textual feature space by implementing a tags cleaning and expansion framework. Finally we used this knowledge to improve annotation algorithms developing a multi-modal approach with the various feature spaces. Specific contributions were done in each chapter:

- We explored in Chapter 3 the baseline implementations of the *k*-NN algorithm concluding that all feature spaces used in this thesis yield relevant information for annotation tasks. This chapter established a baseline from which we could learn the intricacies of public image datasets and feature spaces. Furthermore we developed improvements over the baseline algorithm successfully enhancing its performance.
- In Chapter 4 we have studied textual features, exploring the user tag model and its structure. We discussed tagging motivations and tag relevance in user tag models and implemented a framework to clean and enhance them. This framework is composed by two components: linguistic correction and expansion using spellcheckers and dictionaries, and statistical expansion using a clustering algorithm. Combining the two techniques allowed to improve significantly the user tag accuracy addressing problems such as the low recall derived from the human

tagging effort and irrelevant or noisy tags. With this framework we created a method to withstand the uneven data distribution commonly found in user tags.

- Finally in chapter 5 we have used all the previously learnt knowledge to create a multi-feature implementation. In this chapter we have validated the conclusions from Chapter 4 including enhanced textual features for image annotation. In this chapter we explored a new algorithm, LLGC, which improves over the *k*-NN algorithm. This new algorithm proves to be an improvement for image annotation requiring fewer images for equal performance although with a more difficult parameter estimation phase than the *k*-NN algorithm. Finally we developed a successful multi-feature algorithm for both *k*-NN and LLGC implementations. The successfulness of this algorithm proved the validity of a link between visual features and textual information which allowed increased performance when both combined, thus reducing the semantic gap. By combining the various feature spaces we can withstand uneven data distribution that occurs in most public image datasets and be self sufficient requiring few previously annotated data. This is achieved by using the LLGC algorithm to propagate information.

6.2 Future work

Specific future work concerning the areas of study was also discussed at the end of chapter. Here, we summarize the most important topics:

- **User tag model enhancement:** we would like to explore open web content like Wikipedia to filter and expand the user tag model as done by Overell et al [29]. We also think that further treatment can be done on the user tag model to draw more information from otherwise noisy tags, for instance by implementing a natural language processing framework.
- **Explore other datasets**: the main dataset used in this thesis consisted in an image dataset with very noisy data which we have analyzed to clean and expand its tags. For future work, and to avoid dataset bias, we would like to explore other datasets in the *k*-NN and LLGC implementation, namely the NUS-WIDE dataset to assess if the findings are the same. Further feature spaces exploration is also a possibility namely using the EXIF feature space, which was previously detailed.
- **Further LLGC parameter estimation**: we would also like to explore further parameter estimation on the LLGC algorithm allowing to find better parameters that can yield better performance.

References

[1] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, pp. 837-842, 1996 1996.

[2] H. Tamura, et al., "Textural features corresponding to visual perception," IEEE Transactions on Systems, Man and Cybernetics, vol. 8, pp. 460-472, 1978 1978.

[3] H. Halpin, et al., "The complex dynamics of collaborative tagging," presented at the Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, 2007.

[4] M. J. Huiskes and M. S. Lew, "The MIR Flickr Retrieval Evaluation," presented at the ACM International Conference on Multimedia Information Retrieval Vancouver, Canada, 2008.

[5] T.-S. Chua, et al., "NUS-WIDE: a real-world web image database from National University of Singapore," presented at the Proceeding of the ACM International Conference on Image and Video Retrieval, Santorini, Fira, Greece, 2009.

[6] J. Magalhães, "Statistical models for semantic-multimedia information retrieval," PhD PhD Thesis, Department of Computing, University of London, Imperial College of Science, Technology and Medicine, London, 2008.

[7] S. Mizzaro, "Relevance: the whole history," Journal of the American Society of Information Science, vol. 48, pp. 810-832, September 1997 1997.

[8] E. M. Voorhees, "Variations in relevance judgments and the measurement of retrieval effectiveness," presented at the ACM SIGIR Conf. on research and development in information retrieval, Melbourne, Australia, 1998.

[9] B. Sigurbjornsson and R. v. Zwol, "Flickr tag recommendation based on collective knowledge," presented at the International conference on World Wide Web, Beijing, China, 2008.

[10] M. Ames and M. Naaman, "Why we tag: motivations for annotation in mobile and online media," presented at the ACM SIGCHI conference on Human factors in computing systems, San Jose, California, USA, 2007.

[11] C. Buckley and E. M. Voorhees, "Retrieval evaluation with incomplete information," presented at the ACM SIGIR Conf. on research and development in information retrieval, Sheffield, United Kingdom, 2004.

[12] E. Yilmaz and J. A. Aslam, "Estimating average precision with incomplete and imperfect judgments," presented at the ACM Conf. on information and knowledge management, Arlington, Virginia, USA, 2006.

[13] J. A. Aslam and E. Yilmaz, "Inferring document relevance from incomplete information," presented at the ACM Conf. on information and knowledge management, Lisbon, Portugal, 2007.

[14] T. Volkmer, et al., "Modeling human judgment of digital imagery for multimedia retrieval," IEEE Transactions on Multimedia, vol. 9, pp. 967-974, Aug. 2007 2007.

[15] C. Buckley and E. M. Voorhees, "Evaluating evaluation measure stability," presented at the ACM SIGIR Conf. on research and development in information retrieval, Athens, Greece, 2000.

[16] E. M. Voorhees, "Evaluation by highly relevant documents," presented at the ACM SIGIR Conf. on Research and development in information retrieval, New Orleans, Louisiana, United States, 2001.

[17] D. E. Rose and D. Levinson, "Understanding user goals in web search," presented at the International conference on World Wide Web, New York, NY, USA, 2004.

[18] A. Broder, "A taxonomy of web search," SIGIR Forum, vol. 36, pp. 3-10, 2002.

[19] D. Zhou, et al., "Learning with local and global consistency," presented at the Advances in Neural Information Processing Systems, Vancouver, Canada, 2004.

[20] R. Yan, et al., "Mining relationship between video concepts using probabilistic graphical model," presented at the IEEE International Conference On Multimedia and Expo Toronto, Canada, 2006.

[21] J. Liu, et al., "An adaptive graph model for automatic image annotation," presented at the Proceedings of the 8th ACM international workshop on Multimedia information retrieval, Santa Barbara, California, USA, 2006.

[22] G. A. Miller, "WordNet: A lexical database for English," Communications of ACM, vol. 38, pp. 39-41, November 1995 1995.

[23] C. Marlow, et al., "HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, To Read," presented at the Conference on Hypertext and Hypermedia, Odense, Denmark, 2006.

[24] R.-A. Negoescu and D. Gatica-Perez, "Analyzing Flickr groups," presented at the ACM Conference on Image and Video Retrieval, Niagara Falls, Ontario, Canada, 2008.

[25] J. Xu and W. B. Croft, "Query expansion using local and global document analysis," presented at the ACM SIGIR conference on research and development in information retrieval, Zurich, Switzerland, 1996.

[26] M. Mitra, et al., "Improving automatic query expansion," presented at the ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, 1998.

[27] K. Q. Weinberger, et al., "Resolving tag ambiguity," presented at the Proceeding of the 16th ACM international conference on Multimedia, Vancouver, British Columbia, Canada, 2008.

[28] (2010, July 2010). Hunspell. Available: http://hunspell.sourceforge.net/

[29] S. Overell, et al., "Classifying tags using open content resources," presented at the ACM International Conference on Web Search and Data Mining, Barcelona, Spain, 2009.