Bridging the Semantic Gaps in Information Retrieval : Advanced Context-based Image Search Using Topic Models

Bridging the Semantic Gaps in Information Retrieval: Advanced Context-based Image Search Using Topic Models

Cam-Tu Nguyen

June 2011

Acknowledgements

My deepest thank must first go to my research supervisors, Prof. Takeshi Tokuyama and Prof. Susumu Horiguchi, who offers me endless interest and motivation in scientific research, leading me this research area. I particularly appreciate their unconditional support and advice in both academic environment and daily life during the last three years. This work can not be possible without their support.

Many thanks go to Dr. Xuan Hieu Phan and Dr. Vu Ha Le, who have given me many advices and comments. Also, I would like to thank them for being my friends, my brothers who have encouraged me to overcome challenges in the search of new scientific findings.

My thanks also go to all members of Tokuyama laboratory. I sincerely thank them for their daily support, helping me with documentary procedures and with my poor Japanese. It will be missing to me without mentioning one of the most unforgettable memory - the great northeast earthquake of Japan on March 11, 2011. Above all of my terrifying, however, is the warm feeling when thinking about how much caring I received from my lab members who continued contacting me to ensure my safety (Prof. Takeshi Tokuyama, Prof. Jinhee Chun, Natsuda Kaothanthong) or being with me in that difficulty (Prof. Akiyoshi Shioura, Tran Phuong Nhung).

I highly acknowledge Prof. Quang Thuy Ha and my colleagues who offered me inspiration and led me to the field of Data mining. Especially, I would like to thank my co-authors (Thu Trang Nguyen, Dieu Thu Le) for their discussions, scientific supports. It is very nice working with high motivated researchers like them. Moreover, I would like thank Prof. Kobus Barnard for letting me access the common benchmark in image annotation. This research would be much more difficult without his support.

To all of my dear friends, who I cannot name all of them since I am afraid of missing someone, I sincerely appreciate their friendships, which have brought multiple colors to make my life not just black and white.

Finally, I would like to mention my family (my parent, my younger brother, W). I cannot say enough thankful to them simply because they are the reasons for me to keep on trying.

Contents

Chapter 1

Introduction

1.1 Information Retrieval

So far there have been several definitions of Information Retrieval (IR). According to Jiawei Han and Micheline Kamber [36], IR can be defined as the field developed in parallel with database systems. The difference between database systems and IR is that the former works with query, transaction while the later deals with the organization and retrieval of information from a large number of text-based documents. Christopher D.Manning [64], on the other hand, referred to IR as finding material (usually documents) of unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

These definitions match major activities of search engines in the early days, when text is the center to information retrieval, but not be suitable in recent time. Nowadays, the fast development of hardwares and Internet has led to an enormous amount of data exchanged and stored in multiple types (images, videos, ...) and multiple languages. More and more number of users engage in information retrieval when they are surfing on the Internet. At the same time, search engine companies adapt their products to serve for the greater need of information from users. As a result, the field of IR can cover much broader problems than the core definitions stated above.

When viewing IR as a broader field, which helps users in browsing and filtering information, IR also covers clustering and classification [64]. In classification, documents are (manually or automatically) divided into predefined categories such as “Art”, “Computer”, “Business”, and so on. This functionality has been supported in many large search engines like Google or Yahoo in forms of Directories (Google Directory, Yahoo Directory). On the other hand, clustering gathers documents into coherent groups, in which documents in one group are “close” to each other and they are far away from documents in other group. In comparison with classification, clustering is “unsupervised” that is no expert needed to predefine categories or manually label sample data.

Based on the data types, we can also classify IR into several subcategories such as text retrieval, image retrieval, video retrieval, cross-language retrieval, searching for locations (Map services), etc. In this way, information retrieval fast become the dominant form of information access, surpass the traditional database access. Not only on the Internet, IR can also be applied to personal or institute collections. In addition, IR is related to multiple fields including natural language

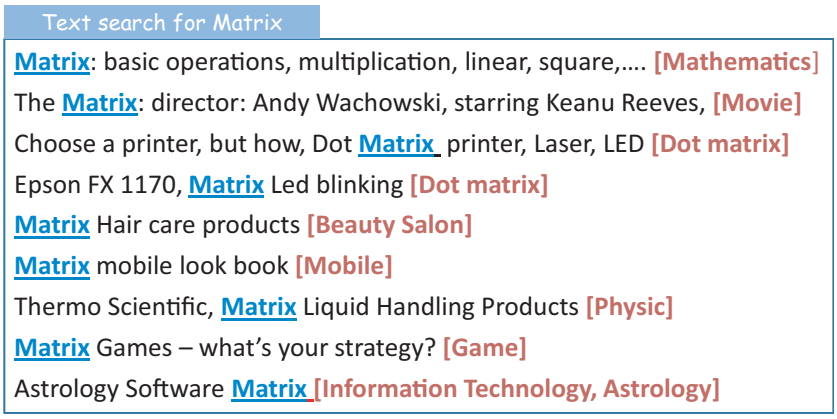


Figure 1.1: Semantic Gaps in Text Retrieval

processing, computer vision, image/video Processing, machine translation, etc.

Toward advanced context-based image retrieval, there exist several semantic gaps that we need to deal with. We consider two types of gaps that are the gap from visual representation of images to words, and the gap from word space stored in the computer and the vocabulary of users. For the former gap, image annotation is often the solution in the literature of image retrieval. The later gap is pretty much related to the gaps in text retrieval. In order to close these gaps, we first focus on semantic gaps in multiple applications of text retrieval, which are clustering and labeling, matching and ranking. We then demonstrate that bridging the gap from word space to concept space can indeed help back to close the gap in between visual space and word space in image retrieval. Before discussing our ideas in detail , let us clarify what the semantic gaps are in information retrieval.

1.2 Semantic Gaps in Information Retrieval

In Information Retrieval (IR), the semantic gap is the difference between what computers store and what users expect via their queries. There are several reasons for the existence of those gaps such as synonymy or polysemy in text retrieval, or the typical gap between low-level representations and textual labels in image retrieval. This problem, which is underlined in many aspects of Web mining and Information Retrieval (IR), has posed a lot of challenges. In text retrieval, the gap is commonly caused by some natural linguistic phenomena such as synonymy or polysemy. Synonymy that is two or more different words have similar meanings causes difficulty in matching two related documents. For example, the similarity between two documents (particularly the short ones) containing "movie" and "film" is probably lower than what we expect. On the other hand, polysemy means a word can have multiple meaning. One example is the word "bank" exists in several contexts such as "organization" or "river side". Consequently, we may accidentally put an advertising message about a bank (an organization) on a Web page about bank (of some river). Another example of semantic gap in text retrieval is demonstrated in Figure 1.1. Here, we can encounter the query "matrix" from different contexts such as in documents about printers, "movies" or "mathematics".

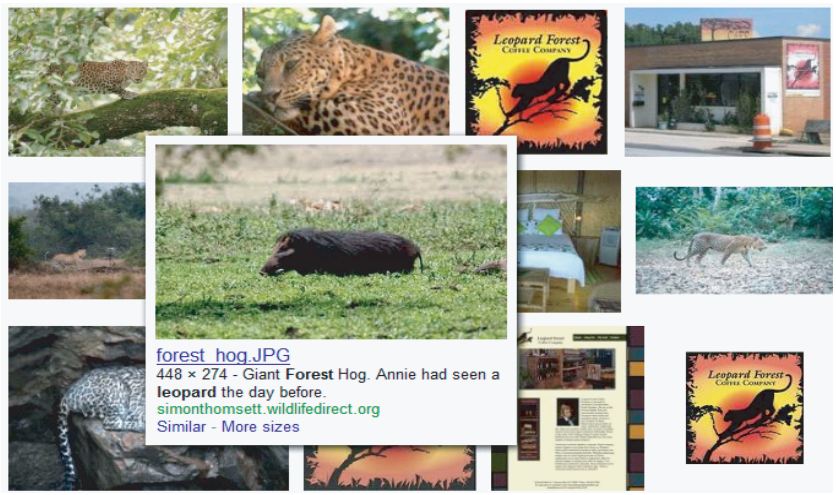


Figure 1.2: Semantic Gap in Image Retrieval. Search results from Google for the query "leopard forest", where the emerging photo depicts "giant hog" but the surrounding text is "Giant *forest* hog. Annie had seen a **leopard** a day before".

This can be seen as the semantic gap between what the computer stored the word "matrix" and what users need (printers, or mathematics, etc.).

The problem of semantic gap is more challenging in Multimedia Mining and Image/Video Retrieval [21]. Most of current commercial search engines still use surrounding texts as the main source for image retrieval. Since they ignore the visual content, some of the results may have nothing to do with the query. Figure 1.2 illustrates this problem where some search results from Google do not depict "leopard" and "forest". Indeed, if you search only the query "leopard", the results from Google are very good in which the first page contains all images with the file names related to "leopard". However, if we combine queries together, the results are much worse even with "common" combination such as "leopard forest" or "mountain sun water", which can be captured with content-based image retrieval. In order to obtain advanced image retrieval, visual content of images need to be considered, which is also the main objective of the subfield named content-based image retrieval (CBIR).

While early content-based image retrieval systems were based on the query-by-example schema, which formalizes the task as search for best matches to example images provided by users, the attention now moves to query-by-semantic schema in which queries are provided in natural language. This leads to the problem of automatic image annotation, which is concerned with assigning labels to images for later retrieval. Due to the semantic gap between low-level image representations (such as color, contour, shape) and high-level concepts (tiger, mountain, etc.), IR researchers commonly find image annotation a difficult problem to cope with. Now, suppose that we have a perfect image annotation system, the difference between the vocabulary that we use for image annotation, which is usually limited, and the large vocabulary that humans use pose another layer of semantic gap for image retrieval. Indeed, this problem is once again the semantic gap in text retrieval. As a result,

in order to bridge the gaps for image retrieval, the semantic gaps in text retrieval also need to be considered as a significant part.

1.3 Bridging the Semantic Gaps: Related Works

1.3.1 Relevance Feedback and Query Expansion

There have been a lot of attempts to close the gaps in Web Mining and Information Retrieval. The relevance of returned documents can be improved by using relevance feedback or query expansion [64].

Relevance feedback is the process where users interact with the system to refine what they need. The process initializes with a query given by users. The system processes the query and displays results for users to judge. Users then select some good results (feedback) and let the system use the feedback to improve searching and display revised results to users. The procedure may continue with several iterations of refining search results. Relevance feedback is suitable to image search where it is hard for users to formulate what they need at the beginning. This method, however, requires online calculation, thus may make impatient users not want to collaborate.

While users give feedback about documents by marking relevant ones in relevance feedback, query expansion, on the other hand, gives additional information in form of query. This type of querying has supported by most of large search engines like Google or Yahoo. In order to suggest queries to users, several resources can be used such as word thesaurus or co-occurrence analysis from query log.

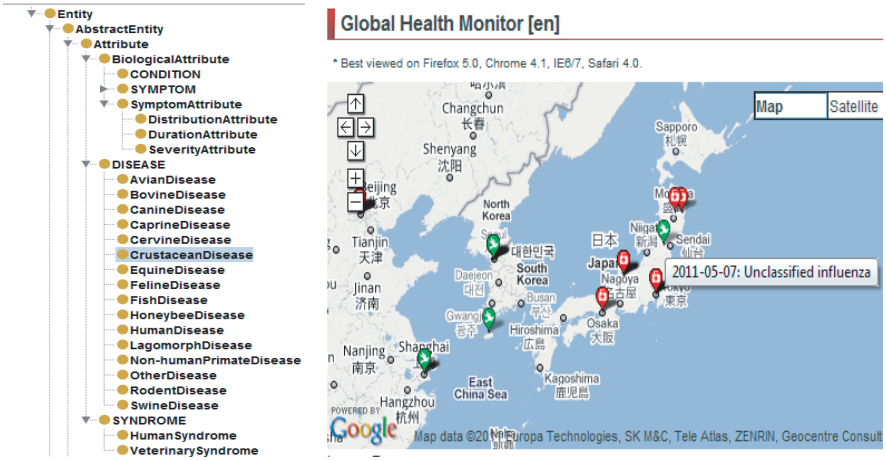
1.3.2 Knowledge Based Approach

Some studies use taxonomy, ontology and knowledge base to represent the semantic correlation between words for better contextual matching [7, 91, 31, 95]. Here, knowledge base requires manually construction with the support of experts in the domain. One example is the BioCaster ontology [95], which is the knowledge base constructed to capture relationships among diseases, symptoms, and other related issues in medical domain. This ontology brings rich semantic to annotate documents with predefined concepts such as automatically recognizing disease in a document. The annotated texts are later used for higher information management problem such as disease tracking from rumors or news on the Internet (Figure 1.3.2).

1.3.3 Context and Semantic Analysis

Since the early days, Latent semantic indexing (LSI), which is based on Latent semantic analysis (LSA) has been exploited to map words into the concept space so as to improve the relevance of retrieved results [59, 64]. Also, LSA has been used to analyze contexts of entities [81] to improve named entity classification.

In the context of image retrieval, many noticeable approaches have been proposed to bridge the "semantic gap" for better image retrieval. Some approaches attempt to reduce annotation errors by making use of word relationships such as {fish, ocean} and {desert, sand}[61, 51, 119, 118, 108].



(a) BioCaster Ontology (b) BioCaster global health management

Other approaches make use of external resources such as auxiliary texts of web images [104, 82, 29], Wordnet and ontology [51, 82], Google distance [108], click through data [98], and Wikipedia articles [85]. Topic-based approaches model joint distributions of visual features and words [12, 69, 41, 42]. On the other hand, global features [96, 25, 62] are used to capture contexts for image retrieval, and object detection/ location. Recently, studies [106, 96] on jointly modeling scene classification and image annotation (or object detection) have been conducted on the attempt to exploit the context from the scene.

1.4 Contributions and Organization of the Thesis

1.4.1 Contributions

The contributions of this thesis is elaborated in three problems, which are related to four main Chapters 3,4, 5 and 6. The problems are related to semantic gaps ranging from low level image feature to high level human concepts. We point out in the following the contributions for each problem separately.

Enriching short texts for better clustering, matching, and ranking

We focus on the problem of search results clustering and labeling in Chapter 3, and the problem of contextual advertising in Chapter 4. The center task to these problems is to measure similarities between short texts such as results obtained from search engines, or advertising messages in content advertising. Unfortunately, short texts, which contain from ten to dozen words, are very sparse. Along with the semantic gaps caused by different word choices, synonymy, polysemy, abbreviations, named entities, etc., we can not obtain enough word co-occurrence or shared contexts for good similarity.

Topic 1	Topic 2	Topic 3	Topic 4
actor	algebra	dot	software
Hollywood	math	print	computer
director	multiplication	matrix	large-scale
screening	matrix	Epson	Hardware
action	invert	printer	matrix
matrix	number	quality	computing

Table 1.1: sample topics of a topic model

Our main contribution is a general framework, which is based on hidden topic analysis from a large dataset, to reduce data mismatching within short texts. Roughly speaking, we can think of a topic model as a word-topic probability table, where one cell represents conditional probability of word given topic. Sample topics of a topic model are illustrated in Table 1.1, where each topic is represented by top words with highest probabilities given that topic. The beauty of topic modeling is that we can do it automatically from a collection of documents with (almost) no interfere from humans. The topic model of a large dataset can be used as one type of knowledge base to bridge the semantic gaps in matching short texts. For example, two short texts *"d1: matrix, basic, operations, multiplication"* and *"d2: matrix, math, vector, real, number"* have only one word "matrix" in common. Consequently, it is difficult for computers to obtain good similarity based on surface word matching. Now, suppose that we have built a topic model in advance as in Table 1.1, the conjunction of words of two short texts highly related to Topic 2. Since the two texts share topic 2, they are closer in the space of both words and topics. Thus, a better way to measure similarities between short texts is obtained for multiple applications such as clustering, labeling, ranking or matching.

Inferring scene settings using topic models for better image annotation

Image annotation is to automatically assign labels to images for a better way of organizing the large amount of images on the Internet. This problem is a difficult task due to the semantic gap between visual features and textual labels, which means extracting semantic labels is hard when using only low level image features such as color or textures. Compared to object recognition, the number of labels/concepts in image annotation is larger. Moreover, training dataset for image annotation is weakly labeling that is labels are assigned to images without indication of the correspondence between regions and labels.

The contribution in Chapter 5 is a feature-word-topic model, which is based on hidden topics to guess scene setting for better annotation. The large number of labels, which is often the case of image annotation, brings more and more ambiguities to annotation systems that are merely based on visual features. One common mistake is between "sky" and "ocean" since they are both stipulated by a large blue region in pictures like in Figure 1.3. In order to resolve this problem, we also apply topic modeling from a set of image captions to find out topics, each of which is one co-occurrence of related labels and considered as one scene setting. The main idea is that in spite of ambiguities, reasonable annotations tends to agree on the scene/topic. For example, top

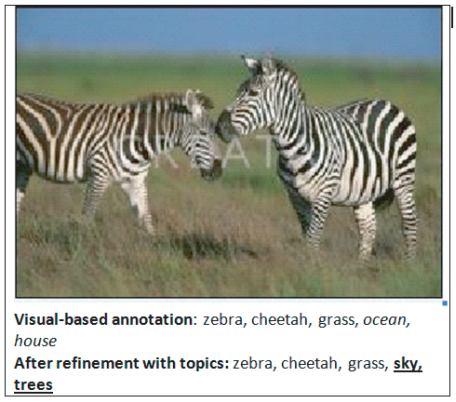


Figure 1.3: Correct annotations agree on the scene setting

annotations based on visual features in Figure 1.3 describe "grass field" scene. By taking topics into account, we can reduce mistaken labels and introduce better annotations as in Figure 1.3.

The feature-word-topic model is simple and can be extended in multiple ways. For example, we can also extend the model to include file names, surrounding texts to guess the scene and refine image annotation. The separation between feature-word and word-topic parts makes feature-word-topic model easier to adapt to richer topic/semantic analysis techniques as well as to tune the annotation performance. For instance, one can improve detecting performance of some (limited) labels and give them higher confidence for topic inference. As a result, we are able to take advantage of the successful object recognition like "face detector", which is very fast and widely used in modern digital cameras.

The role of context in image annotation

Image annotation has several typical challenging problems that are 1) the large variety of visual representation; 2) the weakly labeling problem; and 3) the domination of negative examples over positive examples. The first problem is related to the semantic gap between low level feature and textual labels. One way to overcome this issue is to investigate on feature extraction. Unfortunately, due to the large number of labels, we can not tune feature extraction for every label. The second problem is often addressed by multiple instance learning (MIL) approach that often makes use of the commonality among images with one common label, to reduce the ambiguity of weakly labeling. In this approach, we need to build one classifier per label. This is where comes the third problem, the domination of negative examples (images without the specified label) over positive examples (images with the specified label). The common solution is sampling to reduce the number of negative examples and bring more efficient training process. In spite of current successes of these solutions, we find several questions remaining open:

- How can we compromise between the variety of features and the large number of labels?
- How can we exploit context to reduce ambiguity in image annotation?

- What is the reasonable sampling method for reducing the domination of negative examples and reducing training times?

The three questions have been addressed in Chapter 6 with an integrated solution, a cascade of multi-level multi-instance classifiers (CMLMI) for image annotation. Multi-level means we divide images in multiple ways, from no division to finer grid-based division. For each level, we can exploit multiple feature extraction methods. By making use of multi-level feature extraction, we can select suitable feature types for different types of labels. The point is features from coarse levels (the whole image, for example) are more suitable for background labels such as "sky, ocean, city". On the other hand, features from finer levels are more suitable for foreground labels such as "tiger, house, zebra". Given a label, we build "multi-instance classifiers" across levels in a cascade manner, one per feature space obtained from one feature extraction method. Here, cascading means learning classifiers in finer levels is dependent on learning classifiers in coarse levels. The main idea is that the coarser levels are more successful in detecting the related-scene, we just have to focus on sampling negative examples of related scene to detect foreground objects. The negative samples of same scene often share background with positive examples, in which we do not know which regions are positive to the given label due to weakly labeling. The inclusion of negative samples of same scene provides hints to exclude background regions in positive examples. In other word, we reduce the ambiguity of weakly labeling.

A side contribution of Chapter 6 is the discussion of topic modeling for image captions. We exploit the idea in Chapter 5 to refine annotation with topics. The new content here is we exploit different topic modeling method for refining CMLMI. Also, we discuss the sparseness of image captions and topics for images. The point is the number of topics associated with one image is often smaller than that in normal documents. This discussion is useful for further investigation in modeling scenes. We provides some heuristic methods to partly overcome this difficulty.

1.4.2 Thesis Overview

The rest of this thesis is organized into 6 chapters:

- Chapter 2 presents methodologies in semantic representation such as semantic networks, semantic space, hidden topics as well as the relationships among them.
- Chapter 3 proposes a general framework to enrich data presentation in short documents in order to obtain better Web search clustering and labeling.
- Chapter 4 adapts the framework in Chapter 3 to the problem of matching, ranking for online advertising. By doing so, we show that our framework is adaptable and efficient in multiple applications.
- Chapter 5 proposes a feature-word-topic model in which the feature-word part closes the semantic gap between visual representation and word space, and the word-topic part is in charge of the semantic gap between word and concepts. Also, we show that by closing the second gap bring benefits to improve the first step.

- Chapter 6 presents a novel method for image annotation, which is based on cascading multi-level multi-instance classifiers. We show experimentally and theoretically that this method can help reducing ambiguity in image annotation and reduce training time.
- Chapter 7 concludes the dissertation with important remarks and future works.

Chapter 2

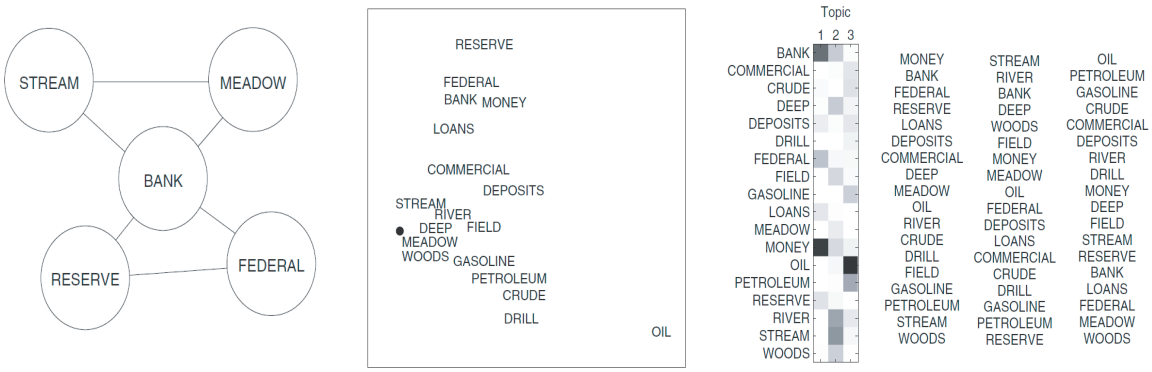
Hidden Semantic and Topic Analysis

2.1 Introduction

Over the years, semantic representation is an active topic in artificial intelligence, machine learning, data mining, etc. as well as a matter for debate in cognitive psychology. In order to make machines "more intelligent" and bridge the "semantic gap", computer scientists are interested in studying how humans perceive semantic concepts. Recent review in psychology [34] claimed that many aspects of human vision is interpretable using statistics of natural scenes. Moreover, human memory tends to associate particular events occurring in the world with high probabilities. They then summarized three typical approaches to semantic representation as follows (see Figure 2.1 for demonstrations).

- Semantic Network: Concepts are represented by nodes and relationships between concepts are encoded by edges. Semantic network usually be hand-coded by analyzing the domain of interest and represented by ontology. Wordnet is one famous example of this type.
- Semantic Space: Words are represented as points in Euclidean space and proximity implies semantic association. This is the solution produced by Latent Semantic Analysis (LSA) [22], which is sometimes referred to as Latent Semantic Indexing (LSI) in the context of its application in Information Retrieval.
- Topic Models: This approach is based on the idea that documents are mixture of topics and each topic is a probability distribution over words. Although topic models also aim at semantic representation and dimensionality reduction as LSA, their approach is in the view of statistically generative models instead of vector space. Probabilistic Latent Semantic Analysis (pLSA) [40] is the pioneer in this approach. Latent Dirichlet Allocation (LDA) [11] was successively proposed as a more complete generative model compared to pLSA, this topic model has received more and more attentions with applications in multiple fields including text and image retrieval.

Although semantic network can capture various relationships among words, building a semantic network is very expensive and domain-dependent. In order to explore flexible solutions to bridge



(a) Semantic Network (b) Semantic Space (c) Topic Model

Figure 2.1: Methods for Semantic Representation [34]

semantic gaps, we follow the last two semantic representations that are semantic space and topic models. The typical method in semantic space representation, Latent Semantic Analysis (LSA), is based on a mathematical tool that is Singular Value Decomposition to map documents of words into a space of smaller dimensions, the space of concepts. On the other hand, a topic model is a generative model for documents, in which it specifies how a document can be generated. For a new document, one chooses a distribution over topics. Then, for each word holder in that document, one chooses a topic randomly according to the distribution and draws a word from that topic. Standard statistical techniques can be used to invert this process to infer the set of topics that is responsible for generating the collection of text. This step is called model estimation. Given a model, we can perform topic analysis to obtain posterior distributions (usually in form of topic distributions) over words of new documents, which is referred to inference step. Maximum likelihood is one of the most popular method for estimation and inference. As topic models become more sophisticated, exact inference of posterior distributions is intractable. As a result, approximation is often used as an alternative solution. Depending on the model, we can exploit a sampling approach (Markov Chain Monte Carlo, Gibbs sampling) or a variational method (Mean Field variational) for approximate inference.

This chapter summarizes methodologies to analyze semantic and topics from a collection of text documents as the foundation for later chapters. We begin with an introduction to Latent Semantic Analysis (LSA) [22] and demonstrate its application in information retrieval. We then discuss more details about topic models, from the earliest one, pLSA, to recent proposed models that are Latent Dirichlet Allocation (LDA) and Correlated Topic Model (CTM). LDA is chosen as a typical model which we can use Gibbs sampling for inference, and CTM is the example of using variational method for inference.

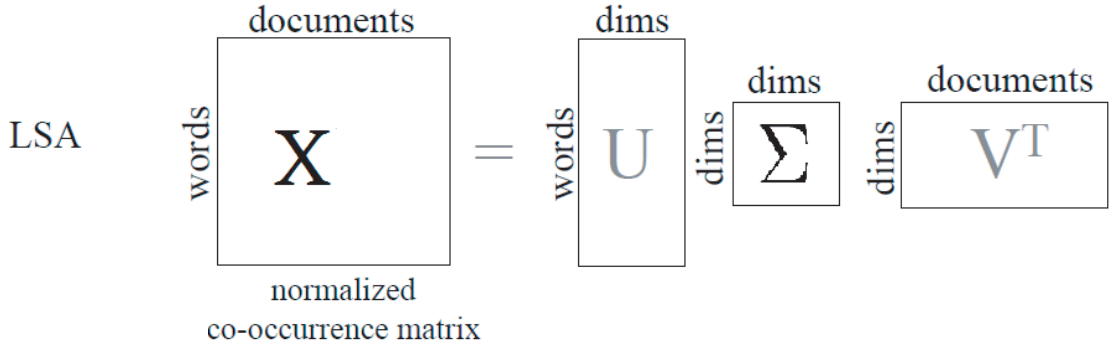


Figure 2.2: Matrix decomposition for Latent Semantic Analysis

2.2 Latent Semantic Analysis

Representing text corpora effectively to exploit their inherent essential relationship between members of the collections has become sophisticated over the years. Latent Semantic Analysis (LSA) [22] is a significant step in this regard. LSA uses a Singular Value Decomposition (SVD) of the term-by-document X matrix to identify a linear subspace in the space of term weight features that captures most of variance in the collection. Singular Value Decomposition (SVD) is one type of factor analysis and related to many other mathematical tools including eigen vector decomposition, principle component analysis. Decomposition of X produces three other matrices of special forms: U and V are the matrices of left and right singular vectors and Σ is the diagonal matrix of *singular values*. The matrices U, V^T, Σ show a breakdown of the original relationships into linearly independent components of factors. In general, matrices U, V and Σ are of full rank, and the diagonal elements of Σ are constructed to be all positive and ordered in decreasing magnitude.

We have the i-th row t_i of matrix U corresponding to the word i, and the j-th column of matrix V corresponding to the document d_j, which depend on all column vectors in matrix U and row vectors in matrix V^T. Many of the components of U, V^T, Σ are very small and can be ignored. Based on this observation, the LSA approximation of X is computed by selecting k largest singular values of Σ and the corresponding k (singular) column vectors from U and k (singular) row vectors of V. The appealing thing in this approximation is that not only does it have the least square error, but also it translates the terms and document vectors into a concept space. The vector $\hat{t}_i$ then has k entries, each gives the occurrence of term w_i in one of the k concepts. Similarly, the vector $\hat{d}_j$ gives the relation between document j and each concept. This approximation can be written as $\hat{X} = \hat{U}\hat{\Sigma}\hat{V}^T$. Based on this approximation, we can perform a number of measurement as follows:

- Measure how related documents j and q are in the concept space by calculating dot product or cosine similarity using vectors $\hat{d}_j$ and $\hat{d}_q$.
- Compare terms i and p by measuring the similarity between $\hat{t}_i$ and $\hat{t}_p$ in concept space.
- Given a query, we can consider it as a short document, and compare it to documents in the concept space. To do that, we must first translate the query into the concept space with

the same transformation used on the document, i.e. $d_j = \hat{U}\hat{\Sigma}\hat{d}_j$ and $\hat{d}_j = \hat{\Sigma}^{-1}\hat{U}d_j$. This means that if we have a query vector q, we must perform the translation $\hat{q} = \hat{\Sigma}^{-1}\hat{U}q$ before comparing it to the document vectors in the concept space.

The advantage of LSA is that it can achieve considerable reduction in large collections and reveal some aspects of basic linguistic notions such as synonymy or polysemy. On the other hand, the drawback of LSA is that the resulting concepts might be difficult to interpret [109]. For example, a linear combination of words such as *car* and *truck* could be interpreted as a concept *vehicle*. However, it is possible for the case in which the linear combination of *car* and *bottle* to occur. This leads to results which can be justified on the mathematical level, but have no interpretable meaning in natural language.

2.3 Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis (pLSA) proposed by Thomas Hofmann[39, 40] was the successive attempt to capture semantic relationship within text. It relies on the idea that each word in a document is sampled from a mixture model, where mixture components are multinomial random variables that can be viewed as representation of "topics". Consequently, each word is generated from a single topic, and different words in a document may be generated from different topics.

2.3.1 The Aspect Model

Suppose that we have given a collection of text documents $D = \{d_1, \ldots, d_N\}$ with terms from a vocabulary $V = \{w_1, \ldots, w_M\}$. The starting point for pLSA is a statistical model namely *aspect model*. The aspect model is a latent variable model for co-occurrence data in which an unobserved variable $z \in Z = \{z_1, \ldots, z_K\}$ is introduced to capture the hidden topics implied in the documents. Here, N, M and K are the number of documents, words and topics respectively. Hence, we model the joint probability over $D \times V$ by the mixture as follows:

$$P(d, w) = p(d)P(w|d) \tag{2.1}$$

$$P(w|d) = \sum_{z \in Z} P(w|z)P(z|d) \tag{2.2}$$

Like other latent variable models, the aspect model depends on a conditional independence assumption, i.e. d and w are independent conditioned on the state of the associated latent variable. The graphical model represents this idea is demonstrated in Figure 2.3.

2.3.2 Model Fitting with Expectation Maximization Algorithm

The aspect model is estimated with the traditional Expectation Maximization (EM) procedure for maximum likelihood estimation. Here, the likelihood is calculated as follows:

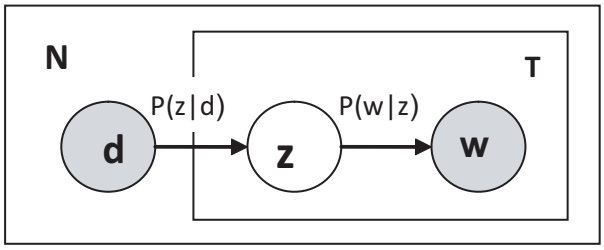


Figure 2.3: Graphical model representation of the aspect model

$$\begin{aligned}\mathcal{L} &= \prod_{d\in D}\prod_{w\in V} P(d,w)^{n(d,w)} \\ &= \prod_{d\in D}\prod_{w\in V}\left\{p(d)\sum_{z} p(w|z)p(z|d)\right\}\end{aligned}$$

where n(d,w) denotes the term frequency that is the number of times w occurred in d. We would like to find $p(z|d)$ and $p(w|z)$ to maximize the likelihood $\mathcal{L}$. However, the presence of sum inside the likelihood function results in complicated expression for the maximum likelihood solution. In practice, Expectation Maximization (EM) algorithm is used to maximize the expectation of the complete data log likelihood, which is the log likelihood with the inclusion of hidden variables z. In this way, it approximates the solutions for likelihood maximization. More specifically, EM iterates two coupled steps: (i) an expectation (E) step in which posterior probabilities are computed for the latent variables; and (ii) a maximization (M) step where parameters are updated. Standard calculations give us the E-step formula:

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'\in Z} P(z')P(d|z')P(w|z')}$$

In the M-step, we maximize the expectation of the complete-data log likelihood with respect to the posterior distributions, which can be calculated as follows:

$$\begin{aligned}E_{p(z|d,w)}[\log p(d,z,w)] &= \sum_{k=1}^{K}\sum_{d\in D}\sum_{w\in V} p(z_k|d,w)\log p(d,z_k,w) \\ &= \sum_{k=1}^{K}\sum_{d\in D}\sum_{w\in V} p(z_k|d,w)\times n(d,w)\left\{\log p(d)+\log p(z_k|d)+\log p(w|z_k)\right\}\end{aligned}$$

Note that $p(z|d,w)$ is calculated in E-step and fixed in M-step. We then exploit Lagrange method to optimize $E_{p(z|d,w)}[\log p(d,z,w)]$ with constraints $\sum_z p(z|d)=1$ and $\sum_{w\in V} p(w|z)=1$. As a result, we have the following updates in M-step:

$$p(w|z) = \frac{\sum_{d \in D} n(d,w) P(z|d,w)}{\sum_{w' \in V} \sum_{d' \in D} n(d',w') P(z|d',w')}$$

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

$$P(z) = \frac{\sum_{d \in D} \sum_{w \in V} n(d,w) P(z|d,w)}{\sum_{z'} \sum_{d' \in D} \sum_{w' \in V} n(d',w') P(z'|d',w')}$$

where $n(d,w)$ and $n(d)$ are the number that the word w appears in document d and the length of document d respectively.

2.3.3 Probabilistic Latent Semantic Space

As mentioned earlier, topic models do not consider words as points in spaces and it is the essential difference from semantic space representation such as LSA. There is, however, a coherent connection between two types of representation. This section will demonstrate this link between semantic space and generative model of pLSA. Similar analysis can also be conducted for other topic models such as LDA, CTM.

Let us consider topic-conditional multinomial distributions $p(.|z)$ over vocabulary as points on the $M-1$ dimensional simplex of all possible multinomial, where M is the size of the vocabulary. Via convex hull, the K points define a K-1 dimensional convex region $\mathcal{R} = conv(p(.|z_1), p(., z_2), \dots p(., z_K))$ on the simplex [40]. The modeling assumption expressed by Equation 2.2 is that conditional distribution $P(w|d)$ for all documents are approximated as a convex combination of $P(w|z)$ in which the mixture component $P(z|d)$ uniquely define a point on $\mathcal{R}$. A simple illustration of this idea is shown in Figure 2.4.

In order to clarify the relation to LSA, it is useful to reformulate the aspect model as parameterized by Equation 2.2 in matrix notation. By defining $\hat{U} = [P(d_i|z_k)]_{i,k}$, $\hat{V} = [P(w_j|z_k)]_{j,k}$ and $\hat{\Sigma} = diag(P(z_k))_k$ matrices, we can write the joint probability model P as a matrix product $P = \hat{U}\hat{\Sigma}\hat{V}^T$. Comparing this with SVD, we can have some observations [40]: (i) outer products between rows of $\hat{U}$ and $\hat{V}$ reflect conditional independence in pLSA; (ii) the mixture proportions inpLSA substitute the singular values. Nevertheless, the main difference between pLSA and LSA lies on the objective function used to specify the optimal approximation. While LSA uses Frobenius or L_2 norm, pLSA relies on the likelihood function of multinomial sampling and aims at an explicit maximization of the predictive power of the model.

2.4 Latent Dirichlet Allocation (LDA)

While Hofmann's work [39, 40] is a useful step toward probabilistic text modeling, it suffers from severe over-fitting problems [38]. Additionally, although pLSA is a generative model of the documents in the estimated collection, it is not a generative model of new documents. In other words, it is not clear how to assign probability to a document outside the training set [11]. The Latent

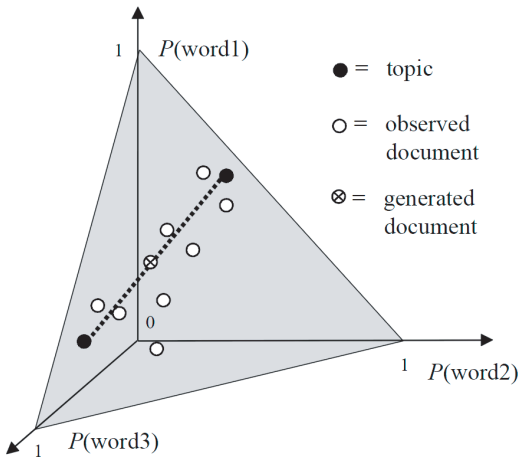


Figure 2.4: Sketch of the probability sub-simplex spanned by the aspect model [93]. Here, the triangle represents the $M-1$ simplex on 3-dimensional word space. The dark line on the simplex represents the sub-simplex spanned by two topic points. Every document is approximated in the topic space by projecting onto the line (the topic sub-simplex).

Dirichlet Allocation (LDA), first introduced by Blei et al. [2003], is the solution to these problems. Some sample topics estimated using LDA are depicted in Figure 2.5.

2.4.1 Generative Model

LDA [11, 38, 79] is a generative graphical model as shown in Figure 2.6. It can be used to model and discover underlying topic structures of any kind of discrete data in which text is a typical example. LDA was developed based on an assumption of document generation process depicted in both Figure 2.6 and Table 2.1. This process can be interpreted as follows.

In LDA, a document $\overrightarrow{w}_m = \{w_{m,n}\}_{n=1}^{N_m}$ is generated by first picking a distribution over topics $\overrightarrow{\vartheta}_m$ from a Dirichlet distribution ($Dir(\overrightarrow{\alpha})$), which determines topic assignment for words in that document. Then the topic assignment for each word placeholder $[m,n]$ is performed by sampling a particular topic $z_{m,n}$ from multinomial distribution $Mult(\overrightarrow{\vartheta}_m)$. And finally, a particular word $w_{m,n}$ is generated for the word placeholder $[m,n]$ by sampling from multinomial distribution $Mult(\overrightarrow{\varphi}_{z_{m,n}})$.
From the generative graphical model depicted in Figure 2.6, we can write the joint distribution of all known and hidden variables given the Dirichlet parameters as follows.

$$p(\overrightarrow{w}_m, \overrightarrow{z}_m, \overrightarrow{\vartheta}_m|\overrightarrow{\alpha}, \Phi) = \prod_{n=1}^{N_m} p(w_{m,n}|\overrightarrow{\varphi}_{z_{m,n}})p(z_{m,n}|\overrightarrow{\vartheta}_m)p(\overrightarrow{\vartheta}_m|\overrightarrow{\alpha})$$

And the likelihood of a document $\overrightarrow{w}_m$ is obtained by integrating over $\overrightarrow{\vartheta}_m$ and summing over $\overrightarrow{z}_m$ as follows.

human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Figure 2.5: Examples of Topics of LDA

$$p(\overrightarrow{w}_m|\overrightarrow{\alpha},\Phi) = \int p(\overrightarrow{\vartheta}_m|\overrightarrow{\alpha}) \cdot \prod_{n=1}^{N_m} p(w_{m,n}|\overrightarrow{\vartheta}_m,\Phi) d\overrightarrow{\vartheta}_m$$

Finally, the likelihood of the whole data collection $\mathcal{W} = \{\overrightarrow{w}_m\}_{m=1}^{M}$ is product of the likelihoods of all documents:

$$p(\mathcal{W}|\overrightarrow{\alpha},\Phi) = \prod_{m=1}^{M} p(\overrightarrow{w}_m|\overrightarrow{\alpha},\Phi) \tag{2.3}$$

2.4.2 Inference Problems

In order to use LDA, we need to compute the posterior distribution of the hidden variables given a document:

$$p(\overrightarrow{\vartheta},\overrightarrow{z}|\overrightarrow{w},\vec{\alpha},\Phi) = \frac{p(\overrightarrow{\vartheta},\overrightarrow{z},\overrightarrow{w}|\vec{\alpha},\Phi)}{p(\overrightarrow{w}|\vec{\alpha},\Phi)}$$

The normalization factor in the dominator can be expressed in terms of model parameters$(\vec{\vartheta},\vec{\alpha},\Phi)$ with the coupling of $\vec{\vartheta}$ and Φ in the summation over latent topics. That this factor is intractable to compute makes the exact inference for posterior distribution of LDA intractable to perform. Fortunately, a wide variety of approximation inference algorithms can be considered such as EM

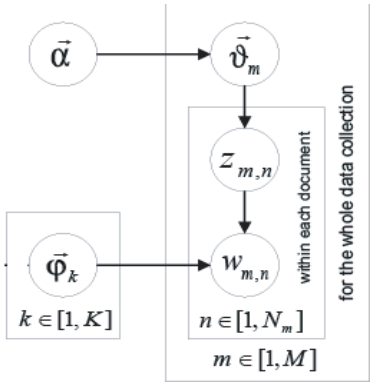


Figure 2.6: The generative process of LDA is represented by a graph, in which nodes are random variables and directed edges represents dependency.

Parameters and variables:
- M: the total number of documents to generate (const scalar)
- K: the number of (hidden/latent) topics /mixture components (const scalar)
- V: number of terms t in vocabulary (const scalar)
- $\overrightarrow{\alpha}$: Dirichlet parameters
- $\overrightarrow{\vartheta}_m$: topic distribution for document m
- $\Theta = \{\overrightarrow{\vartheta}_m\}_{m=1}^{M}$: a $M \times K$ matrix
- $\overrightarrow{\varphi}_k$: word distribution for topic k
- $\Phi = \{\overrightarrow{\varphi}_k\}_{k=1}^{K}$: a $K \times V$ matrix
- N_m: the length of document m, here modeled with a Poisson distribution with constant parameter ξ
- $z_{m,n}$: topic index of nth word in document m
- $w_{m,n}$: a particular word for word placeholder [m, n]

Table 2.1: Notations in LDA

variational algorithm [11] or sampling method [38]. Due to conjugate property of Dirichlet distribution and multinomial distribution, we are able to perform collapsed Gibbs sampling for LDA inference, in which we can integrate out $\vec{\vartheta}$ (collapsed) and keep only statistics of topic indicator (z). This property bring a lot of simplicities compared to Correlated Topic Model (CTM).

2.4.3 Model Fitting with Gibbs Sampling

Parameter estimation for LDA by directly and exactly maximizing the likelihood of the whole data collection in (2.3) is intractable. One solution is to use approximate estimation methods like Variational Methods [11], and Gibbs Sampling [35]. This section first give a brief introduction to Gibbs sampling methods, then presents its application for LDA.

Gibbs Sampling

Gibbs sampling is a special case of Markov-chain Monte Carlo (MCMC) [3] and often yields relatively simple algorithms for approximate inference in high-dimensional models such as LDA. Through the stationary behavior of a Markov chain, MCMC methods can emulate high-dimensional probability distributions $p(\overrightarrow{x})$. This means that one sample is generated for each transition in the chain after a stationary of the chain has been reached, which happens after a so-called "burn-in period" that eliminates the influence of initialization parameters. In Gibbs sampling, the dimensions x_i of the distribution are sampled alternately one at a time, conditioned on the values of all other dimensions, which we denote $\overrightarrow{x}_{\neg i}$. The algorithm works as follows:

1. Choose dimension i (by random or permutation)
2. Sample x_i from $p(x_i|\overrightarrow{x}_{\neg i})$

In order to build a Gibbs sampler, the full conditionals $p(x_i|\overrightarrow{x}_{\neg i})$ must be calculated using:

$$p(x_i|\overrightarrow{x}_{\neg i}) = \frac{p(\overrightarrow{x})}{\int p(\overrightarrow{x})dx_i} \text{ with } \overrightarrow{x} = \{x_i, \overrightarrow{x}_{\neg i}\}$$

For models that contain hidden variables $\overrightarrow{z}$, their posterior given the evidence, $p(\overrightarrow{z}, \overrightarrow{x})$ is a distribution commonly needed. The general formulation of a Gibbs sampler for such latent-variable models becomes:

$$p(z_i|\overrightarrow{z}_{\neg i}, \overrightarrow{x}) = \frac{p(\overrightarrow{z}, \overrightarrow{x})}{\int_z p(\overrightarrow{z}, \overrightarrow{x})dz_i}$$

The integral changes to a sum of discrete variables with a sufficient number of samples $\tilde{\overrightarrow{z}}_r, r \in [1, R]$, and the latent-variable posterior can be approximated using:

$$p(\overrightarrow{z}, \overrightarrow{x}) \approx \frac{1}{R}\sum_{r=1}^{R} \delta(\overrightarrow{z} - \tilde{\overrightarrow{z}}_r)$$

with the Kronecker delta $\delta(\overrightarrow{u}) = 1$ if $\overrightarrow{u} = 0; \text{and} 0$ otherwise.

Model Estimation for LDA

The first use of Gibbs Sampling for estimating LDA is reported in [35] and a more comprehensive description of this method is from the technical report [38]. One can refer to these papers for a better understanding of this sampling technique. Here, we only show the main idea and the most important formula that is used for topic sampling for words.

Let $\overrightarrow{w}$ and $\overrightarrow{z}$ be the vectors of all words and their topic assignment of the whole data collection W. Gibbs Sampling approach [35] is not explicitly representing Φ or ϑ as parameters to be estimated, but instead considering the posterior distribution over the assignments of words to topics, $P(\overrightarrow{z}|\overrightarrow{w})$. We then obtain estimates of Φ and Θ by using this posterior distribution. In order to

estimate the posterior distribution, Griffiths et. al. used the probability model for LDA with the addition of a Dirichlet prior on Φ. The complete probability model is as follows:

$$
\begin{gathered}
w_i|z_i, \Phi^{(z_i)} \sim \text{Mult}(\Phi^{(z_i)}) \\
\Phi \sim \text{Dirichlet}(\vec{\beta}) \\
z_i|\Theta^{(d_i)} \sim \text{Mult}(\Theta^{d_i}) \\
\Theta^{(d_i)} \sim \text{Dirichlet}(\vec{\alpha})
\end{gathered}
$$

Here, $\vec{\alpha}$ and $\vec{\beta}$ are hyper-parameters, specifying the nature of the priors on Θ and Φ. These hyper parameters could be vector-valued or scalar. The joint distribution of all variables given these parameters is $p(\overrightarrow{w}, \overrightarrow{z}, \Theta, \Phi|\vec{\alpha}, \vec{\beta})$. Because these priors are conjugate to the multinomial distributions Φ and Θ, we are able to compute the joint distribution $p(\overrightarrow{w}, \overrightarrow{z})$ by integrating out Φ and Θ.

Using this generative model, the topic assignment for a particular word can be calculated based on the current topic assignment of all the other word positions. More specifically, the topic assignment of a particular word t is sampled from the following multinomial distribution.

$$
p(z_i = k|\overrightarrow{z}_{\neg i}, \overrightarrow{w}) = \frac{n^{(t)}_{k,\neg i} + \beta_t}{[\sum_{v=1}^{V} n^{(v)}_k + \beta_v] - 1} \frac{n^{(k)}_{m,\neg i} + \alpha_k}{[\sum_{j=1}^{K} n^{(j)}_m + \alpha_j] - 1} \tag{2.4}
$$

where $n^{(t)}_{k,\neg i}$ is the number of times the word t is assigned to topic k except the current assignment; $\sum_{v=1}^{V} n^{(v)}_k - 1$ is the total number of words assigned to topic k except the current assignment; $n^{(k)}_{m,\neg i}$ is the number of words in document m assigned to topic k except the current assignment; and $\sum_{j=1}^{K} n^{(j)}_m - 1$ is the total number of words in document m except the current word t. In normal cases, Dirichlet parameters $\overrightarrow{\alpha}$, and $\overrightarrow{\beta}$ are symmetric, that is, all α_k ($k = 1..K$) are the same, and similarly for β_v ($v = 1..V$).

After finishing Gibbs Sampling, two matrices Φ and Θ are computed as follows.

$$
\varphi_{k,t} = \frac{n^{(t)}_k + \beta_t}{\sum_{v=1}^{V} n^{(v)}_k + \beta_v} \tag{2.5}
$$

$$
\vartheta_{m,k} = \frac{n^{(k)}_m + \alpha_k}{\sum_{j=1}^{K} n^{(j)}_m + \alpha_j} \tag{2.6}
$$

2.4.4 Posterior Inference with Gibbs Sampling

Given an estimated LDA model, we can now perform topic inference for unknown documents by a similar sampling procedure as previously [38]. A new document $\tilde{m}$ is a vector of words $\tilde{\vec{w}}_m$; our goal is to estimate the posterior distribution of topics $\tilde{\vec{z}}$ given the word vector $\tilde{\vec{w}}$ and the LDA model $L(\Theta, \Phi)$: $p(\vec{z}|\vec{w}, L) = p(\tilde{\vec{z}}, \tilde{\vec{w}}, \vec{w}, \vec{z})$. Here, $\vec{w}$ and $\vec{z}$ are vectors of all words and topic assignment of

the data collection upon which we estimate the LDA model. The similar reasoning is made to get the Gibbs sampling update as follows

$$p(\tilde{z}_i = k|\tilde{\vec{z}}_{\neg i}, \tilde{\vec{w}}; \vec{z}_{\neg i}, \vec{w}) = \frac{n_k^{(t)} + \tilde{n}_{k,\neg i}^{(t)} + \beta_t}{\left[\sum_{v=1}^{V} n_k^{(v)} + \tilde{n}_k^{(v)} + \beta_v\right] - 1} \frac{n_{\tilde{m},\neg i}^{(k)} + \alpha_k}{\left[\sum_{z=1}^{K} n_{\tilde{m}}^{(z)} + \alpha_z\right] - 1} \tag{2.7}$$

where the new variable $\tilde{n}_k^{(t)}$ counts the observation of term t and topic k in new documents. This equation gives an illustrative example of how Gibbs sampling works: high estimated word-topic association $n_k^{(t)}$ will dominate the multinomial masses in comparison with the contributions of $\tilde{n}_k^{(t)}$ and $n_{\tilde{m}}^{(t)}$, the masses of topic-word associations are propagated into document-topic associations [38].
After performing topic sampling, the topic distribution of new document $\tilde{m}$ is $\vec{\vartheta}_{\tilde{m}} = \{\vartheta_{\tilde{m},1}, ..., \vartheta_{\tilde{m},k}, ..., \vartheta_{\tilde{m},K}\}$ where each component is calculated as follows

$$\vartheta_{\tilde{m},k} = \frac{n_{\tilde{m}}^{(k)} + \alpha_k}{\sum_{z=1}^{K} n_{\tilde{m}}^{(z)} + \alpha_z} \tag{2.8}$$

2.5 Correlated Topic Model

Since its birth, Latent Dirichlet Allocation (LDA) has gained a lot of success in modeling data thanks to its simplicity, its ability to produce more interpretable topics compared to LSA, and less overfitting model compared to pLSA. There were a variety of LDA-like models proposed in the literature on the attempt to obtain richer semantic representation. There is, however, a trace-off between the richness of semantic representation and the complexity of the model.

Correlated Topic Model [10] (CTM) is the topic model that aims at capturing the correlation among topics, which is unable to obtained in LDA due to the application of Dirichlet distribution. One example is a document about genetic is related to disease more than x-ray astronomy. More examples of correlated topics of CTM is given in Figure 2.7. In order to encode the correlation among topics, logistic normal distribution has been introduced instead of Dirichlet distribution. Since logistic normal distribution is not conjugate with Multinomial distribution, model fitting and inference of CTM becomes more complicated in comparison with LDA.

2.5.1 Generative Model

Let $\{\mu, \Sigma\}$ be a K-dimensional mean and covariance matrix, and let topics $\boldsymbol{\beta}_{1:K}$ be K multinomial's over a fixed word vocabulary. The correlated topic model assumes that an N-word document arises from the following generative process:

1. Draw $\vec{\eta}|\{\vec{\mu}, \Sigma\}$ from Normal$(\vec{\mu}, \Sigma)$
2. For $n \in \{1, ..., N\}$:
    - Draw topic assignment $z_n|\vec{\eta}$ from $Mult(f(\vec{\eta}))$.

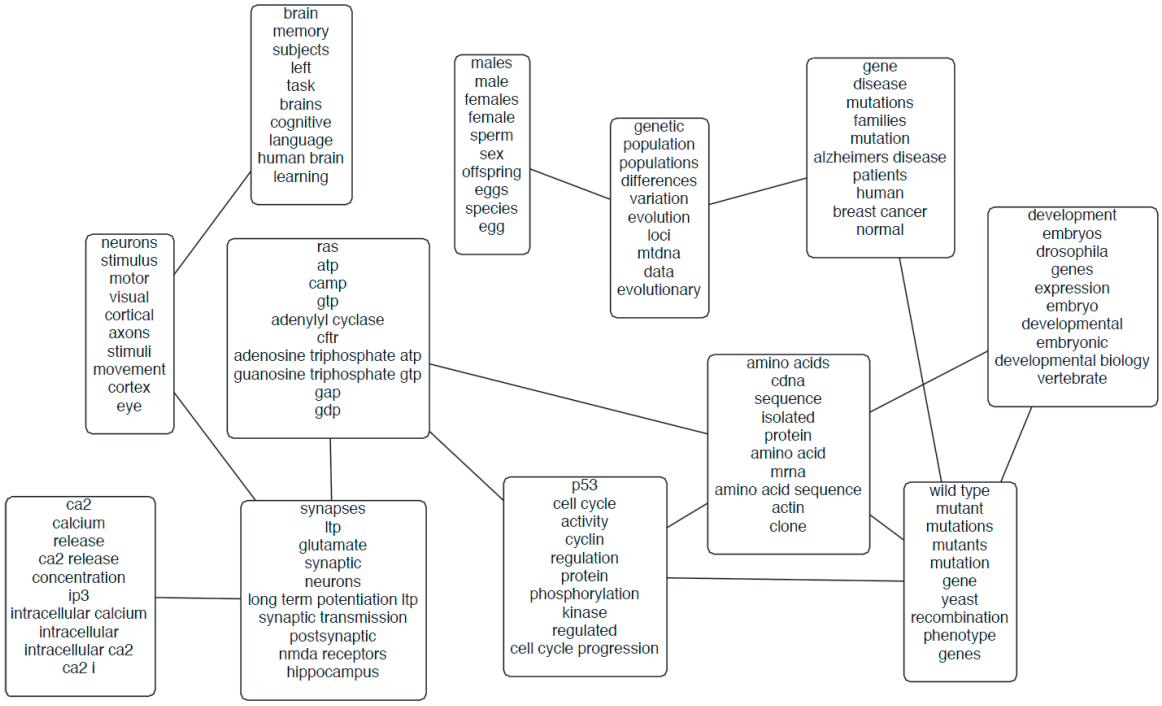


Figure 2.7: Example of Correlated Topics [10]

- Draw word $w_n|\{z_n, \boldsymbol{\beta}_{1:K}\}$ from $Mult(\vec{\beta}_{zn})$.

The key to the correlated topic model is the logistic normal distribution. Here, $f(\vec{\eta})$ is a (logistic) mapping from the normal distribution $\vec{\eta}$ to the simplex (multinomial distribution $z_n|\vec{\eta}$). That is, $f(\eta_i) = \exp \eta_i / \sum_{j=1}^{K} \exp \eta_j$. The two inference problems in Correlated Topic Model are described as follows:

- **Inference Problem**: Given a model $M = \{\boldsymbol{\beta}_{1:K}, \vec{\mu}, \Sigma\}$, and an unknown document $\vec{w}_{1:N}$ (E), we need to estimate the posterior distribution of the latent variables conditioned on the words of that document $p(\vec{\eta}_{1:K}, \vec{z}_{1:N}|\vec{w}_{1:N}, \boldsymbol{\beta}_{1:K}, \vec{\mu}, \Sigma)$. Exact inference is intractable due to these complicated modeling assumptions. As a result, a fast variational inference algorithm is used to approximate this posterior.

- **Model Fitting (Estimation Problem)**: Given a collection of documents, model parameters $\{\boldsymbol{\beta}_{1:K}, \vec{\mu}, \Sigma\}$ are estimated using a variant of expectation maximization algorithm, where the E-step is the per-document posterior inference problem stated above.

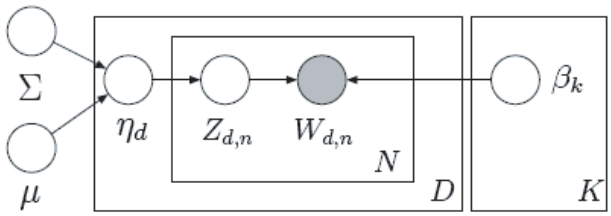


Figure 2.8: Graphical model representation of the Correlated Topic Model

2.5.2 Posterior Inference with Variational Method

Variational Method

Mean field variation method forms a factorized distribution of the latent variables, parameterized by free variables which are called the variational parameters. These parameters are fit so that the KL divergence between the approximate and true posterior is small.

We begin by estimating the lower bound of log likelihood of a document. Let E and H denote observed nodes, (evidences) and hidden nodes in the graphical model, and $q(H|E)$ is a variational distribution.

$$\begin{aligned}
\log p(E|M) &= \log \sum_{\{H\}} p(H,E|M) = \log \sum_{\{H\}} q(H|E) \frac{p(H,E|M)}{q(H|E)} \\
&\geq \sum_{\{H\}} q(H|E) \log \frac{p(H,E|M)}{q(H|E)} \text{ (Jensen inequality)} \\
&= \sum_{\{H\}} q(H|E) \log p(H,E|M) + \mathcal{H}(q)
\end{aligned}$$

Here, $\mathcal{H}(q)$ denotes the entropy of the variational distribution.

Variational Model for CTM

Based on the dependencies encoded in the graphical model, $p(H,E|M)$ can be calculated as follows:

$$\begin{aligned}
p(H,E|M) &= p(\vec{w}_{1:N}, \vec{z}_{1:N}, \vec{\eta}|\vec{\mu}, \Sigma, \boldsymbol{\beta}) \\
&= p(\vec{w}_{1:N}|\vec{z}_{1:N}, \boldsymbol{\beta}) \times p(\vec{z}_{1:N}|\vec{\eta}) \times p(\vec{\eta}|\vec{\mu}, \Sigma) \\
&= (\prod_{n=1}^{N} p(w_n|z_n, \boldsymbol{\beta}) p(z_n|\vec{\eta})) \times p(\vec{\eta}|\vec{\mu}, \Sigma)
\end{aligned}$$

Blei et al. [9] used the factorized variational distribution as follows:

$$q(\vec{\eta}_{1:K}, \vec{z}_{1:N}|\vec{\lambda}_{1:K}, \nu^2_{1:K}, \Phi_{1:N}) = \prod_{i=1}^{K} q(\eta_i|\lambda_i, \nu_i^2) \prod_{n=1}^{N} q(z_n|\vec{\phi}_n) \tag{2.9}$$

Here, the variational distributions of the discrete variables $\vec{z}_{1:N}$ are specified by the K-dimensional multinomial parameters $\Phi_{1:N}$. The variational distribution of the continuous variables $\vec{\eta}_{1:K}$ are K independent univariate Gaussian $\{\lambda_i, \nu_i^2\}$. Because the variational parameters are fit using a *single* document $\vec{w}_{1:N}$, there is no advantage in introducing a non-diagonal variational covariance matrix [10]. Replacing the variational distribution and $p(H, E|M)$ the above inequality, we obtain the lower bound of log likelihood of a document:

$$\begin{aligned} \log p(\vec{w}_{1:N}|\vec{\mu}, \Sigma, \boldsymbol{\beta}) &\geq E_q[\log p(\vec{\eta}|\vec{\mu}, \Sigma)] + \sum_{n=1}^{N}(E_q[\log p(z_n|\vec{\eta})] \\ &+ E_q[\log p(w_n|z_n, \boldsymbol{\beta})]) + \mathcal{H}(q) \end{aligned} \tag{2.10}$$

where the expectation is taken w.r.t the variational distribution of the latent variables. Note that $z_n|\vec{\eta}$ is drawn from $Mult(f(\vec{\eta}))$ and z_n is a topic indicator, which can be represented by a K dimensional vector $\vec{z}_n$ of which only one element has the value of 1 and the other elements have the value of zero.

$$\log p(z_n|\vec{\eta}) = \log \prod_{i=1}^{K} f(\eta_i)^{z_i} = \vec{\eta}^T \vec{z}_n - \log \sum_{i=1}^{K} \exp(\eta_i) \tag{2.11}$$

From this, we can represent the expected log probability of a topic assignment as follows:

$$E_q[\log p(z_n|\vec{\eta})] = E_q[\vec{\eta}^T \vec{z}_n] - E_q[\log(\sum_{i=1}^{K} \exp \eta_i)] \tag{2.12}$$

Set $\zeta x = \sum_{i=1}^{K} \exp \eta_i$ where ζ is another variational variable. Due to the fact that $\log \zeta x \leq x - 1 + \log \zeta$, we obtain

$$E_q(\log \zeta x) \leq \frac{1}{\zeta} \sum_{i=1}^{K} E_q[\exp(\eta_i)] - 1 + \log \zeta \tag{2.13}$$

The expectation $E_q[\exp(\eta_i)]$ is the mean of log normal distribution with mean and variance obtained from the variational parameters $\{\lambda_i, \nu_i^2\}$; thus, $E_q[\exp(\eta_i)] = \exp\{\lambda_i + \nu_i^2/2\}$ for $i \in \{1, ..., K\}$

Maximizing Likelihood Bound

This section describes the coordinate ascent optimization algorithm for the likelihood bound in equation 2.10 with respect to the variational parameters [10]. The first term of equation 2.10 is

$$E_q[\log p(\vec{\eta}|\vec{\mu},\Sigma)] = (1/2)\log|\Sigma^{-1}| - (K/2)\log 2\pi - (1/2)E_q[(\vec{\eta}-\vec{\mu})^T\Sigma^{-1}(\vec{\eta}-\vec{\mu})] \tag{2.14}$$

where

$$E_q[(\vec{\eta}-\vec{\mu})^T\Sigma^{-1}(\vec{\eta}-\vec{\mu})] = Tr(diag(\nu^2)\Sigma^{-1}) + (\vec{\lambda}-\vec{\mu})^T\Sigma^{-1}(\vec{\lambda}-\vec{\mu}) \tag{2.15}$$

The second term of equation 2.10 using the additional bound in equation 2.13 is

$$E_q[\log p(z_n|\vec{\eta})] = \sum_{i=1}^{K}\lambda_i\phi_{n,i} - \zeta^{-1}(\sum_{i=1}^{K}\exp\{\lambda_i+\nu_i^2/2\}) + 1 - \log\zeta \tag{2.16}$$

The third term of equation 2.10 is

$$E_q[\log p(w_n|z_n,\boldsymbol{\beta})] = \sum_{i=1}^{K}\phi_{n,i}\log\beta_{i,w_n} \tag{2.17}$$

Finally, the fourth term is the entropy of the variational distribution:

$$\sum_{i=1}^{K}\frac{1}{2}(\log\nu_i^2 + \log 2\pi + 1) - \sum_{n=1}^{N}\sum_{i=1}^{K}\phi_{n,i}\log\phi_{n,i} \tag{2.18}$$

In order to maximize the bound in equation 2.10 with respect to the variational parameters $\{\vec{\lambda}_{1:K}, \vec{\nu}_{1:K}, \Phi_{1:N}, \zeta\}$, Blei et. al. used a coordinate ascent algorithm, iteratively maximize the bound with respect to each parameter.
First, by maximizing equation 2.10 with respect to the variational ζ, the derivative with respect to ζ is

$$f'(\zeta) = N(\zeta^{-2}(\sum_{i=1}^{K}\exp\{\lambda_i+\nu_i^2\}) - \zeta^{-1}) \tag{2.19}$$

which has a maximum at

$$\hat{\zeta} = \sum_{i=1}^{K}\exp\{\lambda_i+\nu_i^2\} \tag{2.20}$$

Second, we maximize with respect to ϕ_n. This yields a maximum at

$$\hat{\phi}_{n,i} \propto \exp\{\lambda_i\}\beta_{i,w_n}, i \in \{1,...,K\}$$

Third, we maximize with respect to λ_i. Since equation 2.10 is intractable to analytic maximization, we use a conjugate gradient algorithm with derivative

$$dL/d\lambda = -\Sigma^{-1}(\vec{\lambda} - \vec{\mu}) + \sum_{n=1}^{N} \vec{\phi}_{n,1:K} - (N/\zeta)\exp\{\vec{\lambda} + \nu^2/2\} \tag{2.21}$$

Finally, we maximize with respect to ν_i^2. Again, there is no analytic solutioin. We use Newton's method for each coordinate, constrained such that $\nu_i > 0$:

$$dL/d\nu_i^2 = -\Sigma_{ii}^{-1}/2 - N/2\zeta\{\vec{\lambda} + \nu_i^2/2\} + 1/(2\nu_i^2) \tag{2.22}$$

Iterating between these optimizations defines a coordinate ascent algorithm on equation 2.10.

2.5.3 Model Fitting with Variational Expectation Maximization

We carry out parameter estimation for the correlated topic model by maximizing the likelihood of a corpus of documents as a function of the topics $\beta_{\mathbf{1:K}}$ and the multivariate Gaussian $(\mu, \mathbf{\Sigma})$. As in many latent variable models, we cannot compute the marginal likelihood of the data because of the latent structure that needs to be marginalized out. To address this issue, Blei et.al. use variational expectation-maximization (EM).
The objective function of variational EM is the likelihood bound given by summing equation 2.10 over the document collections $\{\mathbf{w_1}, ..., \mathbf{w_D}\}$

$$\log p(\mathbf{w}_{1:D}|\boldsymbol{\beta}_{1:K}, \boldsymbol{\mu}, \mathbf{\Sigma}) \geq \sum_{d=1}^{D} E_{q_d}[\log p(\boldsymbol{\eta}_d, z_d, \boldsymbol{w}_d|\boldsymbol{\mu}, \mathbf{\Sigma}, \boldsymbol{\beta}_{1:K})] + H(q_d) \tag{2.23}$$

The variational expectation-maximization

- E-step: given the data and current model parameters, approximate the posterior using variational method for each document in the corpus as described in the previous section. In other words, we maximize the bound with respect to the variational parameters by performing variational inference for each document as in Section 2.5.2.

- M-step: maximize the likelihood bound with respect to the model parameters. This amounts to maximum likelihood estimation of the topics and multivariate Gaussian using expected sufficient statistics, where the expectation is taken with respect to the variational distributions computed in the E-step

$$\hat{\boldsymbol{\beta}}_i \propto \sum_d \phi_{d,i}\mathbf{n_d}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{D}\sum_d \boldsymbol{\lambda}_d \tag{2.24}$$

$$\hat{\Sigma} = \frac{1}{D}\sum_{d} \mathbf{I}\boldsymbol{\nu}_{\mathbf{d}}^{\mathbf{2}} + (\boldsymbol{\lambda}_{\mathbf{d}} - \hat{\boldsymbol{\mu}})(\boldsymbol{\lambda}_{\mathbf{d}} - \hat{\boldsymbol{\mu}}) \tag{2.25}$$

where $\mathbf{n_d}$ is the vector of word counts for document d.

2.6 Conclusion

This chapter reviews some typical approaches to semantic representation with the focus on semantic space and topic models. We began with Latent Semantic Analysis, which is based on Singular Value Decomposition (SVD) to map words, documents into concept space with smaller dimension. We then presented brief introduction to some typical topic models that are Probabilistic Latent Semantic Analysis (pLSA), Latent Dirichlet Allocation (LDA) and Correlated Topic Models (CTM). Also, we present the link between pLSA and LSA as an example of the relationship between topic models and semantic space.

The important issue for statistical models in general and topic model in particular is inference, that is to derive posterior distributions given observations. While we are able to derive posterior distributions in some model (pLSA), it is intractable to do so in more complicated models (LDA, CTM). In this case, we need to apply approximation methods, which can be stochastic or deterministic. This chapter demonstrated two typical examples, which are Gibbs sampling (stochastic approach) and Variational EM (deterministic), to perform estimation and inference with LDA and CTM.

In the following chapters, we will demonstrate the usage of some topic models to obtain more semantic for text and image retrieval. In text retrieval, topic models are used to obtain richer representation to improve multiple applications in information retrieval including matching, ranking, clustering and labeling. In image retrieval, which we pay much of our attention to, topic models for image captions are considered to analyze the combinations of words to form scene, which helps automatically annotate images with topic-consistent labels for later retrieval.

Chapter 3

Web Search Clustering and Labeling with Hidden Topics

3.1 Introduction

It has been more than a decade since the first day Vietnam connected to the Internet in 1997. At that time, the Internet was served for a small group of people but became popular very quickly. In June 2006, VnExpress[1] - one of the most popular electronic newspapers in Vietnamese - appeared in the list of top 100 most accessed sites ranked by Alexa. It has been reported that the number of Internet users has reached 20 million [103], that accounts for about 23% of the population of Vietnam. For efficient access and exploration of such information on the Web, appropriate methods for searching, organizing and navigating through this enormous collection are of critical need. To this end, there were several emerging Web services such as [4, 92, 110], Web directory [120] and so on.

Although the performance of search engines is enhanced day by day, it is a tedious and time-consuming task to navigate through hundreds to hundred thousands of "snippets" returned from search engines. A study of search engine logs [46] argued that *"over half of users did not access result beyond the first page and more than three in four users did not go beyond viewing two pages"*. Since most of search engines display from about 10 to 20 results per page, a large number of users is unwilling to browse more than 30 results. One solution to manage that large result set is clustering. Like document clustering, search results clustering groups similar "search snippets" together based on their similarity; thus snippets relating to a certain topic will hopefully be placed in a single cluster. This can help users locate their information of interest and capture an overview of the retrieved results easily and quickly. In contrast to document clustering, search results clustering needs to be performed for each query request and be limited to the number of results returned from search engines [114] [73]. This adds extra-requirements to such kind of clustering [114]:

- Coherent Clustering: The clustering algorithm should group similar documents together. It should separate relevant documents from irrelevant ones.

[1] http://vnexpress.net

- Efficiently Browsing: Descriptive and meaningful labels should be provided so as to ease user navigation.

- Snippet-Tolerance: The method ought to produce high quality clusters even when it only has access to the snippets returned by the search engines, as most users are unwilling to wait while the system downloads the whole documents from the Web.

These requirements in general and the third one in particular introduce several challenges to clustering. In contrast to normal documents, these snippets are usually noisier, less topic-focused, and much shorter, that is, they contain from a dozen words to a few sentences. Consequently, they do not provide enough shared-context for good similarity measure.

There have been a lot of studies that attempted to overcome this data sparseness to achieve a better (semantic) similarity [79]. One solution is to utilize search engines to provide richer context of data [89, 13, 112]. For each pair of short texts, they use statistics on the results returned by a search engine (e.g., Google) in order to determine the similarity score. A disadvantage is that repeatedly querying search engines is quite time-consuming and not suitable for real-time applications. Another solution is to exploit online data repositories, such as Wikipedia[2] or Open Directory Project[3] as external knowledge sources [7, 91, 31]. In order to have benefits, the data sources should be in fine structures. Unfortunately, such types of data sources are not available or not rich enough in Vietnamese.

Inspired by the idea of using external data sources mentioned above, we present a general framework for clustering and labeling with hidden topics discovered from a large-scale data collection. This framework is able to deal with the shortness of snippets as well as provide better topic-oriented clustering results. The underlying idea is that we collect a large collection, which we call the "universal dataset", and then do topic estimation for it based on recent successful topic models such as pLSA [40], LDA [11]. It is worth reminding that the topic estimation needs to be done for a large corpus of long documents (the universal dataset) so that the topic model can be more precise. Once the topic model has been converged, it can be considered as one type of linguistic knowledge which captures the relationships between words. Based on the converged topic model, we are able to perform topic inference for (short) search results to obtain the intended topics. The topics are then combined with the original snippets to create expanded, richer representation. Exploiting one of the similarity measures (such as widely used cosine coefficient), we now can apply any of successful clustering methods based on similarity such as HAC, K-means [55] to cluster the enriched snippets. The main advantages of the framework include the following points:

- Reducing data sparseness: different word choices make snippets of the same topic less similar, hidden topics do make them more related than the original. Including hidden topics in measuring similarity helps both reduce the sparseness and make the data more topic-focused.

- Reducing data mismatching: some snippets sharing unimportant words, which could not removed completely in the phase of stop word removal, are likely close in similarity. By

[2]http://wikipedia.org
[3]Open Directory Project: http://www.dmoz.org

taking hidden topics into account, the pairwise similarities among such snippets are decreased in comparison with other pairs of snippet. As a result, this goes beyond the limitation of shallow matching based on word/lexicon.

- Providing informative and meaningful labels: traditional labeling methods assume that repetitious terms/phrases in a cluster are highly potential to be cluster labels. This is true but not enough. In this work, we use topic similarity between terms/phrases and the cluster as an important feature to determine the most suitable label, thus provide more descriptive labels.
- Adaptable to another languages: The framework is simple to implement. All we need is to collect a large-scale data collection to serve as the universal data and exploit the topics discovered from that dataset as additional knowledge in order to measure similarity between snippets. Since there are not many linguistic resources (Wordnet, Ontology, linguistic processing toolkits, etc.) in Vietnamese (and languages other than English), this framework is an economic and effective solution to the problem of Web search clustering and labeling in Vietnamese (and other Asian languages).
- Easy to reuse: The remarkable point of this framework is the hidden topic analysis of a large collection. This is totally unsupervised process but still takes time for estimation. However, once estimated, the topic model can be applied to more than one task which is not only clustering and labeling but also classification, contextual matching, etc.

Also, the framework is general enough to be applied to many clustering methods. In this chapter, we performed a careful evaluation for clustering search results in Vietnamese with the universal dataset containing several hundred megabytes of Wikipedia and VnExpress Web pages and achieved impressive clustering and labeling quality.

3.2 Related Work

Document clustering in general and web search results clustering in particular have become an active research topic during the past decade. Based on the relationship between clustering and labeling, we can classify solutions to the problem of web snippet clustering and labeling into two approaches: (1) perform snippets clustering and then labeling the generated clusters; or (2) generate significant phrases each of which is a cluster representative, snippets are then clustered based on these cluster representative. In the following, we will present our survey on the approaches to snippets clustering and labeling as well as the methods to deal with short texts, which is also one major part in our proposal.

3.2.1 Finding clusters first

Chen et al. developed a user interface that organizes Web search results into hierarchical categories [19]. To do that, they built a system that achieves the web pages returned by a search engine and classifies them into a known hierarchical structure such as LookSmart's web directory. Labels of the categories in the hierarchy are then used as labels of the clusters. Cutting et al., on the other

hand, considered clustering as a document browsing technique [20]. A large corpus is partitioned into clusters associated with their summaries which are frequent words in clusters. Based on the summaries, users navigate through the clusters of interest. These clusters are gathered together to form a sub-collection of the corpus. This sub-collection is then scattered *on-the-fly* into smaller clusters. The process of merging and re-clustering based on user navigation continues until the generated clusters become small enough. The most detailed (latest) clusters are represented by enumerating individual documents. The system built by [114] was the first post-retrieval system, which is designed especially for clustering web search results. The authors used novel Suffix Tree Clustering (STC) algorithm to group together documents sharing phrases (ordered sequence of words). This algorithm made use of special data structure called suffix tree - a kind of inverted index of phrases for a document collection. Using the constructed suffix tree, "base clusters" are created, each of which is associated with a phrase indexed in the tree. Base clusters with high degree of overlapping (in their document sets) are combined to generate final clusters. Shared phrases, which appear in many documents of one cluster, are used to convey the content of the documents in that cluster. According the authors, the advantage of this approach is the ability to obtain overlapping clusters in which a document can occur in more than one cluster. Chi-Lang Ngo used a method based on K-means and Tolerance Rough Set Model to generate overlapping clusters [73]. They then generated cluster labels by adapting an algorithm for n-gram generation to extract phrases from the contents of each cluster. They also hypothesized that phrases which are relatively infrequent in the whole collection but occurs frequently in clusters will be good candidate for cluster label. Unfortunately, they did not explain how to formalize this hypothesis in practice. Recently, Geraci et al. performed clustering by means of a fast version of the furthest-point-first algorithm for metric k-center clustering [32] . Cluster labels were obtained by combining intra-cluster and inter-cluster term extraction based on a variant of the information gain measure.

Supposed that clusters are somehow available, several researches aimed at assigning labels to these clusters. Given document clusters in hierarchy, Popescul et al. presented two methods of labeling document clusters [83]. The first one is to use a χ^2 test of significance to detect different word usage across categories in the hierarchy. The second method selects words which both occur frequently in a cluster and effectively discriminate the given cluster from the other clusters. Treeratpituk et al. labeled document hierarchy by exploiting a simple linear model to combine a phrase's features into a DScore [97] . They used features like DF (document frequency), TFIDF (term frequency, inverted document frequency), ranking of DF, the difference of these features at the parent and child node, and so on. The coefficients in the DScore model were learned and evaluated using DMOZ[4].

3.2.2 Finding labels first

The second approach to the problem of web search results clustering is from the idea of "*finding cluster description first*". Vivisimo is one of most successfully commercial clustering engine on the Web. Although most of the algorithm is kept unknown, their main idea is *"rather than form clusters and then figure out how to describe them, we only form well-described clusters in the first place"*.

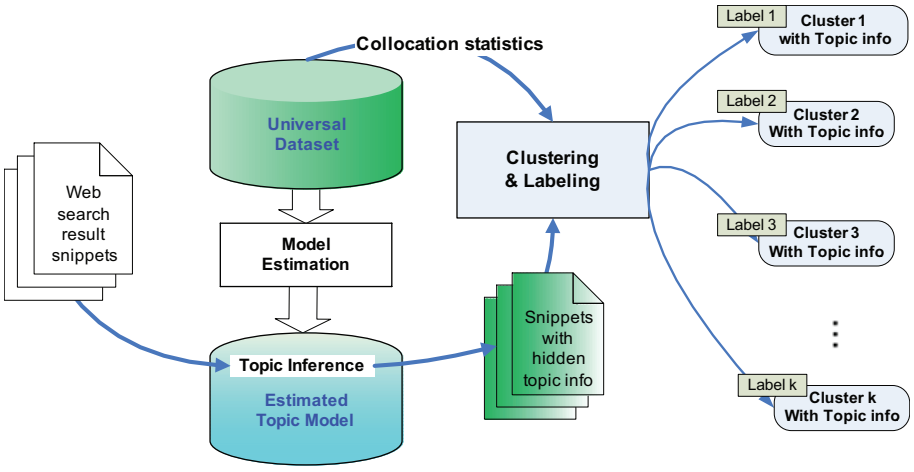
[4]Open Directory Project: http://www.dmoz.org

Toward this trend, Osinki tried to find out labels by a three-phase process [77]: (1) extract most frequent terms (words and phrases), (2) use Latent Semantic Indexing (LSI) [22] to approximate term-document matrix, forming concept-document matrix, and (3) select labels for each concept by matching previously extracted terms that are closest to a concept by standard cosine measure. Each concept become a cluster in their system, they later used Vector Space Model to determine snippets in clusters and merge clusters by calculating cluster scores. Zeng et al., on the other hand, extracted and ranked "salient phrases" as labels by using a regression model learned from human labeled training data [115]. The documents were assigned to relevant salient phrases to form cluster candidates, the final clusters were generated by merging these cluster candidates. Ferragina et al. selected (gaped) sentences by a merging and ranking process [30]. This process begins with words, then merges words in the same snippet and within a proximity window into a (longer) gaped sentence. Selected sentences are ranked and the low ranked sentences are discarded. All sentences, which have not been discarded, are marged with words in the similar manner. The process is repeated until no merge is possible or sentences are formed by 8 words (this can be customizable). The results of this process are sentences which form labels for "leaf clusters". These "leaf clusters" are then merged to achieve higher level clusters based on the sharing of "gaped sentences".

3.2.3 Dealing with short texts

Enriching short texts like snippets has achieved a lot of attentions recently. Banerjee et al. queried Wikipedia indexed collection for each snippet [7]. They then achieved titles of top Wikipedia pages as additional features for that snippet. Bollegala et al. proposed a robust semantic similarity measure that uses the information available on the Web to measure similarity between words or entities (Web search results) [13]. Not only based on the cooccurance of words in top ranked search results, they also extracted linguistic patterns to measure word semantic similarity. Cai et al. automatically extracted concepts from a large collection of text using pLSA [16]. They then exploited these concepts for classification with AdaBoost, a boosting technique which combines several weak, moderately accurate classifiers into one highly accurate classifier. Chi-Lang Ngo provided an enriched representation by exploiting Tolerance Rough Set Model (TRSM) [73]. With TRSM, a document is associated with a set of tolerance classes. In this context, a tolerance class represents a concept that is characterized by terms it contains. For example, {jaguar, OS, X} and {jaguar, cars} are two tolerance classes discovered from the collection of search results returned by Google for the query "jaguar". Ferragina et al. used two databases to improve extracted cluster labels [30]. The first one is an indexed collection of *anchor texts* extracted from more than 200 millions web pages. This knowledge base is used to enrich the content of the corresponding (poor) snippets. The second knowledge base is a ranking engine over the web directory DMOZ[5] which is freely available, controlled by humans and thus of high quality. The fundamental disadvantage of this method when applying to another languages other than English is the requirement of the human-built knowledge base (DMOZ). Recent research [43] used a concept thesaurus extracted from Wikipedia to enrich snippets in order to improve clustering performance.

[5]Open Directory Project: http://www.dmoz.org



(a) *Choosing an appropriate "universal dataset"*
(b) *Performing topic analysis for the universal dataset*
(c) *Finding collocations in the universal dataset*
(d) *Performing topic inference for search snippets*
(e) *Combining the original snippets with their hidden topics*
(f) *Building a clustering/labeling system on the enriched snippets*

Figure 3.1: The general framework of clustering Web search results with hidden topics

3.3 General Framework

In this section, we present the proposed framework that aims at building a clustering system with hidden topics from large-scale data collections. The framework is depicted in Figure 3.1 and consists of six major steps.

Among the six steps, choosing a right universal dataset (a) is probably the most important one. The universal dataset, as its name suggests, must be large and rich enough to cover a lot of words, concepts, and topics that are relevant to the domain of application. Moreover, the vocabulary of the dataset should be consistent with future unseen data that we will deal with. The universal dataset, however, is not necessary in a fine structure like Wikipedia in English or DMOZ. This implies the flexibility of the external data collection in use as well as of our framework. The dataset should also be pre-processed to exclude noise and non-relevant words, so the phase (b) can achieve good results. More details of (a) and (b) steps for a specific collection in Vietnamese will be discussed in the Section 3.4. Along with performing topic analysis, we also exploit the dataset to find collocations (c) (see Section 3.5.3). The collocations are then used for labeling clusters in (f). One noticeable point is that (a), (b) and (c) are performed offline and with no supervisor. The estimated model can be reused as a knowledge base to enrich documents for another tasks such as classification [79]. As a result, topic analysis is an economic, extensible and reusable solution to enrich documents in text/web mining.

In general, topic analysis for the universal dataset (b) can be performed by using one of the

Figure 3.2: Pipeline for data preprocessing and transformation

well-known hidden topic analysis models such as pLSA, LDA, CTM. It is worthy to notice that there is a trade-off between the richness of topic information and the time complexity of the system. LDA is chosen in this research because it is a more completely generative model than pLSA but not so complicated. With LDA, we are able to capture important semantic relationships in textual data but keeping time overhead acceptable. More details about topic analysis and LDA are given in Chapter 2.

The result of the step (b) is an estimated topic model including hidden topics and probability distributions of words given those topics (in the case of LDA). Based on this model and a collection of search results, we can perform topic inference (d) for those search snippets. Note that these short, sparse snippets are performed topic inference based on the model of the Universal Dataset, which has already been analyzed and converged. In another words, once the topics has been estimated in a huge dataset, they can be used as a background knowledge for adding more semantic to these search snippets. For each snippet, the output of (d) is the distribution of hidden topics in which high probabilities are assigned to its related topics. For instance, a snippet for the query "ma tr?n" (*matrix*) is probably related to topics such as "mathematic" or "movie". How to use this information as rich and useful features for clustering and labeling (e) (f) depends on the clustering algorithm.

This framework does not confine us to any clustering/labeling approaches. In this research, for simplicity, we applied the "*find clusters first*" approach and used Hierarchical Agglomerative Clustering (HAC) for the clustering step (see Section 3.5). However, other method such as *K-means* can be used for clustering. For K-means, we are able to choose initial centroids as snippets with emerging topics in the collection instead of random selection. Moreover, we can use the "*find cluster descriptions first*" approach to clustering and labeling in which the topic information is very helpful to achieve "*topic-oriented (significant) phrases*".

3.4 Hidden Topic Analysis of Vietnamese Dataset

3.4.1 Preprocessing and Transformation

Data preprocessing and transformation are necessary for data mining in general and for hidden topic analysis in particular. Since we target at topic analysis for Vietnamese, it is necessary to perform preprocessing in the consideration of specific characteristics of this language. The main steps for our preprocessing and transformation are described in the following and summarized in Figure 3.2.

Segmentation and Tokenization

This step includes sentence segmentation, sentence tokenization and word segmentation.

Sentence segmentation is to determine whether a "sentence delimiter" is really a sentence boundary. Like English, sentence delimiters in Vietnamese are full-stop, the exclamation mark and the question mark (.!?). The exclamation mark and the question mark do not really pose the problems. The critical element is the period: (1) the period can be a sentence-ending character (full stop); (2) the period can denote an abbreviation; (3) the period can used in some expressions like URL, Email, numbers, etc.; (4) in some cases, a period can assume both (1) and (2) functions. Given an input string, the results are sentences separated in different lines.

Sentence Tokenization is the process of detaching marks from words in a sentence. For example, we would like to detach "," or ":" from the previous words, which they are attached to.

Word Segmentation There is no clear word boundaries in Vietnamese since words are written in several syllables separated by white space (thus, we do not know which white space is actual word bounary and which is not). This leads to the task of word segmentation, i.e. segment a sentence into a sequence of words. Vietnamese word segmentation is a prerequisite for any further processing and text mining. Though being quite basic, it is not a trivial task because of the following ambiguities:

- Overlapping ambiguity: String *abc* are called overlapping ambiguity when both *ab* and *bc* are valid Vietnamese word. For example: "học sinh học sinh học" (Student studies biology) → "học sinh" (student) and "sinh học" (biology) are found in Vietnamese dictionary.
- Combination ambiguity: String *ab* were called combination ambiguity when *a*, *b* or *ab* are possible choices. For instance: "bàn là một dụng cụ" (Table is a tool) → "bàn" (Table), "bàn là" (iron), "là" (is) are found in Vietnamese dictionary.

For word segmentation, we used Conditional Random Fields approach to segment Vietnamese words [74] in which F1 measure is reported to be about 94%. After this step, sequences of syllables are joined to form words. For examples, a string like "công nghệ và cuộc sống" will become "công-nghệ và cuộc-sống" (*technology and life*).

Filters and Non Topic-Oriented Word Removal

After word segmentation, tokens, which can be word-tokens, number-tokens and so on, now are separated by white-space. Filters remove trivial tokens such as tokens for number, date/time, too-short tokens (of which length is less than 2 characters). Too short sentences, English sentences, or Vietnamese sentences without tones (The Vietnamese sometimes write Vietnamese text without tone) also should be filtered or manipulated in this phase.

Non topic-oriented words are those we consider to be trivial for topic analyzing process. These words can cause much noise and negative effects for our analysis. Here, we consider functional words, too rare or too common words as non topic-oriented words. The typical categories of

The universal dataset
After removing HTML tags, duplicate, too short or navigating pages, doing sentence and word segmentation: **size** $\approx$ 480M; |**docs**| $\approx$ 69,371
After filtering and removing non-topic oriented words:
size $\approx$ 101M, |**docs**| = 57,691
|**words**| = 10,296,286; |**vocabulary**| = 164,842
Topics assigned by humans in VnExpress Dataset
Society: Education, Entrance Examinations, Lifestyle of Youths
International: Analysis, Files, Lifestyles
Business: Business man, Stock, Integration,
Culture: Music, Fashion, Stage, Cinema
Sport: Football, Tennis
Life: Family, Health
Science: New Techniques, Natural Life, Psychology
Topics assigned by humans in Wikipedia Dataset
Mathematics and Natural Science: geology, zoology, chemistry, meteorology, biology, astronomy, mathematics, physics, etc.
Technologies and Applied Science: Nano technologies, biologic technology, information technology, Internet, computer science,etc.
Social Science and Philosophy: economics, education, archaeology, agriculture, anthropology, sociology, etc.
Culture & Arts: Music, tourism, movie industry, stage, literature, sports, etc.
Religion & Belief: Hinduism, muslim, buddhism, confucianism, atheistic, etc.

Table 3.1: Statistics of the Universal Dataset

functional words in Vietnamese includes classifier noun (similar to articles in English), conjunction (similar to *and*, *or* in English), numeral, pronoun, adjunct, and so on.

3.4.2 The Universal Dataset

Choosing a universal dataset is an important step in our proposal. In order to cover many useful topics, we used Nutch[6] to collect web pages from two huge resources in Vietnamese, which are Vnexpress[7] and Wikipedia[8]. VnExpress is one of the highest ranking e-newspapers in Vietnam, thus containing a large number of articles in many topics in daily life ranging from science, society and business, and many more. Vietnamese Wikipedia, on the other hand, is a huge online encyclopedia and contains thousands of articles which are either translated from English Wikipedia or written by Vietnamese contributors. Although Vietnamese Wikipedia is smaller than the English version, it contains useful articles in many academic domains such as mathematics, physics, etc. We combined two collections to form the universal dataset. The statistic information of the two collections is

[6] http://lucence.apache.org/nutch/
[7] http://vnexpress.net
[8] http://vi.wikipedia.org

Topic 3	Topic 4	Topic 7	Topic 9	Topic 10	Topic 15
hàm *(function)* **không gian** *(space)* **toán học** *(mathematics)* **định nghĩa** *(definition)* **phần tử** *(elements)* **bài toán** *(problem)* **lý thuyết** *(theory)* **tính toán** *(calculation)* **xác định** *(definite)* **định lý** *(theorem)* **phương trình** *(equation)* **ánh xạ** *(mapping)* **đại số** *(algebra)*	**phần mềm** *(software)* **chương trình** *(programs)* **Windows**$^{+}$ *(Windows)* **phiên bản** *(version)* **Microsoft**$^{+}$ *(Microsoft)* **hệ điều hành** *(operating system)* **ứng dụng** *(applications)* **cài đặt** *(install)* **giao diện** *(interface)* **trình duyệt** *(browser)* **internet*** *(internet)* **server*** *(server)*	**cầu thủ** *(football player)* **HLV**$^{\times}$ *(couch)* **đội bóng** *(football team)* **trận đấu** *(match)* **tiền đạo** *(offensive player)* **bàn thắng** *(goal)* **hậu vệ** *(defensive player)* **thủ môn** *(goalkeeper)* **chấn thương** *(injury)* **trọng tài** *(referee)* **đội hình** *(line-up)* **SLNA**$^{\times}$ *(SLNA*	**máy bay** *(aircraft)* **sân bay** *(airport)* **hàng không** *(airline)* **giao thông** *(traffic)* **tai nạn** *(accident)* **chuyến bay** *(flight)* **quốc tế** *(international)* **khách hàng** *(customer)* **Boeing**$^{+}$ **vận chuyển** *(deliver)* **phương tiện** *(vehicle)* **vận tải** *(transportation)* **đường sắt** *(railway)* **nhà ga** *(station)*	**tác giả** *(author)* **sách** *(book)* **nhà văn** *(writer)* **văn học**$^{\#}$ *(literature)* **truyện** *(stories)* **thơ** *(poem)* **tiểu thuyết** *(novel)* **xuất bản**$^{\#}$ *(publish)* **nhà thơ** *(poet)* **độc giả** *(readers)* **văn chương**$^{\#}$ *(literature)* **nhà xuất bản** *(publisher)* **phát hành**$^{\#}$ *(publish)*	**quốc hội** *(congress)* **tổng thống** *(president)* **dân chủ** *(democratic)* **hội đồng** *(council)* **chính quyền** *(goverment)* **nhân dân** *(people)* **cộng hòa** *(republican)* **nhà nước** *(state)* **hiến pháp** *(constitution)* **lãnh đạo** *(leadership)* **bầu cử** *(election)* **hội nghị** *(meeting)* **đảng cộng sản** *(communist party)*

Figure 3.3: Most likely words of some sample topics analyzed from the Universal Dataset ($K = 60$).

given in Table 3.1. Note that topics listed here are just for reference and not be taken into the topic analysis process.

3.4.3 Analysis Results and Outputs

After data preprocessing and transformation, we obtained 101MB data. We performed topic analysis for this processed dataset using GibbsLDA++ [9]. The parameters *Alpha* and *Beta* were set at $50/K$ and 0.1 respectively where K is the number of topics. The results of topic analysis with $K = 60$ and $K = 120$ are shown in Figure 3.3 and Figure 3.4. The complete results can be viewed online [10].

Figure 3.3 and 3.4 indicate that hidden topic analysis can model some linguistic phenomena such as synonyms or acronyms. For instance, the synonyms "văn học" (*literature*) and "văn chương" (*literature*) (Figure 3.3) are connected by the topic 10. The acronyms such as HLV (Huấn Luyện Viên - *couch*) and SLNA (Sông Lam Nghệ An - name of a famous football club) (Figure 3.3)

[9] GibbsLDA++:http://gibbslda.sourceforge.net

[10] http://jgibblda.sourceforge.net/vnwiki-120topics.txt

Topic 0	Topic 5	Topic 6	Topic 9	Topic 12	Topic 82
triết học *(phylosophy)* **khái niệm** *(concept)* **nhận thức** (*conceive*) **tri thức** *(knowledge)* **tồn tại** *(existence)* **học thuyết** *(doctrine)* **quan niệm** *(idea)* **bản chất** *(essence)* **quy luật** *(law)* **lý luận** *(reasoning)* **trường phái** *(school of thought)* **tranh luận** *(argument)*	**năng lượng** *(energy)* **điện** *(electricity)* **sóng** *(wave)* **ánh sáng** *(light)* **điện từ** *(electro-magnetic)* **dòng điện** *(electric current)* **vật liệu** *(material)* **tần số** *(frequency)* **tia** *(ray)* **từ trường** *(magnetic field)* **tín hiệu** *(signal)* **công suất** *(electricity power)*	**nhạc** *(music)* **album** *(album)* **ca sĩ** *(singer)* **ban nhạc** *(band)* **Pop** *(pop)* **Madonna** *(Madonna)* **âm nhạc** *(music)* **biểu diễn** *(performance)* **bảng xếp hạng** *(musical chart)* **ca khúc** *(song)* **lưu diễn** *(concert tour)* **Grammy** *(Grammy)* **MTV** *(MTV)* **ghi âm** *(recording)*	**cảnh sát** *(police)* **chết** *(dead)* **nạn nhân** *(victim)* **tai nạn** *(accident)* **phát hiện** *(discovery)* **xác** *(dead body)* **điều tra** *(investment)* **thi thể** *(dead body)* **bắt cóc** *(kidnap)* **súng** *(gun)* **an ninh** *(security)* **bị bắt** *(catch)* **mất tích** *(missing)*	**nguyên tố** *(elements)* **kim loại** *(metal)* **vật liệu** *(material)* **nhiệt độ** *(temperature)* **nhôm** *(aluminium)* **hợp chất** *(compound)* **hóa học** *(chemical)* **tinh thể** *(crystal)* **thép** *(steel)* **ô-xít** *(oxide)* **thủy tinh** *(glass)* **cấu trúc** *(structure)* **hợp kim** *(alloy)*	**nghệ thuật** *(art)* **tranh** *(picture)* **triển lãm** *(exhibit)* **họa sĩ**# *(painter)* **mỹ thuật** *(art)* **bảo tàng** *(museum)* **nghệ sĩ** *(artist)* **nhiếp ảnh** *(photograph)* **chân dung** *(portrait)* **hội họa** *(painting)* **bức ảnh** *(image)* **chất liệu** *(material)* **họa sỹ**# *(painter)*

Figure 3.4: Most likely words of some sample topics analyzed from the Universal Dataset ($K = 120$).

were correctly put in the topic of football (topic 7). Furthermore, hidden topic analysis is an economic solution to capture the semantic of new words (foreign words, named entities). For example, words such as "windows", "microsoft", "internet" or "server" (Figure 3.3), which are not covered by general Vietnamese dictionaries, were specified precisely in the domain of computer (topic 4). Figure 3.4 demonstrates another interesting situation in which the gap between two ways of writing the word *painter* in Vietnamese ("họa sĩ" - the correct spelling - and "họa sỹ" - the informal spelling but commonly accepted) were bridged by the topic about "painting, art" (topic 82). We will demonstrate how these relationships between words (via topics) can be used to provide good clustering in Section 3.6.

3.5 Clustering and Labeling with Hidden Topics

Clustering and labeling with Hidden Topics is summarized in Figure 3.5. Based on the estimated LDA model of the universal dataset (see Section 2.4), the collection of snippets is cleaned and performed topic analysis (see Section 2.4). This provides an enriched representation of the snippets. A specific clustering method is then applied on the enriched data. Here, we use Hierarchical Agglomerative Clustering (HAC) for the clustering phase. The generated clusters are shifted to

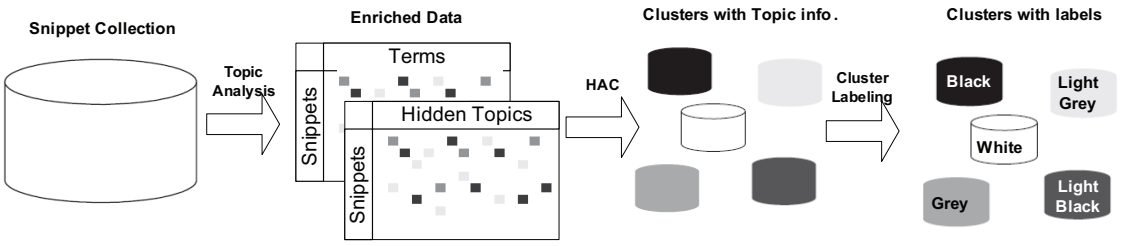


Figure 3.5: Clustering and labeling with hidden topics

the "Cluster Labeling Assignment" step which assigns descriptive labels to these clusters.

3.5.1 Topic Analysis and Similarity

Similarity between two snippets is fundamental to measure similarity between clusters. This section describes our representation of snippets with hidden topic information, which are inferred based on the topic model of the universal dataset, and presents a method to measure similarity between snippets.

For each snippet d_i, after topic analysis, we obtain the topic distribution $\vec{\vartheta}_{d_i} = \{\vartheta_{d_i,1}, ..., \vartheta_{d_i,k}, ..., \vartheta_{d_i,K}\}$. Upon this, we are able to build the topic vector $\vec{t}(d_i) = \{t_{d_i,1}, t_{d_i,2}, ..., t_{d_i,K}\}$ in which the weight $t_{d_i,k}$ of the topic kth is determined with regard to its probability $\vartheta_{d_i,k}$ as follows:

$$t_{d_i,k} = \begin{cases} \vartheta_{d_i,k} & \text{if } \vartheta_{d_i,i} \geq cutoff \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Note that K is the number of topics, and *cutoff* is the lower bound threshold for a topic to be considered important. Let V be the vocabulary of the snippet collection, the term vector of the snippet d_i has the following form:

$$\vec{w}(d_i) = \{w_1, ..., w_{|V|}\}$$

Here, the element w_i in the vector, which corresponds to the word/term i^{th} in V, is weighted by using some schema such as TF, TFxIDF. In order to calculate the similarity between 2 snippets d_i and d_j, the cosine measure is used for the topic-vectors as well as the term-vectors of 2 snippets.

$$sim_{di,dj}(topic - vectors) = \frac{\prod_{k=1}^{K} t_{i,k} \times t_{j,k}}{\sqrt{\sum_{k=1}^{K} t_{i,k}^2}\sqrt{\sum_{k=1}^{K} t_{j,k}^2}}$$

$$sim_{di,dj}(term - vectors) = \frac{\prod_{t=1}^{|V|} w_{i,t} \times w_{j,t}}{\sqrt{\sum_{t=1}^{|V|} w_{i,t}^2}\sqrt{\sum_{t=1}^{|V|} w_{j,t}^2}}$$

Combining two values, we obtain similarity between two snippets as follows:

$$sim(d_i, d_j) = \lambda \times sim(topic - vectors) + (1 - \lambda) \times sim(term - vectors) \tag{3.2}$$

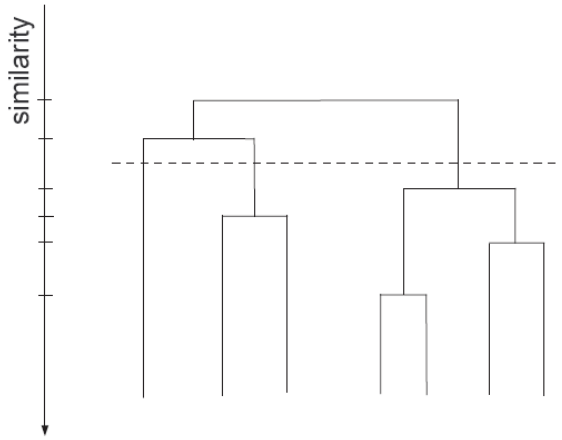


Figure 3.6: Dendrogram in Hierarchical Agglomerative Clustering.

Here, λ is a mixture constant. If $\lambda = 0$, the similarity is calculated without the support of hidden topics. If $\lambda = 1$, we measure the similarity between topic vectors of the two snippets without concerning words within them.

3.5.2 Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering [73] begins with each snippet as a separate cluster and merge them into successively larger clusters. Consequently, the algorithm builds a structure called ***dendogram*** - a tree illustrating the merging process and intermediate clusters. Cutting the tree at a given height will give a clustering at a selected precision.
Based on similarity between two snippets, similarity between two clusters A & B can be measured as follows:

- The minimum similarity between snippets of each cluster (also called ***complete linkage clustering***):
$$\min\{sim(x, y) : x \in A, y \in B\}$$
- The maximum similarity between snippets of each cluster (also called ***linkage clustering***):
$$\max\{sim(x, y) : x \in A, y \in B\}$$
- The mean similarity between snippets of each clusters (also called ***average linkage clustering***):
$$\frac{1}{|A||B|} \sum_{x \in A} \sum_{y \in B} sim(x, y)$$

Algorithm 1: Hierarchical Agglomerative Clustering

input : A collection $D = \{d_1, ..., d_n\}$, a cluster similarity measure Δ, a **merging threshold** ϵ
output: A set of cluster C

```
1 C = {initial clusters}                    /* each snippet forms an initial cluster */
2 repeat
3     (c1, c2) ← the pair of clusters which are most similar in C
4     if Δ(c1, c2) ≥ ε then
5         c3 ← c1 ∪ c2
6         add c3 into C
7         remove c1 and c2 from C
8     end
9 until can not merge                       /* can not find c1 and c2 with Δ(c1, c2) ≥ ε */
```

3.5.3 Cluster Label Assignment

Given a set of clusters for a snippet collection, our goal is to generate understandable semantic labels for each cluster. Let $C = \{c_1, c_2, ..., c_{|C|}\}$ be a set of $|C|$ clusters, we now state the problem of cluster labeling similarly to the "topic labeling problem" [67] as follows:

- **Definition 1**: A cluster $c \in C$ in a text collection has a set of "close" snippets, each cluster is characterized by an "*expected topic distribution*" ϑ_c, which is the average of topic distributions of all snippets in that cluster.
- **Definition 2**: A "*cluster label*" or a "*label*" l for a cluster $c \in C$ is a sequence of words which are semantically meaningful and best describe the latent meaning of c.
- **Definition 3 (Relevance Score)**: The relevance score of a label l to a cluster c, which is denoted as $s(l, c)$, measures the semantic similarity between the label and the cluster. Given that both l_1 and l_2 are meaningful label candidates, l_1 is a better label for c than l_2 if $s(l_1, c) > s(l_2, c)$

With these definitions, the problem of cluster labeling can be defined as follows: Let $L_i = \{l_{i1}, l_{i2}, ..., l_{im}\}$ be the set of label candidates for the cluster i^{th} in C. Our goal is to rank label candidates and select the most relevant labels for each cluster.

Label Candidate Generation

The first step in cluster label assignment is to generate phrases as label candidates. We extract two types of label candidates from the collection of search snippets. The first one includes unigrams (single words except for stop words); and the second one consists of meaningful bigrams (a meaningful phrase of two words - or bigram collocation). While extracting unigrams does not cause many issues, the difficulties lie in meaningful bigram extraction. The problem is how to know a bigram is a meaningful phrase or not. One method is based on "Hypothesis Testing" in which we extract phrases from n consecutive words (n-gram) and conduct statistical tests to know whether

Algorithm 2: Label Candidate Generation

input : Set of snippets $D = \{d_1, d_2, ..., d_n\}$; set of clusters $C = \{c_1, ..., c_{|C|}\}$
A frequency threshold **lblThreshold**; an "external collocation list" EC
A collocation threshold **colocThreshold**
output: Label candidates for clusters $LC = \{LC_1, LC_2, ..., LC_{|C|}\}$

```
1  extract and do statistics for all unigrams and bigrams from D
2  for each c_i ∈ C do
3      LC_i ← ∅
4      for each unigram u do
5          if frequency of u in c_i ≥ lblThreshold then
6              if u not a stop-word then  LC_i ← LC_i ∪ u
7          end
8      end
9      for each bigram b do
10         if frequency of b in c_i ≥ lblThreshold then
11             t ← t-score of b in D                    /* according to Eqn.3.3 */
12             if EC contains b or t ≥ colocThreshold then
13                 LC_i ← LC_i ∪ b
14             end
15         end
16     end
17 end
```

these words occurs together often than by chance. The null hypothesis usually assumes that "the words in a n-gram are independent", and different statistic testing methods have been proposed to test the significance of violating the null hypothesis. Process of generating label candidates for clusters are summarized in Algorithm 17. Although we only use n-grams ($n \leq 2$) as label candidates of clusters, the experiments show that this extraction is quite good for Vietnamese due to the fact that Vietnamese word segmentation (see 3.4) is able to also combine named entities (like "Hồ Chí Minh" - name of the famous former president in Vietnam) and some other frequently used combination (like "hệ điều hành" (*operating system*)). Longer phrases can be constructed by concatenating bigrams and unigrams.

A famous hypothesis testing method showing good performance on phrase extraction is Student's T-Test [65] [6]. Suppose that the sample is drawn from a normal distribution with mean μ, the test considers the difference between the observed and expected means, which are scaled by the variance of the data, and generates the probability of getting a sample of that mean and variance . We then compute the t statistic to specify the probability of getting our sample as follows:

$$t = \frac{x - \mu}{\sqrt{\frac{s^2}{N}}} \tag{3.3}$$

where x is the sample mean, s^2 is the sample variance, N is the sample size and μ is the mean of the distribution. We can reject the null hypothesis if the t statistic is large enough. By looking up

t-score	$C(w^1\ w^2)$	$C(w^1)$	$C(w^2)$	w^1	w^2
31.45	995	2130	1708	Điện thoại (*Phone*)	Di động (*Mobile*)
30.49	992	5223	4664	Thị trường (*Market*)	Chứng khóan (*Stock*)
21.24	469	2854	3713	Công nghệ (*Technology*)	Thông tin (*Information*)
19.1	365	2033	447	Vốn (*Capital*)	Điều lệ (*Charter*)
19.05	363	1278	860	Hội đồng (*Board*)	Quản trị (*Director*)
18.44	340	1492	2434	Đội tuyển (*Team*)	Quốc gia (*National*)
16.88	285	764	972	Vũ khí (*Weapon*)	Hạt nhân (*Nuclear*)
15.49	246	860	4005	Quản trị(*Administration*)	Kinh doanh (*Business*)
15.09	228	560	1021	Hệ điều hành (*OS*)	Windows (*Windows*)
13.82	191	409	1940	Nhà cung cấp (*Supplier*)	Dịch vụ (*Services*)
13.65	204	3432	3094	Trung tâm (*Center*)	Thương mại (*Trade*)
2.65	7	356	349	Khủng hoảng (*Crisis*)	Tiền tệ (*Money*)
2	4	238	407	Ứng cử viên (*Candidate*)	Nghiêm túc (*Serious*)
1.78	5	937	1373	Ủng hộ (*Support*)	Bà (*Her)*
1.73	3	1448	200	Chuẩn bị (*About to*)	Quảng bá (*Advertise*)
1.42	2	658	48	Người sử dụng (*User*)	Tra cứu (*Look up*)
1	1	1040	167	Đặc biệt (*Particularly*)	Yêu mến (*Love*)
1	1	3	2230	Nghiệm thu (*Check*)	Xây dựng (*Construction*)
0	3	5363	379	Chương trình (*Program*)	Cần thiết (*Necessary*)

Figure 3.7: Collocations and non-collocations specified from the uinversal dataset. Here, $C(s)$ is the frequency of the string s in the dataset, and s can be a word or a bigram. The bigrams with t value greater than 2.576 (the confident value of 99.5%) are collocations. All the collocations are extracted into a list called the "External Collocation List"

.

the table of the t distribution, we can find out how much confident for us to reject that hypothesis with a predefined threshold. Based on this t test, we now can examine whether a bigram is a collocation or not. Indeed, we find collocations in two situations (using JNSP[11]). The first one is to find collocations (in advance) from the universal dataset. This is performed (offline) to produce what we called the "External Collocation List". Examples of collocations and non-collocations drawn from the universal dataset is shown in Figure 3.7. The second situation is to determine collocations for each snippet collection to be clustered. Extracting collocations from the universal dataset is to obtain common used noun phrases such as "thị trường chứng khoán" (*stock market*) or "điện thoại di động" (*mobile phone*) which probably has not enough statistic information in the snippet collection to be verified as a collocation. On the other hand, finding collocations from the snippet collection is able to achieve specific phrases such as named entities which may not occur in the external collection.

Relevance Score

Given a set of clusters C and their label candidates, we need to measure the relevance between each cluster $c \in C$ and each label candidate l. In this work, we considered the relevance score as a

[11] http://jnsp.sourceforge.net/

linear combination of some specific features of l, c and other clusters in C as following

$$relevance(l, c, C) = \sum_{i=1}^{|F|} \alpha_i \times f_i(l, c, C) + \gamma \tag{3.4}$$

Here, α_i and γ are real-value parameters of the relevance score; $|F|$ is the number of features in use, and each feature $f_i(l, c, C)$ is a real-value function of the current label candidate l, current cluster c and the cluster set C. We considered five types of features ($|F| = 5$) for labeling clusters with hidden topics:

- **Intra-cluster topic similarity**: topic similarity between the label candidate l and the "*expected topic distribution*" of the cluster c (TSIM). If the label candidate l and the cluster c have some common topic with high probability, the two are likely related. We measure TSIM as the cosine of the two topic distribution vectors

$$TSIM(l, c) = \cos(\vec{\vartheta}_l, \vec{\vartheta}_c)$$

- **Cluster document frequency**: number of snippets in the cluster c containing the phrase l (CDF).

- **T-score**: the t-score of the phrase l in the snippet collection. If l is a unigram, its TSCORE is assigned to 2 (long phrases are preferred only if they are meaningful phrases).

- **Inter-cluster topic similarity**: the sum of intra-topic similarity of the label candidate l and other clusters

$$OTSIM(l, c, C) = \sum_{c' \in C, c' \neq c} TSIM(l, c')$$

- **Inter-cluster document frequency**: the sum of CDF in other clusters

$$OCDF(l, c, C) = \sum_{c' \in C, c' \neq c} CDF(l, c')$$

The label candidates of a cluster are sorted by its relevance in descending order and the most relevant candidates are then chosen as labels for the cluster. The inclusion of topic related features is a remarkable aspect of our proposal in comparison with previous work in cluster labeling (Section 3.2).

3.6 Experiments

3.6.1 Experimental Data

We evaluated clustering and labeling with hidden topics on two datasets:

Types	Query
General Terms	Bảo hiểm (*Insurance*), Công nghệ (*Technology*), Du lịch (*Tourism*), Hàng hóa (*Goods*), Thị trường (*Market*), Triển lãm (*Exhibition*), Đầu tư (*Investment*), Tài khoản (*Account*), Dân gian (*Folk*), Địa lý (*Geography*), Xây dựng (*Construct)*, Tết (*Tet Holiday*)
Ambiguous Terms	Táo (*Apple, Constipation, Kitchen God*), Chuột (*Mouse*), Cửa sổ (*Windows*), Không gian (*Space*), Ma trận (*Matrix*), Hoa hồng (*Commission, Rose*)
Named Entities	Hồ Chí Minh (*Ho Chi Minh*), Việt Nam (*Vietnam*)

Table 3.2: Queries submitted to Google

- *"Web dataset"* consists of *2357* snippets in *9* categories (business, culture & arts, health, laws, politics, science - education, life style & society, sports, technologies). These categories can be used as "key clusters" for later evaluation. Since this dataset contains the general categories, it can be used for evaluating the overall performance of clustering across domains as well as the quality of topic models (which topic model best describe the categories).

- *"Query dataset"* includes query collections. We collected this dataset by submitting *20* queries to Google and obtaining about *150* distinguished snippets in "key clusters" (but ignore "minor clusters") for each query (query collection). The list of search queries are listed in Table 3.2. The reason for choosing these queries is that they are likely to occur in multiple sub-topics, so we will benefit more from clustering search results. Since this dataset is sparse, it is much closer to realistic data that the search clustering system need to deal with. We used "key clusters" in each query collection to evaluate both clustering and labeling with hidden topics.

3.6.2 Evaluation

Clustering Evaluation

For evaluation, we need to compare the "generated clusters" with the "key clusters". To do that, we used BCURED scoring method [5], which originally exploited for evaluating entity resolution but also used for clustering evaluation [13]. This scoring algorithm models the accuracy of the system on a per-document basis and then build a more global score. For a document i, the precision and recall with respect to that document are calculated as follows:

$$P_i = \frac{\text{number of correct documents in the output cluster containing } document_i}{\text{number of documents in the output cluster containing } document_i}$$

$$R_i = \frac{\text{number of correct documents in the output cluster containing } document_i}{\text{number of documents in the key cluster containing } document_i}$$

Here, given a document i, the document j is correct if it is in the same key cluster as the

document i. The final precision and recall numbers are computed by the following two formulae:

$$FinalPrecision = \sum_{i=1}^{N} 1/N \times P_i$$

$$FinalRecall = \sum_{i=1}^{N} 1/N \times R_i$$

Usually, precision and recall are not used separately, but combined into F_β measure as following:

$$F_\beta = (1 + \beta^2) \times (precision \times recall)/(\beta^2 \times precision + recall) \tag{3.5}$$

For clustering evaluation, we used $F_{0.5}$ (or $\beta = 0.5$) to weight precision twice as much as recall. This is because we are willing to have average-size clusters but high precision than merging them into a large cluster for higher recall but low precision (thus, low coherence within clusters).

Labeling Evaluation

We performed label candidate generation for fixed "key clusters" in the "Query dataset". After this step, we had a list of label candidates for each "key cluster". We manually assigned "1" to appropriate labels and "0" to inappropriate ones. These scores were used for estimating parameters for the relevance score as well as for evaluation. As mentioned earlier, the label assignment is to rank label candidates for each cluster using relevance score and select the first-rank label. So, we measured the quality of the relevance score (or the ranking quality) by calculating precision (P) at top N label candidates in the generated ranking list:

$$P@N = \frac{\text{Number of correct label candidates}}{N} \tag{3.6}$$

Here, correct label candidates of a given cluster are the ones with the score of "1". In the following experiments, we use P@5, P@10, P@20 for evaluating our labeling method.

3.6.3 Experimental Settings

We conducted topic analysis for the Universal dataset using Latent Dirichllet Allocation with different number of topics (K=20, 60, 80, 100, 120, 160, 180 topics). The topic models are exploited for experiments hereafter. In the following experiments, we refer to clustering (using HAC) without hidden topics as baseline and clustering (using HAC) with K-topic model (K = 20, 60, etc.) as HTK.

The default parameters are specified in Table 3.3. These default parameters are basically unchanged in our experiments except for lambda which is changed in one specific experiment. The other parameters are changed more often, such as the merging threshold ϵ for clustering (see Algorithm 1), the number of hidden topics (K) for the Universal dataset. The parameters of relevance score for labeling, on the other hand, is learned from the "Query Dataset" (see Section 7.4.3). By keeping some parameters unchanged and varying others, we measured the influence of the main parameters on the clustering and labeling performance.

Parameters	**Values**	**Explanation**
Clustering Parameters		
Term weighting method	TF	Term Frequency
Lambda	0.35	Mixture constant in the similarity formula between two snippets (Equation 3.2).
Cluster similarity	Average Linkage	The mean similarity between elements of each clusters
Frequency threshold	30%	Terms/topics occur more frequent than this rate will be cutoff
Rare threshold	2 or 6	Terms occur less than this threshold will be removed. This threshold is set to 6 for "Web dataset" and to 2 for "Query collections".
Topic Cutoff	0.02	Topics with probability less than this value will not be used for enriching snippets
Labeling Parameters		
Collocation Threshold	2	A bigram with t score calculated in a snippet collection larger than this value is probably used as a label candidate. This is set by looking up the t-score table (for infinite degree of freedom and the confidence value of 97.5%)
Label threshold	2	Phrases with the frequency (in a cluster) less than this value will not be chosen as label candidates for that cluster

Table 3.3: Default parameters for clustering and labeling with hidden topics. The parameters are basically set as in the following table. Note that the rare word threshold is set differently for "Web dataset" and "Query collections" (in "Query dataset". This is because "Web dataset" is much larger than any "Query collection" and removing rare words can help to reduce the computational time.

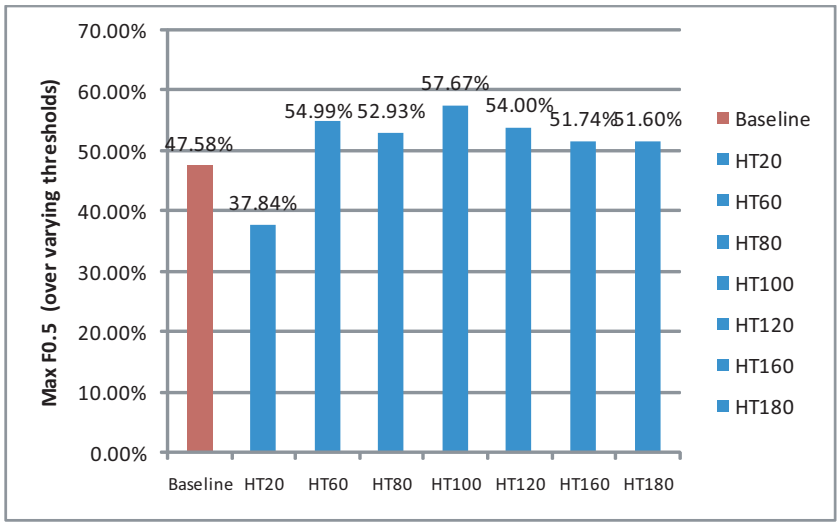


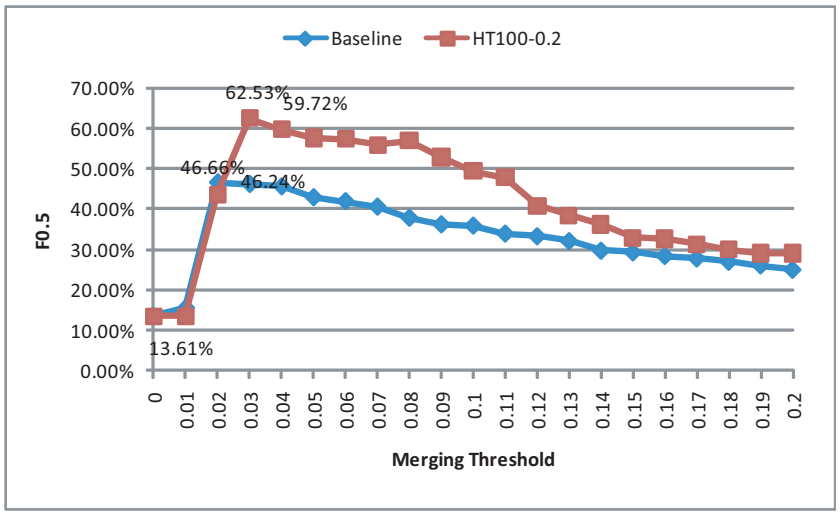
Figure 3.8: Performance of clustering using HAC (in baseline) and HAC with different topic models in Web dataset. For each clustering setting (without or with hidden topic models), we changed merging threshold and obtained the maximum F0.5 for comparison

3.6.4 Experimental Results and Analysis

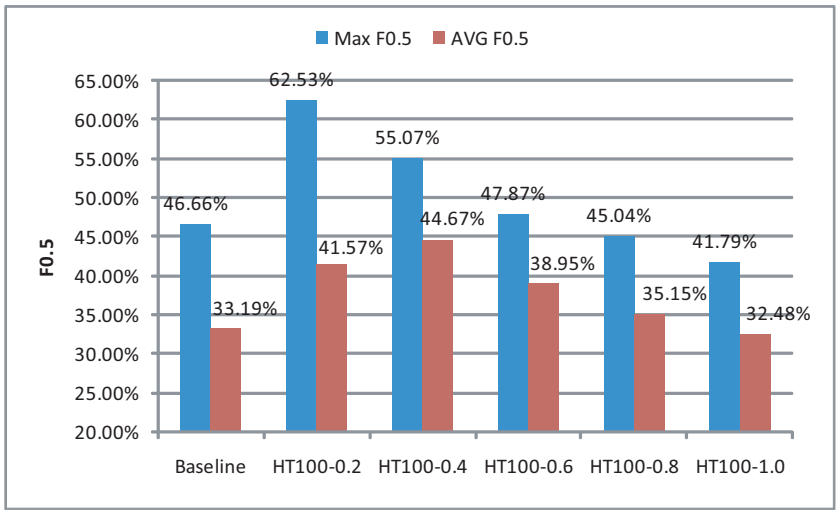
Clustering Performance

The comparison between baseline and HTK (K = 20, 60, 80, etc.) in the "Web dataset" is demonstrated in Figure 3.8. Using the categories of the dataset as "key clusters", we evaluated clustering performance with precision, recall, and F0.5 as described in the previous section. By taking the maximum value of F0.5 (among different merging thresholds), we compare the performance of baseline and HTK (K = 20, 60, 80, etc.) in Figure 3.8 . As depicted in the figure, clustering with hidden topics in most cases (other than 20-topic model) improve clustering performance. The bad performance of HT20 (9.74% worse than in the baseline) indicates that the number of topics for analysis should be suitable to reflect the topics in the Universal Dataset. Once the number of topics is large enough (like larger than 60 topics), the F0.5 is quite stable. It can also be observed that the 100-topic model best describes these general categories. As a result, $K \approx 100$ is probably the suitable number of topics for the Universal dataset.

We showed the results of the baseline and clustering using 100-topic model with lambda of 0.2 (HT100-0.2) in Figure 3.9 (a). From the figure, we can see that HT100-0.2 can provide significant improvement over the baseline. The maximum value of F0.5 in HT100-0.2 is 62.52% which is nearly 16% better than the baseline. When merging threshold is zero, all the snippets are merged into one cluster. That explains why HT100-0.2 and the baseline have the same starting value of F0.5. In addition, the inclusion of hidden topics increases similarity among snippets. As a result, when merging threshold is small, HT100-0.2 does not show an advantage over the baseline. When merging threshold is large enough, on the other hand, we can always obtain better results with



(a)



(b)

Figure 3.9: Baseline vs. HT100 in "Web dataset": (a) Baseline vs. HT100 and lambda=0.2 (HT100-0.2); (b)Merging threshold is varied from 0 to 0.2 like in (a). We compared the maximum and average values of F0.5 among clustering with different settings. Note that HT100-X (X is from 0.2 to 1) means clustering with 100 hidden topic model and lambda=X)

	AVG Max F0.5	AVG Precision	AVG Recall
Baseline (HAC)	**65.35%**	**76.86%**	**45.77%**
HT20	62.26%	74.49%	39.97%
HT60	72.72%	80.41%	54.31%
HT80	**73.60%**	**82.76%**	**53.58%**
HT100	72.58%	81.56%	53.90%
HT120	72.19%	81.25%	52.62%
HT160	72.95%	82.07%	51.68%
HT180	72.41%	81.57%	53.45%

Table 3.4: Baseline vs. clustering with different topic models in the "query dataset": For each clustering setting, the maximum value of F0.5 for each query collection is obtained. We then average these maximum values across query collections for comparing clustering settings

HT100-0.2.

In order to evaluate the influence of lambda in clustering performance, we conducted similar experiments to the one in Figure 3.9 (a) but with different lambda (0.2 to 1.0). The maximum values and average values of F0.5 (when merging threshold is changed from 0 to 0.2) were obtained for comparison in Figure 3.9 (b). As you can see from the figure, HT100-0.2 (lambda=0.2) and HT100-0.4 (lambda = 0.4) provides the most significant improvements. This means lambda should be chosen from 0.2 to 0.4.

Since the "Web dataset" is large and much more condensed than real search results, the above evaluation cannot give us a closer look at the performance of the real system. For this reason, we evaluated clustering performance using "Query dataset" which are collected from search results for some sample queries. For each query collection in the dataset, we conducted 8 experiments (clustering without hidden topics (the baseline) and with 7 different topic models). Taking the maximum F0.5 (and the corresponding precision and recall), we averaged these measures of the same experiment across query collections and summarized in Table 3.4 and Figure 3.10. According to the table, HT20 is still fail to provide an improvement (3.09% worse than the baseline) but the situation is not as bad as in "Web dataset" (9.09% worse than the baseline). Clustering with hidden topic models (other than HT20) provides significant improvements in both precision and recall. F0.5 reaches its peak in HT80 with 8.31% better than the baseline. Like in the "Web dataset", the value of F0.5 changes slightly over different hidden topic models. This supports the above observation that clustering with hidden topics outperforms the baseline when the number of hidden topics is large enough.

Detailed Analysis

We considered two cases in which hidden topics can be helpful toward clustering/labeling. The first one is the diverse of word choices in the same domain (also the sparseness of snippets). This is not only caused by the large number of words in one domain, but also by a variety of linguistic phenomena such as synonyms, acronyms, new words and words originating from foreign languages

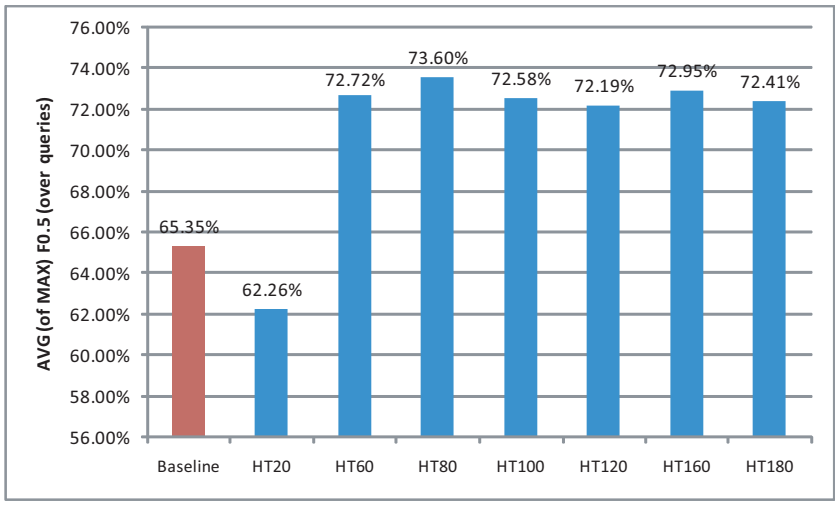
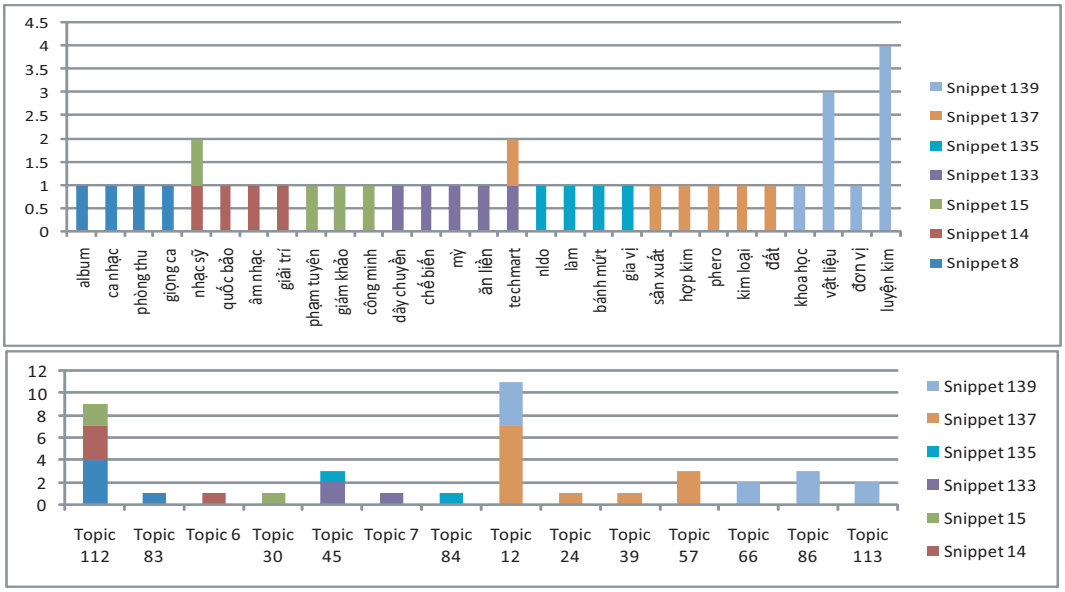


Figure 3.10: Baseline and clustering with different topic models on the query dataset

which are probably not covered by dictionaries, and different writing ways like "color" and "colour". As described in 3.4, hidden topics from the universal dataset can help us to bridge the semantic gap between these words. As a result, when taking hidden topics into account, the snippets in the same domain but with different word choice can be more similar. The second case is the existence of trivial words but with high frequencies. Although we eliminate stop words before clustering, it is impossible to totally get rid of them.

To better understand the reasons why our proposal works better than the baseline, we analyze one example (Figure 3.11) to see how hidden topics can be used to reduce data sparseness and mismatching. The figure reveals that snippet 133 and snippet 135 are about "food industry" but have no term in common. Similarly, snippet 137 and snippet 139 should be in the cluster of "material production" but share no term. Snippet 8, snippet 14, snippet 15 about "music activities" share only one term "nh?c si" (*musician*) and not close enough for good clustering. This is due to different word choices or the sparseness of the snippets. On the other hand, although snippet 133 and snippet 137 are in totally different topics - the first one is about "food industry" while the second one is about "material production", they share the term "techmart"- the name of the website from which two snippets extracted - which is a trivial word here. Since the term-based similarity only makes use of frequencies, and treats words equally, it does not reflect contextual similarity among the snippets. By taking topics into account, snippet 133 and snippet 137 (bridged by the topic 45) are closer in similarity. The same effect happens to the pair of snippet 137 and snippet 138 (bridged by the topic 12), and the triple of snippet 8, snippet 14 and snippet 15 (bridged by the topic 112). Snippet 133 and snippet 137, however, have no topic in common. As a result, the similarity between snippet them decreases in relative to the other pairs in the collection.



Preferred Clusters:
Music Activities: Snippet 8, Snippet 14, Snippet 15
Food Industry: Snippet 133, Snippet 135
Material Production: Snippet 137, Snippet 139
Shared words across snippets:
Nhạc sỹ (musician); Techmart (name of a multiple field website)
Shared topics across snippets:
Topic 112: hát (*sing*), ca sĩ (*singer*), nhạc (*music*), nhạc sĩ (*musician)*, bài hát (*song*), …
Topic 45: món(*dish*), ngon (*delicious*), bánh (*cake*), nhà hàng (*restaurant*), nấu (*cook*), …
Topic 12: nguyên tố (*elements*), kim loại (*metal*), sắt (*iron*), nhiệt độ (*temprature*), vật liệu (*material*), …

Figure 3.11: Illustration of the important contributions of hidden topics toward achieving better clustering/labeling

Labeling Performance

As mentioned earlier, the query dataset consists of several query collections, each of which include snippets returned by Google for a specific query. We manually partitioned each query collection into "key clusters". We then fixed these "key clusters" and generated "label candidates" for each of them. We also associated each "key cluster" with a list of scored label candidates (label candidates are assigned "1" if appropriate and "0" otherwise). Based on these specified clusters and their scored label candidates, we used "linear regression" to learn parameters for relevance score. To do that, we split the query dataset into two parts: (1) The testing data containing query collections of 4 queries {"tài khỏan" (*account*), "táo" (*apple*), chuột (*mouse*) and "ma trận" (*matrix*)}; (2) The training data containing the rest of query collections. Some statistics about the training and testing sets are provided in Table 3.5.

	#Queries	#Clusters	#Label Candidates
Testing data	4	27	797
Training data	16	119	3113

Table 3.5: Testing and training data for cluster labeling

The training data was put into the module *linear regression* of Weka[12] to learn parameters for relevance score. We tested two set of features: (1) the full set containing all five feature types as described in the Section 6; and (2) the partial set which exclude features associated with topics of the universal dataset. After learning process, we achieved the relevance scores as shown in the following:

- Learning with the full set of features: Relevance Score with the 120-topic model of the Universal Dataset (RS-HT120)

$$\begin{aligned} \text{RS-HT120} &= 0.4963 \times TSIM + 0.5903 \times CDF \\ &- 0.0755 \times TSCORE - 0.3312 \times OTSIM \\ &- 0.064 \times OCDF - 0.2722 \end{aligned}$$

- Learning with the partial set of features: Relevance Score without Hidden Topics (RS-base)

$$\begin{aligned} \text{RS-base} &= 0.6389 \times CDF - 0.0866 \times TSCORE \\ &- 0.4177 \times OCDF + 0.891 \end{aligned}$$

As we can see from the formula of RS-HT120, TSIM is the second important feature after the most significant one - CDF. The inter-cluster document frequency (OCDF) is quite important in RS-base (with the weight absolute of 0.4177) but less important than inter-cluster topic similarity (OTSIM) in RS-HT120. In both relevance scores, TSCORE does not have much effect on ranking label candidates.

Based on two relevance scores, we ranked label candidates in "key clusters" in the testing data. We then compared P@5, P@10, and P@20 of two scores in Figure 3.12. As observable in the figure, labeling with hidden topics can improve nearly 10% precision on average in the testing dataset. This showed the effective of hidden topics in label assignment.

Figure 3.13 shows the difference between labeling without and with hidden topics for some "key clusters" in the testing dataset. For the same cluster "điện thoại" (*mobile phone*) of the query "tài khoản" (*account*), 4 out of 5 label candidates in labeling with RS-HT120 are related to "phone" while there are only 3 good candidates out of 5 in labeling with RS-base (the first and fifth ones are inappropriate). The same situations occur in the other "key clusters" of the queries "chuột" (*mouse*), "táo" (*apple*) and "ma tr?n" (*matrix*). Moreover, better ranking was obtained in labeling with RS-HT120. It can be observed that the first ranking positions of the cluster "điện thoại"

[12]http://www.cs.waikato.ac.nz/ml/weka/

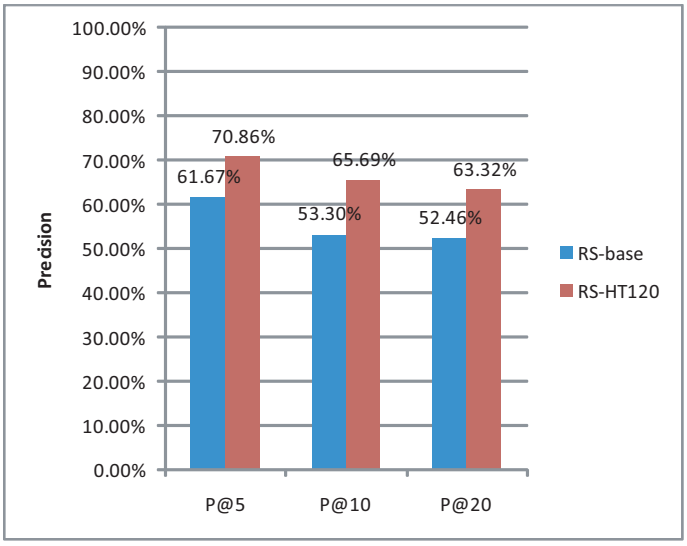


Figure 3.12: Comparison of the baseline (labeling without hidden topics) and labeling with 120 topics in the testing collection.

(*mobile phone*) (of the query "tài khoản" (*account*)) and the cluster "y tế" (*health services*) (of the cluster "chuột" *mouse*) in labeling with RS-base are "tiền" (*money*) and "dùng" (*take*) repectively which are not as much related to the content of the clusters as "tài khoản điện thoại" (*phone account*) and "thuốc" (*medicine*) in labeling with RS-HT120.

Computational Time Analysis

We compared the computational time between the baseline and clustering and labeling with HT120 in Figure 3.14. Since topic estimation of the Universal dataset is conducted offline, the phase, which requires online computation, is the topic inference for snippets. However, it seems to be acceptable when the number of snippets is around 200 snippets - the default number of snippets to be clustered in Vivisimo [102]. Additionally, using hidden topics enables us to remove more rare words than without hidden topics. The point is rare words, for example ones occurring only twice in the snippet collection, sometimes play an important role in connecting snippets. Suppose that we can divide a set of snippets about "movie" into two separated parts: those contains the word "actor" and those includes "director". If we have two snippets in two parts containing the same word such as "movie" which occurs only two times, we can join two parts into one coherent cluster. However, using hidden topics, you can remove such rare words without losing that connection because they all share the topic about "movie". This leads to significant reduction in the size of term vectors; and an improvement is obtained in computational time.

Query/Cluster	Labeling without RS-base	Labeling with RS-HT120
Tài khoản/Điện thoại *(Account/Mobile phone)*	**Tiền** *(Money)* **Tài-khoản điện-thoại** *(Phone Account)* **Tài-khoản di-động** *(Mobile Account)* **Điện-thoại di-động** *(Mobile Phone)* **Việt-nam** *(Vietnam)*	**Tài-khoản điện-thoại** *(Phone Account)* **Tiền** *(Money)* **Tài-khoản di-động** *(Mobile Account)* **Điện-thoại di-động** *(Mobile Phone)* **SIM** *(SIM card)*
Chuột/Y tế *("Mouse" or in "Cramp"/Health Services)*	**Dùng** *(Take)* **Thuốc** *(Medicine)* **Bệnh** *(Disease)* **Chữa** *(Cure)* **Loại** *(Type)*	**Thuốc** *(Medicine)* **Dùng** *(Take)* **Chữa** *(Cure)* **Bệnh** *(Disease)* **Chứng chuột-rút** *(The Cramp Trouble)*
Táo/Đồ ăn *("Apple" or "Name of a company", .../Food)*	**Bánh táo-nướng** *(Baked Apple Cake)* **Trái-táo** *(a fruit of Apple)* **Thử một-số** *(Try some)* **Thay-vì ăn** *(Insead of eating)* **Ăn bánh** *(Eating cake)*	**Bánh táo-nướng** *(Baked Apple Cake)* **Trái-táo** *(a fruit of Apple)* **Bột** *(Flour)* **Ăn bánh** *(Eating cake)* **Muối** *(Salt)*
Ma trận/Âm nhạc *(" name of a movie" or "a music album"/Music)*	**Ca-sĩ trẻ** *(Young Singers)* **Hình tượng** *(Image)* **Phuongthanhfc** *(phuongthanhfc)* **Thời-sự** *(Current Events)* **POP** *(POP Music)*	**Nhạc-sỹ** *(Musician)* **Ca-sỹ trẻ** *(Young Singers)* **Âm-nhạc** *(Music)* **POP** *(POP Music)* **Ca-sỹ** *(Singers)*

Figure 3.13: Examples of labeling without hidden topics and labeling with 120 topics in the testing collection. Note that the "cluster" in Query/Cluster column is the "key cluster" label assigned manually

Query Examples

We obtained 4 real query collections from Google for 4 another queries "sản phẩm" (*products*), "Hồng Sơn" (*a common name*), "ngôi sao" (*star*), "không hoảng" (*crisis*) which are not in the "Query Dataset". In comparison with the query collections in the "Query Dataset", these collections are not cleaned by the fact that we do not exclude "minor clusters" from them.

We then conducted clustering and labeling with 120-hidden topics and the baseline. The default parameters were set like in 3.3 and the merging threshold of 0.18. Other parameters for the experiments were set according to Table 3.3. We also submitted the queries to Vivisimo [102] in order to obtain clustering results. We compared the clusters generated for the queries in clustering/labeling with 120-hidden topic model, in the baseline, in Vivisimo in Figure 3.15 and Figure 3.16. The number of snippets in each cluster is written in the bracket next to the cluster label. Note that the query collections, which Vivisimo used, is different from the collection used in the baseline and clustering/labeling with hidden topics.

It can be observable from Figure 3.15 and Figure 3.16 that our proposal can provide better clustering/labeling results in comparison with Vivisimo and the baseline. Since Vivisimo is not optimal for Vietnamese, the clustering results are totally unsatisfactory. One obvious example is the cluster label "chính, khủng hoảng tài" of the query "khủng hoảng" (*crisis*). This phrase

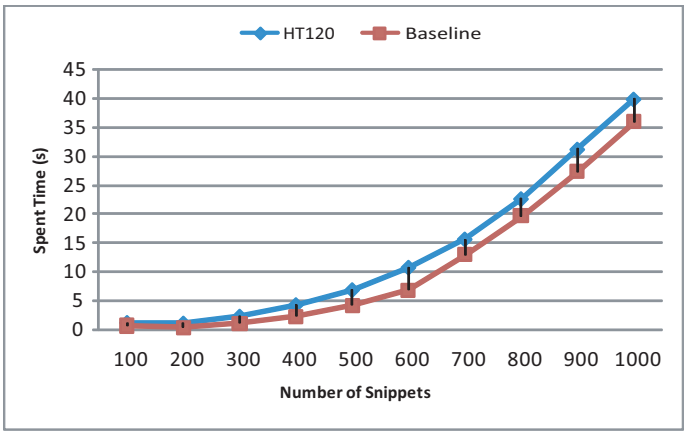


Figure 3.14: Computational time of HAC with hidden topics compared to HAC without hidden topics.

should be "khủng-hoảng tài-chính" in which "khủng hoảng" (*crisis*) is one valid word and "tài chính" is another valid word with two syllables in Vietnamese. Because word segmentation is not performed in Vivisimo, the two syllables "tài" and "chính" can not be joined to form the correct word. In comparison with the baseline, the clusters generated by our proposed method are better and assigned with more descriptive labels. Considering the query "sản phẩm" (*products*), for example, it is clear that the clusters in the baseline (introduction, news, vietnam) are either two vague or two general in comparison with the clusters in our proposed method (software product, mobile phone, insurance product, etc.). Another example is that the cluster of "singer, music stars" (of the query "start") should be a major cluster, which is recognized in our method, but are not generated in the baseline. For the query "Hồng Sơn", the cluster "môn phái" (*martial art group*) in our method actually corresponds to the cluster "Vietnam" in the baseline but the label in our method is much more descriptive.

3.6.5 Discussion

Analysis of clustering results affirmed the advantages of our approach. All in all, the main points having been discussed so far include:

- Clustering snippets with hidden topics: it is able to overcome the limitation of different word choices by enriching short, sparse snippets with hidden topics of the "universal dataset". This is particularly useful when dealing with web search results - small texts with only a few words and having less context-sharing. The effective of exploiting hidden topics from the universal dataset is expressed in two aspects: (1) increase similarity between two snippets having common topics but using different words; and (2) decrease similarity between two snippets sharing non-topic oriented words (including trivial words) which may not be removed completely in the phase of preprocessing. As a result, good clustering is achieved when we are able to assure the "snippet-tolerance" condition - an important feature for a practical

Hồng Sơn *(A personal name)*		
Clustering and Labeling with HT120	**The Baseline**	**Vivisimo**
Bác sĩ Phạm Hồng Sơn (14) *Doctor Pham Hong Son (14)* **Thủ môn Dương Hồng Sơn (8)** *Goal Keeper Duong Hong Son* **Diễn viên (8)** *Actor/Actress* **Đạo diễn Vũ Hồng Sơn (5)** *Director Vu Hong Son* **Ca sỹ (5)** *Singer* **Môn phái (5)** *Martial Art Group* **Nghệ an, ngôi đền (4)** *Nghe An, Temple* **Xã (4)** *Commune*	**Bác- sĩ Phạm Hồng Sơn (13)** *Doctor Pham Hong Son* **Thủ môn Dương Hồng Sơn (10)** *Goal Keeper Duong Hong Son* **Diễn viên (8)** *Actor/Actress* **Đạo diễn Vũ Hồng Sơn (5)** *Director Vu Hong Son* **Nguyễn Hồng Sơn (4)** *Nguyen Hong Son* **Xã (4)** *Commnune* **Việt Nam (4)** *Vietnam* **Ca sỹ (3)** *Singer*	**Nam, Việt (50)** *Nam, Viet* **Vietnamese (37)** *Vietnamese* **Phạm Hồng Sơn (20)** *Pham-Hong-Son* **Công (23)** *Cong* **Quang (13)** *Quang* **Dương Hồng Sơn (12)** *Duong Hong Son* **Thông tin (11)** *Information* **Dân trí (8)** *Dan tri*
Khủng hoảng *(Crisis)*		
Clustering and Labeling with HT120	**The Baseline**	**Vivisimo**
Ngân hàng (31) *Banks* **Khủng hoảng lương thực (18)** *Food Crisis* **Nền kinh tế (14)** *Economic* **Săn tiền vệ, khủng hoảng nhân sự (10)** *Hunt Players, Human Resource Crisis* **Doanh nghiệp Việt nam (10)** *Vietnam companies* **Xử lý khủng hoảng (9)** *Crisis management* **Giáo dục Việt nam (7)** *Vietnam Education* **Nhà đất (6)** *Real Estate* **Khủng hoảng chính trị (6)** *Political Crisis*	**Tín dụng Mỹ (22)** *United State Credit* **Thế giới (21)** *World* **Tài chính (17)** *Finance* **Vietnam (15)** *Vietnam* **Chính trị (8)** *Politics* **Nhân sự (7)** *Human Resource* **Giáo dục (7)** *Education* **Xử lý khủng hoảng (6)** *Crisis Management* **Thực phẩm thế giới (6)** *World Food*	**Chính, khủng hoảng tài (49)** *the phrase "Financial Crisis" but in the wrong order* **Việt Nam (43)** *Vietnam* **Vietnam(30)** *Vietnam* **Khủng hoảng kinh (21)** *Part of the phrase "Economic Crisis"* **Thông tin (19)** *Information* **Doanh (15)** *in "***Doanh nghiệp***" (Companies)* **Vietnamnet (12)** *Vietnamnet* **Thực, khủng hoảng lương (8)** *the phrase "Food Crisis" but in the wrong order*

Figure 3.15: Clustering using HAC with HT120 and labeling with RS-HT120 in new query collections

***Ngôi sao**(Star)*		
Clustering and Labeling with HT120	**The Baseline**	**Vivisimo**
Blogger (13) *Blogger* **Ca sĩ, ngôi sao ca nhạc (12)** *Singer, Music Stars* **Ngôi sao trẻ (9)** *Young Stars* **Ngôi sao phim (9)** *Film Stars* **Sân Mỹ Đình (9)** *My Dinh Stadium* **Mặc đẹp (8)** *Nice wearing* **Vũ trụ (8)** *Universe* **Tiền vệ (5)** *Half-back*	**Miley Cyrus (10)** *Miley Cyrus* **Thế giới (9)** *World* **Blogger (9)** *Blogger* **Công ty TNHH (9)** *LLC Companies* **Mặc đẹp (8)** *Nice Wearing* **Người lớn (8)** *Adult* **Đầu tiên, vũ trụ (8)** *First, Universe* **Vn, Tiền vệ (7)** *Vn, Half-back*	**Việt, Nam (37)** *Viet, Nam* **Phim (21)** *Film* **Những ngôi sao (14)** *Stars* **Nhac (11)** *Music* **Star (11)** *Star* **Lên, Đang (8)** *Len, Dang* **Bóng (5)** *in "***bóng đá***" (football)* **May mắn (4)** *Lucky*
***Sản phẩm** (Products)*		
Clustering and Labeling with HT120	**The Baseline**	**Vivisimo**
Sản phẩm phần mềm (20) *Software Product* **Doanh nghiệp (13)** *Companies* **Doanh số bán (11)** *Sell turnover* **Chất lượng sản phẩm (11)** *Product Quality* **Dịch vụ (9)** *Services* **Điện thoại di động (9)** *Mobile Phone* **Sản phẩm bảo hiểm (9)** *Insurance Product* **Intel, sản phẩm máy tính (8)** *Intel, Computer Products*	**Dịch vụ (13)** *Services* **Giới thiệu, Thái lan (10)** *Introduction, Thailan* **Chất lượng (9)** *Quality* **Tin tức, trang chủ (8)** *News, Homepage* **Thông tin (8)** *Information* **Công nghệ (8)** *Technology* **Việt nam (7)** *Vietnam* **Mã (6)** *Code*	**Giới, thiệu (30)** *Two syllables of the word "***giới thiệu***" (Introduction)* **Dịch (27** *The first syllable of the word "***dịch vụ***" (services)* **Vietnam(26)** *Vietnam* **Tính (17)** *The last syllable of the word "***máy tính***" (computer)* **Mới (17)** *New* **Mua bán (15)** *Selling* **Công ty TNHH (15)** *LLC Companies*

Figure 3.16: Clustering using HAC with HT120 and labeling with RS-HT120 in new query collections

clustering system. We conducted evaluation on two datasets - the "Web dataset" and "Query Dataset" - and showed significant improvement of our proposal.

- Labeling clusters using hidden topic analysis: by exploiting hidden topic information, we can assign clusters with more topic descriptive labels. Since snippets sharing topics are also gather in our method, there are not many repeating words in such clusters. Consequently, word frequency is not enough to determine labels for clusters generated by our method because. In this aspect, phrases sharing topics with most of the snippets in the cluster should be considered significant. Thank to the complete generative model of Latent Dirichlet Allocation, we have a coherent way to map snippets, clusters, and label candidates into the same topic space. As a result, similarity in terms of topics between these clusters, snippets, label candidates are easy to be formalized by using some typical similarity measures such as cosine measure. For evaluation, we split the "Query dataset" into 2 parts (training data and testing data). We learned two relevance scores from the training data (RS-base, in which we do not consider hidden topic information, and RS-HT120, in which we take topics from the 120-topic model of the Universal dataset into account). We then conducted labeling and measured ranking performance (P@5, P@10, and P@20) for two relevance scores in the testing data and showed that labeling with hidden topics can provide better performance.

- Finding collocations in the universal dataset: using the universal dataset helps to find out meaningful phrases such as "điện_thoại di_động" (*mobile phone*), "thị_trường chứng_khoán" (*stock market*) as labels for clusters. For labeling, we need to extract label candidates and then rank them with regards to some specific conditions. In order to obtain meaningful phrases as label candidates, we find collocations (two or more words commonly used together as fix phrases) using hypothesis testing. Due to the fact that the universal dataset is much larger than snippet collections but snippet collections contain query-oriented text, we find collocations both in the universal dataset and snippet collections. This helps to find out both common noun phrases such as "công nghệ thông tin" (*information technology*), which probably have not enough statistics in snippet collections to be verified as collocations, and named entities or specific phrases which may not occur in the universal dataset such as "Doctor Phạm-Hồng-Sơn" in the snippet collection "Hồng Sơn" (*a common name*)).

- Computational time vs. Performance: this is an important aspect to consider in any practical applications. Hidden topics bring improvement to clustering process but add extra computational time caused by the analysis process and the usage of topic vectors. For the analysis process, we use Gibbs sampling based on the estimated model. Once the model is converged in the estimation process, 30-50 sampling iterations is quite enough for topic analysis for each snippet collection. So, the complexity of the additional time caused by this step is O(n) in which n is the number of snippets in the collection. However, since the size of these topic vectors are fixed (because the number of topic is fixed) while the number of rare words can be removed without losing the connections between snippets are increased (as analyzed in the previous section), term-vectors of snippets can be reduced in size. This helps us to obtain good clustering performance while decreasing the additional time.

- Flexibility and Simplicity: these are advantages of the framework which have been pointed out in our proposal. Here, all we need is to collect a large collection and use it for several phases in our framework. Analysis of the large collection is totally unsupervised, it requires small effort of humans for preprocessing the collection. This is particularly useful when dealing with languages lacking knowledge bases and other linguistic processing toolkits. As a result, this solution works well for Vietnamese and similar languages. The flexibility of our framework is also shown by the fact that the framework does not limit to any topic model or clustering algorithm. We can use CTM or topical n-gram model with K-means to obtain better results while optimizing clustering/labeling time complexity.

3.7 Conclusion

This chapter has presented a framework for clustering and labeling with hidden topics, which (to the best of our knowledge) is the first careful investigation of this problem in Vietnamese. The main idea is to collect a large dataset and then estimate hidden topics for the collection based on one of the recent successful topic models such as pLSI, LDA, CTM. Using this estimated model, we can perform topic inference for snippet collections which need to be clustered. The old snippets are then combined with hidden topics to provide a richer representation of snippets for clustering and labeling. It has been shown that this integration helps overcome the sparseness of snippets returned by search engines and improve quality of clustering. By using hidden topics for labeling clusters, we can assign more descriptive and meaningful labels to the clusters. We have evaluated the quality of the framework via a lot of experiments. Also, through examples and analyzing clusters, we have proved that our approach is somewhat satisfies the three requirements of Web search clustering (high quality clustering, effective labeling and snippet-tolerance) in Vietnamese.

Once we estimated a topic model from the universal dataset, we can use it for multiple applications in information retrieval. In next chapter, we will adapt the framework to matching and ranking problem and show the application in online contextual advertisement.

Chapter 4

Matching and Ranking toward Online Contextual Advertising

4.1 Introduction

Along with the rapid growth of the Internet, online advertising has become an essential part of e-commerce nowadays [1]. According to the Interactive Advertising Bureau (IAB) [45], Internet advertising revenues reached its new peak (26.0 billion dollar) in 2010, up 15% from 2009 (see Figure 4.1). This is also the first time that online advertising surpasses newspaper in ad revenue. Its growth is expected to continue as consumers spend more and more time online.

Since its birth in 1994, online advertising has developed both in its appearance and the way it attracts Web user's attention. Figure 4.1 shows typical ad formats and their sharing in U.S. advertising market in 2009 and 2010. Display is the earliest type of advertising when marketers pay Web owners for space to place static, graphic banners or logo on Web sites. This type of advertising is simple but still be very common until now, which amounts to 24% of advertising market in 2010, increases 2% compared to 2009. The disadvantage of the display method is that it is unable to automatically discover potential customers. Beside, it sometimes relates to annoying issues such as pop-up banners, and unexpected threat to users' computers such as Trojan. Although search marketing reduced a little in 2010 compared to 2009, it still occupied the largest share in U.S. advertising market (46%) and be supported by most of search engines such as Google, Yahoo, etc. Search advertisement has several forms to advertise via search engines, in which two important methods are listed in the following:

- Sponsored search: the marketers pay search engine companies to associate their links to chosen keywords. When a user search for those keywords, text links appear at the top or side of search results. The more the marketers pay for each click, the higher position they can obtain.

- Contextual Search: text links appear on some articles based on the correspondence between

[1] http://onlinejournal.com/artman/publish/article_8343.shtml

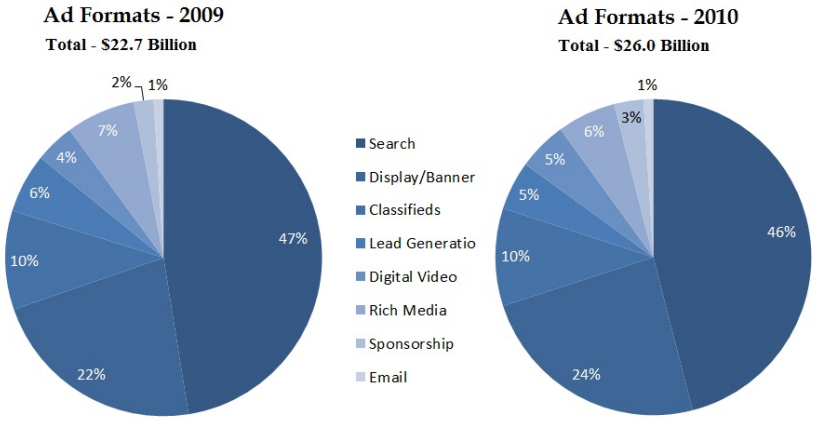


Figure 4.1: Advertising categories and their shares in adverting market (From [45])

the content of the articles and ad messages. Contextual search is usually conducted by an advertising network of some search engine company, which plays as an agent between Web publisher, users and advertisers. Google AdSense[2] (Figure 4.2) is one example of such type of advertising network, which helps Website publishers of all size to earn money by displaying targeted Google ad messages on their websites. Payment only occurs when the links are clicked.

Contextual advertisement is not only included in searching format, which gains the largest revenue, but also in Email format (Figure 4.1). In Gmail, yet another product of Google company, ads are placed to relate to the content of the emails. The automatic advertising in emails are essential to ensure the privacy, i.e. no humans read your emails. The technique in Gmail is the same contextual advertising technology that powers Google AdSense.

Formally, the problem of contextual advertising is based on the content to deliver ad messages, which normally consist of four parts: title, body, URL, and keywords, to the Web pages that users are surfing. It can therefore provide Internet users with information they are interested in and allow advertisers to reach their target customers in a non-intrusive way. In contextual advertising, one important observation is that the relevance between target Web pages and advertising messages is a significant factor to attract online users and customers [18, 105]. In order to suggest the "right" ad messages, we need efficient and elegant contextual ad matching and ranking techniques.

Different from sponsored search, in which advertising are chosen depending on only the keywords provided by users, contextual ad placement depends on the whole content of a Web page. Keywords given by users are often condensed and reveal directly the content of the user's concerns, which make it easier to understand. Analyzing Web pages to capture the relevance is a more complicated task. Firstly, as words can have multiple meanings and some words in the target page are not important, they can lead to mismatch in lexicon-based matching method. Moreover, a target page and an ad can still be a good match even when they share no common words or terms.

[2]Google Adsense: http://adsense.google.com

Figure 4.2: An example of Google Adsense

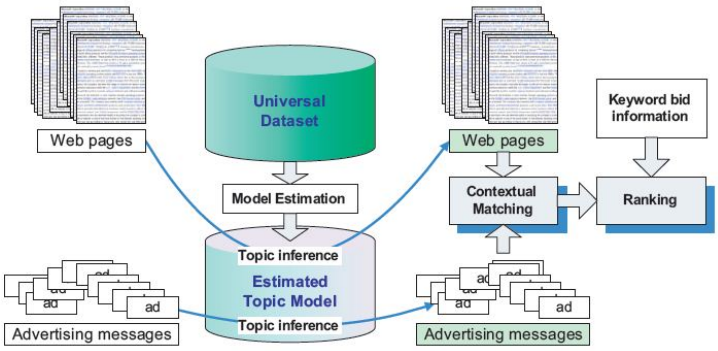
To deal with these problems, we present a framework[3] that can discover the semantic relatedness between Web pages and ads by analyzing *implicit* or *hidden* topics for them. After that, both Web pages and advertisements are expanded with their most relevant topics, which helps reduce the sparseness and make the data more topic-focused. The framework can therefore overcome the limitation of word choices, deal with a wide range of Web pages and ads, as well as process future data, that is, previously unseen ads and Web pages, better. It is also easy to implement and general enough to be applied in different domains of advertising and in different languages.

The organization of the rest of this chapter consists of six sections. Section 4.2 represents typical approaches to contextual advertising. Section 4.3 formalizes the problem of contextual advertisement and presents the framework to bridge the semantic gaps between Web pages and ad messages to support matching and ranking. Hidden topic analysis for a collected universal dataset will be presented in Section 4.4. Based on the estimated topic model, we can perform matching and ranking as given in Section 4.5. Finally, conclusions will be given in Section 4.7.

4.2 Related Work

Inspired by the success of sponsored search, Yih et al. [113] proposed a method that analyzes Web pages to extract keywords and matches them against a given database of ads, which are also associated with keywords chosen by advertisers. A good keyword selection is very important toward content-based advertising, and brings great benefits to users, web page owners and advertisers. The proposed method consists of four phases, i.e. *preprocessing*, *candidate selector*, *classifier*, and *post-processor*. Firstly, a web page is analyzed to remove HTML while preserving *blocks*, i.e. texts in the same table should be placed together without HTML tags. Also in this step, the authors

[3]This is a joint work with Dieu-Thu Le and Xuan-Hieu Phan



(a) *Choosing an appropriate "universal dataset"*
(b) *Doing topic analysis for the universal dataset*
(c) *Doing topic inference for Web pages and ads*
(d) *Page-Ad Matching and Ranking*

Figure 4.3: Framework of page-ad matching & ranking with hidden topics

performed shallow text processing such as sentence splitter, part-of-speech (POS) tagging, etc. to extract useful features for latter steps. Second, several strategies have been proposed to select candidate phrases (up to 5 consecutive words) from title, meta-data as well as in the body. In the third and most important phase, the author used logistic regression to build a classifier that map from a candidate phrase to [0,1] where 1 means the candidate is selected as keyword and 0 otherwise. The output of the third step is a ranked list of keywords. Based on the strategies of selecting candidates in the second step, different methods were proposed to further shorten or combine the list of keywords to generate final keywords and phrases that describe the content of Web page.

In another study, Lacerda et al. [56] improved the ranking function based on Genetic Programming (GP). Given the available evidences, such as term and document frequencies, document length and collection's size, they used GP to select relevant ads for Web pages. According to the authors, GP was able to select ranking functions that are very effective in placing ads in web pages.

One challenge of contextual matching task is the difference between the vocabularies of Web pages and ads. Ribeiro-Neto et al. [88] focused on solving this problem by using additional pages. It is similar to ours in the idea of expanding Web pages with external knowledge to decrease the distinction between their vocabularies. However, they determined added terms from other similar pages by means of a Bayesian model. Those extended terms can appear in ad's keywords and potentially improve the overall performance of the proposed method. Their experiments have proved that when decreasing the vocabulary distinction between Web pages and ads, we can find better ads for a target page. Broder et al. [14] proposed a method for matching ads based on both semantic and syntactic features. For syntactic features, they used the TF-IDF score and section score (title, body or bid phrase section) for each term of Web pages or ads. For semantic matching, the authors exploited the structure of taxonomy and relied on it to classify ads.

Corpus Statistics
After removing html, performing sentence and word segmentation: $size \approx 219M$, $\|docs\| = 40,328$
After filtering and removing non-topic oriented words: $size \approx 53M$, $\|docs\| = 40,268$, $\|words\| = 5,512,251$, $\|vocabulary\| = 128,768$

Table 4.1: VNExpress news collection serving as "Universal Dataset" for contextual advertising

Bearing the semantic gaps between Web pages and ads in mind, our framework also try to enrich Web pages and ads to improve matching performance. Unlike [88, 14], which use additional pages and taxonomy, we are based on hidden topic analysis, which have proved to be very effective and adaptable to different domains and languages.

4.3 Page-Ad Matching and Ranking Framework

Given a set of n target Web pages $P = \{p_1, p_2, \ldots, p_n\}$, and a set of m ad messages (ads) $A = \{a_1, a_2, \ldots, a_m\}$. For each Web page p_i, we need to find a corresponding ranking list of ads: $A_i = \{a_{i1}, a_{i2}, \ldots, a_{im}\}, i \in 1 \ldots n$ such that more relevant ads will be placed higher in the list. These ads are ranked based on their relevance to the target page the keyword bid information. However, in the scope of our work, we only take linguistic relevance into consideration and assume that all ads have the same priority, i.e, the same bid amount.

As depicted in Figure 4.3 and similar to the framework of Chapter 3, the first important thing to consider in this framework is collecting an appropriate external large-scale document collection (a) which is called Universal Dataset. To take the best advantage of it, we need to find an approximate universal dataset for the Web pages and ad messages. First, it must be large enough to cover words, topics, and concepts in the domains of Web pages and ads. Second, its vocabularies must be consistent with those of Web pages and ads, so that it will make sure topics analyzed from this data can overcome the vocabulary impedance of Web pages and ads. The universal dataset should also be processed to remove noise and stop words before analysis to get better results. The result of step (b), hidden topic analysis , is an estimated topic dataset and the distributions of topics over terms. Topic models for step (a) can be any topic modeling methods like those presented in Chapter 2. Similar to topic analysis of universal dataset in Chapter 3, we demonstrated the topics estimated using Latent Dirichlet Allocation (LDA) in 3.4. After step (b), we can again perform topic inference for both Web pages and ads based on this model to discover their meanings and topic focus (c). This information will be integrated into the corresponding Web pages or ads for matching and ranking (d). Both steps (c) and (d) will be discussed more in section 4.3.

4.4 Hidden Topic Analysis of Universal Dataset

This section brings an in-detail description of hidden topic analysis of a large-scale Vietnamese news collection that serves as a "universal dataset" in the general framework for contextual advertising.

Topic 3	Topic 15	Topic 44	Topic 48	Topic 56	Topic 172
bác_sĩ (doctor)	**thời_trang** (fashion)	**thiết_bị** (equipment)	**chứng_khoán** (stock)	**bánh** (cake)	**thẻ** (card)
bệnh_viện (hospital)	**người_mẫu** (model)	**sản_phẩm** (product)	**công_ty** (company)	**mcdonald** (McDonald)	**khoá** (lock)
thuốc (medicine)	**mặc** (wear)	**máy** (machine)	**đầu_tư** (investment)	**thịt** (meat)	**rút** (withdraw)
bệnh (disease)	**trang_phục** (clothes)	**màn_hình** (screen)	**ngân_hàng** (bank)	**pizza** (pizza)	**chủ** (owner)
phẫu_thuật (surgery)	**thiết_kế** (design)	**công_nghệ** (technology)	**cổ_phần** (joint-stock)	**ba_tê** (pate)	**chìa** (key)
điều_trị (treatment)	**đẹp** (beautiful)	**điện_thoại** (telephone)	**thị_trường** (market)	**bánh_mì** (bread)	**thẻ_tín_dụng** (credit card)
bệnh_nhân (patient)	**váy** (dress)	**hãng** (company)	**giao_dịch** (transaction)	**bánh_ngọt** (pie)	**atm** (ATM)
y_tế (medical)	**sưu_tập** (collection)	**sử_dụng** (use)	**đồng** (VND)	**cửa_hàng** (shop)	**tín_dụng** (credit)
ung_thư (cancer)	**mang** (wear)	**thị_trường** (market)	**mua** (buy)	**xúc_xích** (hot dog)	**thanh_toán** (pay)
tình_trạng (condition)	**phong_cách** (style)	**usd** (USD)	**phát_hành** (publish)	**kem** (ice-cream)	**visa** (visa)
cơ_thể (body)	**quần_áo** (costume)	**pin** (battery)	**niêm_yết** (post)	**khai_trương** (open)	**tối_thiểu** (minimum)
sức_khoẻ (health)	**nổi_tiếng** (famous)	**cho_phép** (allow)	**bán** (sell)	**nguội** (cold)	**mastercard**
đau (hurt)	**quần** (trousers)	**samsung** (Samsung)	**tài_chính** (finance)	**hamburger** (hamburger)	**phát_hành** (release)
gây (cause)	**trình_diễn** (perform)	**di_động** (mobile)	**đấu_giá** (auction)	**thịt** (meat)	**trả_nợ** (pay debt)
khám (examine)	**thích** (like)	**sony** (Sony)	**trung_tâm** (center)	**nhà_hàng** (restaurant)	**sẵn_sàng** (ready)
kết_quả (result)	**quyến_rũ** (charming)	**nhạc** (music)	**thông_tin** (information)	**đồ_ăn** (food)	**mật_mã** (password)
căn_bệnh (illness)	**sang_trọng** (luxurious)	**máy_tính** (computer)	**doanh_nghiệp** (business)	**sandwich** (sandwich)	**thường_niên** (annual)
nặng (serious)	**vẻ_đẹp** (beauty)	**hỗ_trợ** (support)	**cổ_đông** (shareholder)	**khẩu_vị** (taste)	**cảnh_giác** (alert)
cho_biết (inform)	**gái** (girl)	**điện_tử** (electronic)	**nhà_đầu_tư** (investor)	**tiệm_bánh** (bakery)	**chủ_thẻ** (card owner)
máu (blood)	**gương_mặt** (figure)	**tính_năng** (feature)	**nhà_nước** (government)	**bảo_đảm** (ensure)	**theo_dõi** (follow)
xét_nghiệm (test)	**siêu** (super)	**kết_nối** (connect)	**tổ_chức** (organization)	**nướng** (grill)	**nhà_băng** (bank)
chữa (cure)	**áo_dài** (aodai)	**thiết_kế** (design)	**triệu** (million)	**bí_quyết** (secret)	**tội_phạm** (criminal)
chứng (trouble)	**giày** (shoes)	**chức_năng** (function)	**quỹ** (budget)	**ngon** (delicious)	**trộm** (steal)

Figure 4.4: Sample topics analyzed from VnExpress News Collection

With the purpose of using a large scale dataset for Vietnamese contextual advertising, we choose VnExpress as a large number of articles in many topics in daily life. For this reason, it is a suitable data collection for advertising areas.

This news collection includes different topics such as Society, International news, Lifestyle, Culture, Sports, Science, etc. We crawled 220 Megabytes of approximately 40,000 pages using Nutch. We then performed some preprocessing steps (HTML removal, sentence/word segmentation, stop words and noise removal, etc.) and finally got more than 50 Megabyte plain text. See Table 4.1 for the details of this data collection.

We performed topic analysis for this news collection using LDA (Chapter 2) with different number of topics (60, 120, and 200). Figure 5 shows several sample hidden topics discovered from VnExpress. Each column (i.e., each topic) includes Vietnamese words in that topic and their corresponding translations in English in the parentheses. These analysis outputs will be used to enrich both target Web pages and advertising messages (ads) for contextual advertising.

4.5 Matching and Ranking with Hidden Topics

4.5.1 Topic inference for Ads & Target Pages

For each snippet Web page/ad $\underline{m}$, after topic analysis, we obtain the topic distribution $\vec{\vartheta}_{\underline{m}} = \left\{\vartheta_{\underline{m},1}, ..., \vartheta_{\underline{m},k}, ..., \vartheta_{\underline{m},K}\right\}$. Topics that have high probability $\vartheta_{\underline{m},k}$ will be added to the corresponding Web page/ad $\underline{m}$. Each topic integrated into a Web page/ad will be treated as an *external term* and its frequency is determined by its probability value. Technically, the number of times a topic

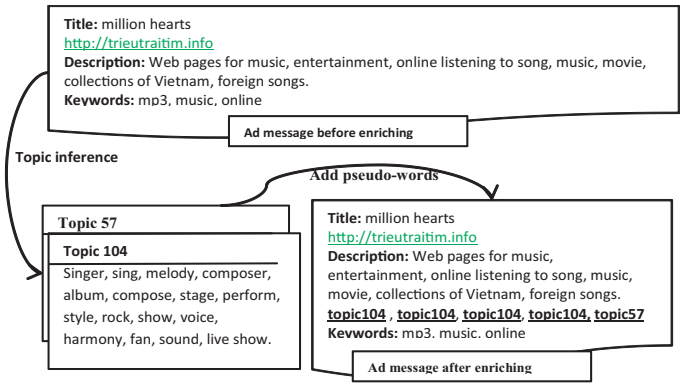


Figure 4.5: An example of topic integration into an ad message

k is added to a Web page/ad $\underline{m}$ is decided by two parameters *cut-off* and *scale*:

$$Frequency_{\underline{m},k} = \begin{cases} round(scale \times \vartheta_{\underline{m},k}), & \text{if } \vartheta_{\underline{m},k} \geq \textit{cut-off} \\ 0, & \text{if } \vartheta_{\underline{m},k} < \textit{cut-off} \end{cases} \tag{4.1}$$

where *cut-off* is the topic probability threshold, *scale* is a parameter that determines the topic frequency added. An example of topic integration into ads is illustrated in Figure 4.5. The ad is about an entertainment Web site with a lot of music albums. After doing topic inference for this ad, hidden topics with high probabilities are added to its content in order to make it enriched and more topic-focused.

4.5.2 Matching and Ranking

After being enriched with hidden topics, Web pages and ads will be matched based on their cosine similarity. For each page, , ads will be sorted in the order of its similarity to the page. The ultimate ranking function will also take into account the keyword bid information. But this is beyond the scope of this work.

We verified the contribution of topics in many cases that normal keyword-based matching strategy cannot find appropriate ad messages for the target pages. Since normal matching is based on only the lexical feature of Web pages and ads, it is sometimes deviated by unimportant words which are not practical in matching. An example of such case is illustrated in Figure 4.6. The word "triệu" (million) is highly frequent in the target page, hence play important role in lexical matching. The system then misleads in proposing ad messages for this target page. It puts ad messages having the same highly frequent word "triệu" in the top ranked list (c). However, those ads are totally irrelevant to the target page as the word "triệu" does not tell the content of the target page that is about real estate services. The words "chung cư" (apartment) and "giá" (price) shared by top ads proposed by our method ($Ad_{21}, Ad_{22}, Ad_{23}$) and the target page, on the other hand, are important words but less frequent than the word "triệu" (f). However, by analyzing topics, we can find out their latent semantic relations and thus realize their relevance since they share the same topic 155 (g) and important words "chung cư" (apartment) and "giá" (price).

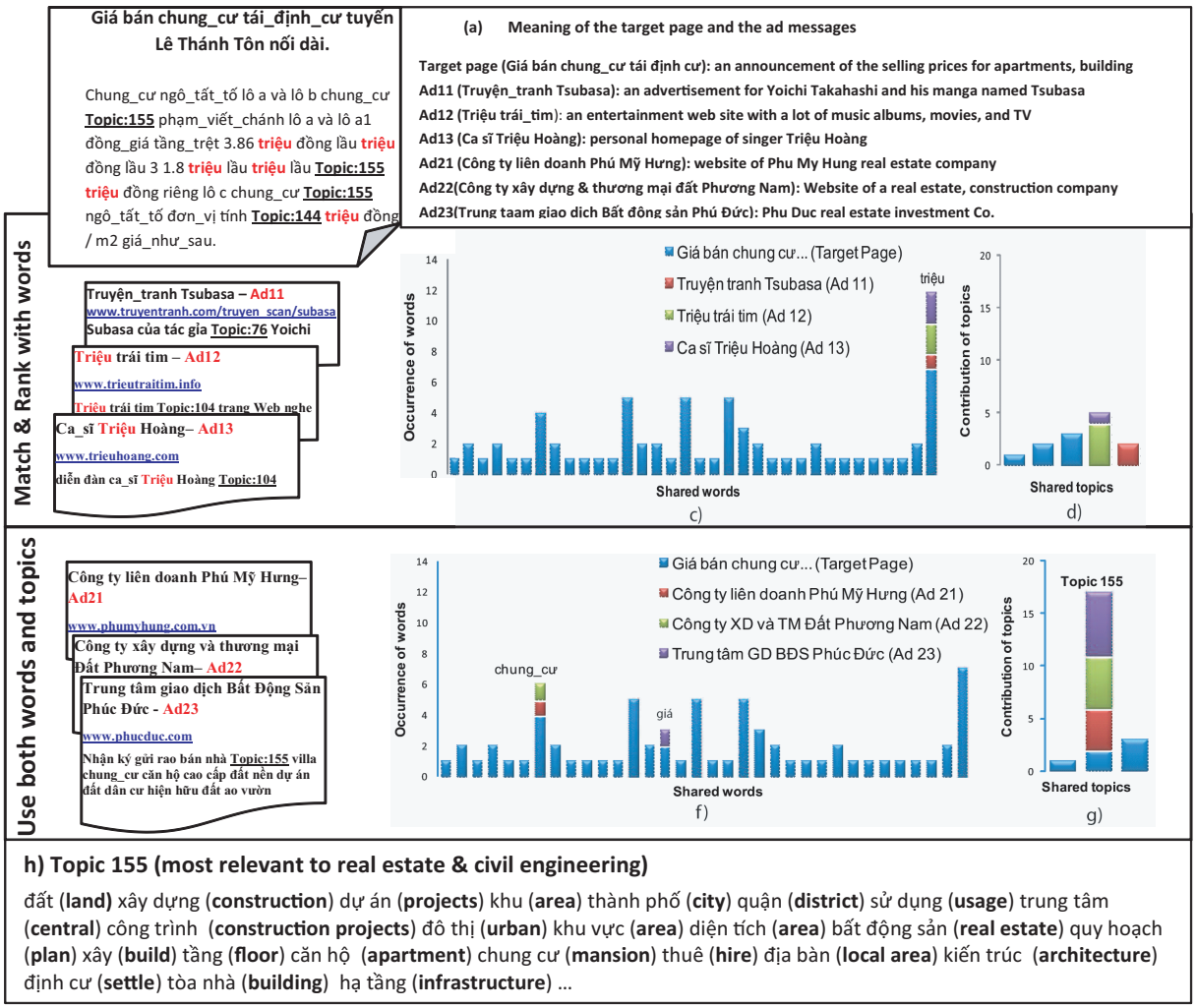


Figure 4.6: A visualization of an example of a page-ad matching and ranking *without* and *with* hidden topics. This figure attempts to show how hidden topics can help improve the matching and ranking performance by providing more semantic relevance between the target Web page and the ad messages. All the target page and the ads are in Vietnamese. The target page is located at the top-left corner. Part (a) explains the meanings of the target page and the ads, i.e., $Ad_{11}, Ad_{12}, and Ad_{13}$ in the ranking list *without* using hidden topics (i.e., using keywords only); Part(c) is the visualization of shared words between the target page and the three ads $Ad_{21}, Ad_{22}, Ad_{23}$ in the ranking list using hidden topics. Part(f) visualizes the shared words between the target page and the three ads $Ad_{21}, Ad_{22}, Ad_{23}$; Part (g) shows the shared topics between the target page and $Ad_{21}, Ad_{22}, Ad_{23}$; Part (h) shows the content of hidden topic number 155 (most relevant to *real estate* and *civil engineering* that is much shared between the target page and the ads $Ad_{21}, Ad_{22}, Ad_{23}$)

.

4.6 Experiments

In contextual advertising, matching and ranking ad messages based on their relevance to the targeted web page are important factors. As stated earlier, they help increase the likelihood of visits to the website pointed by the ad. So far, we have introduced our framework to perform this task. In this framework, we use hidden topics discovered from a huge external document collection (i.e., the universal dataset) in order to solve the sparse data problem (i.e., few common keywords between target pages and ads) and the synonym & homonym phenomena. The universal dataset in the VnExpress news collection that has been described earlier. All the test target Web pages and the test ads were collected from Vietnamese Web sites. We will present experimental data, experimental settings, evaluation methodology & metrics, as well as the experimental results & analysis in more detail in the following subsections.

4.6.1 Experimental Data

We quantified the effect of matching and ranking *without* and *with* hidden topics using a set of 100 target Web pages and 2,706 unique ads.

For target Web pages, we chose 100 pages randomly from a set of 27,763 pages crawled from VnExpress, one of the highest ranking e-newspapers in Vietnam. Those pages were chosen from different topics:Food, Shopping, Cosmetics, Mom & children, Real estate, Stock, Jobs, Law, etc. These topics are primarily classified on the e-newspaper. Note that the information of these classified topics not used in our experiments, just for reference here only.

For ad messages, as contextual advertising has not yet been applied in Vietnam to our knowledge, it is difficult to find a real Vietnamese advertisement collection. Up to now, advertisement types in Vietnam are mainly banners, thus such kind of real ad messages are not available. We have also contacted some online advertising companies, such as VietAd[4], a company in which keyword-based advertising system has once been tested in Vietnamnet [5]. However, their database was just for testing and the number of such advertisements was only a few. In order to conduct the experiments, we chose another resource: Zing.VN[6], a rich online directory of Vietnamese Web sites. It suits the form of contextual ad messages perfectly. Each ad message is composed of four parts:title, Web site's URL, its description and some important keywords. After crawling all 3,982 ad messages from Zing.VN directory, we preprocessed the data by doing sentence segmentation, word tokenization, removal of non-topic-oriented words, e.g. stop words. Nevertheless, keywords in this database are almost none-tone, so we cannot use them directly to enhance the matching performance. However, keywords play an important role in contextual advertising. The contribution of them in matching and ranking has been proved through experiments and affirmed in many previous studies [14, 88]. Therefore, we recovered tone for all keywords of the ads in order to improve the performance. After preprocessing, we selected 2,706 unique ads for evaluation.

[4]VietNam Advertise Company: http://vietad.vn/
[5]Vietnamnet: http://vietnamnet.vn
[6]Vietnamese Zing Directory: http://directory.zing.vn/

Settings	**Target Web page p**	Ad message a
AD	p	a
AD_KW	p	$a \cup KWs$
AAK_EXP	$p \oplus r$	$a \cup KWs$
HT[m]_[n]	$p \oplus HT_{s_p}$	$a \cup KWs \cup \oplus HT_{s_a}$

•p = Web page content
• a = ad title + a short ad description
• KWs: tone-recovered keywords from the original ad messages
• HT_{s_p}, HT_{s_a}:two sets of most likely hidden topics inferred from the topic model for p and a respectively.
• $\oplus$ means the inclusion of hidden topics by doing topic inference
• $m := 60, 120, 200$, the #hidden topics in the topic models
• $n := 10, 20$, the scaling value used for hidden topic integration
• We have 3 baselines (AD, AD_KW, AAK_EXP) and 6 hidden topic based HT60_10, HT60_20, HT120_20, HT200_10, HT200_20.

Table 4.2: Experimental settings for page-ad matching & ranking

4.6.2 Experimental Settings

In order to evaluate the importance of keywords in contextual match and the contribution of hidden topics in this framework, we performed some different matching strategies as follows: First, to access the impact of keywords in contextual match, we implemented two retrieval baselines following the approach of Ribeiro-Neto et al. [88]. The first strategy is called title and description only. The second is AD_KW, that is, matching a Web page and an ad message using ad's additional keywords, which have already been tone-recovered. The similarity between a target Web page and ads is computed using *cosine* function. The, the similarity of a Web page p and an ad message a is defined as follows.

$$sim_{AD}(p, a) = similarity(p, a)$$

$$sim_{AD\_KW}(p, a) = similarity(p, a \cup KWs)$$

where KWs is the set of keywords associated with the ad message a. We then used these two settings as the baselines for comparison. Second, to compare the contribution of hidden topics with additional terms in the Impedance Coupling method [88], we implemented the AAK_EXP method as follows:

$$sim_{AAK\_EXP}(p, a) = similarity(p \oplus r, a \cup KWs)$$

where AAK_EXP follows the implementation in [88], r is the set of additional terms provided by Impedance Coupling technique. These terms are extracted from a large enough dataset of additional web pages. First, the relation between this dataset, its terms and each target web page is represented in a Bayesian network model. Let $\mathcal{N}$ be the set of the k most similar documents d_j to each target page. The probability that term T_i in set $\mathcal{N}$ is a good term for representing a topic

the web page P is then determined as follows:

$$P(T_i|P) = \rho((1-\alpha)w_{i0} + \alpha \sum_{j=1}^{k} w_{ij} sim(r, d_j))$$

where ρ is a normalizing constant, w_{i0} and w_{ij} are the weights associated with term T_i in page P and in document d_j. The number of additional terms in r for enriching target page P is decided by the given threshold β. To perform this method, we use the same $40,268$ Web pages in universal dataset as additional dataset. In the experiment, we chose $\beta = 0.05$ as mentioned in [88] and $\alpha = 0.7$, which adjust the amount of additional terms in each target page. The set r will then be integrated with content of each target page to match with advertisements.

In order to evaluate the contribution of hidden topics, we carried out six different experiments, which are called HT (hidden topic) strategies. After doing topic inference for all Web pages and ads, we expanded their vocabularies with their most likely hidden topics. As described earlier, each Web page or ad have a distribution over hidden topics. We then chose topics having high probability values to enrich that page or ad. The similarity measure between a target Web page p and an ad a, denoted by $sim_{HT[m]\text{-}[n]}(p, a)$ is computed as follows.

$$sim_{HT[m]\text{-}[n]}(p, a) = similarity(p \oplus HT_{s_p}, a \cup KWs \oplus HT_{s_a})$$

in which m and n are the total number of topics in the topic model and the scale value as described in Table 4.2, we used the value cutoff of 0.05 and tried two different scale values: 10 and 20. We therefore performed six experiments: HT60_10, HT60_20, HT120_20, HT200_10, HT200_20.

4.6.3 Evaluation Methodology and Metrics

To evaluate the extent to which hidden topics contribute to the improvement of matching and ranking performance, we prepared the test advertising data for 100 target Web pages with the same methodology used in Ribeiro-Neto et al. [88]. The test data preparation, as depicted in Figure [14] is as follows. First, we started by matching each Web page to all the ad messages and ranking them to their similarities. The 9 methods proposed 9 different rank lists of ad messages to a target page. Since the number of ad messages is large, these lists can be different from this method to another method with little or no overlap. To determine the precision of each method and compare them, we selected top four ranked ads of each method and put them into a pool for each target page. Consequently, each pool will have no more than 36 ad messages. We then manually selected from these pools the most relevant ads and excluded irrelevant ones. On average, each Web corresponds with a key list of 6.9 ads eventually. To calculate the precision of each method, we used 11-point-average score, a metric often used for ranking evaluation in Information Retrieval. For every Web page and an ad message rank list of each method, we are based on the corresponding key list to calculate the precisions at 11 points of recall: $0, 0.1, 0.2, \ldots, 0.8, 0.9, 1.0$. The average precision of these 11 points is measured to obtain 11-point-average.

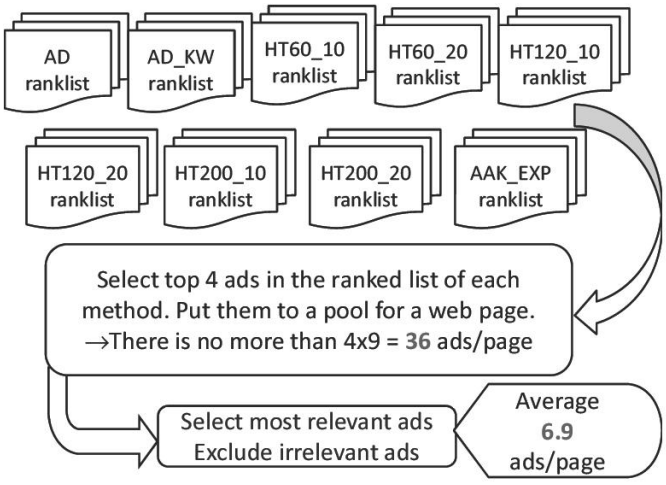


Figure 4.7: Preparation for test ads

4.6.4 Results and Analysis

We used the method AD_KW as a baseline for our experiments which uses hidden topics. We examined the contribution of hidden topics using different estimated models: the model of 60, 120 and 200 topics. As illustrated in Figure 4.8 and Table 4.3, using hidden topics significantly improves the performance of the whole framework. Figure 4.8 shows seven precision-recall curves of seven experiments in which the most inner line is the baseline and all the others are with hidden topics. From the curves, we can see the extent to which hidden topics can improve matching and ranking accuracy, and how the parameter values (i.e., number of topics, scale value) affect the performance. From Table 4.3, we can see hidden topics help increase the precision on average from 66% to 73 % and reduces almost 21% error (HT200_20).

For the overall methods, we also calculated the number of corrected ads found in the first, second and third position of the rank lists proposed by each strategy (#1, #2, #3 in Table 4.3). Because in contextual advertising, we normally consider only some first ranked ads, we want to examine the precision of these top slots. It also reflects that the precision of our hidden-topic methods is higher than that of the baseline matching method. Moreover, the precision at position 1 (#1) is generally higher than that of position 2 and 3 (#2, #3). If the system ranks the relevant ads near the top of the ranking list, it is possible that the system can suggest most appropriate ads for the corresponding page. It therefore shows the effectiveness of the ranking system.

Impedance Coupling method is another solution to match Web pages and ads by expanding the text of the web page, which is similar to the Hidden Topic idea in reducing vocabulary impedance. To compare with this method, we use the same web pages in universal dataset to extract additional terms. As shown in Figure 4.9, the accuracy of AAK_EXP method is almost the same as HT60 method but less than HT120 and HT200 method (Table 4.3. However, one limitation of the Impedance Coupling method is time consuming. Using the same number of web pages in universal dataset, for each target page, the system has to compute the similarity of the target page with each

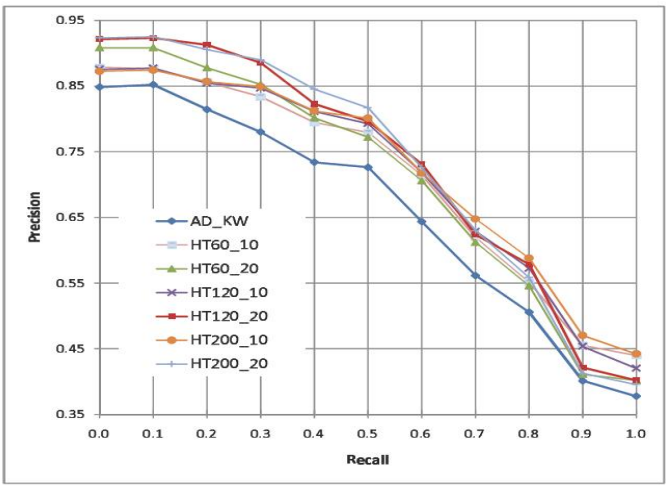


Figure 4.8: Precision-recall curves of the baseline (without hidden topics) and the 6 settings with hidden topics

Methods	Correct ads found				11-point avg. precision
	#1	#2	#3	Totals	
AD	72	56	54	182	50.22%
AD_KW	79	71	66	216	65.85%
AAK_EXP	87	76	79	242	70.98%
HT60_10	82	78	74	233	70.89%
HT60_20	85	80	74	239	70.87%
HT120_10	82	78	79	239	71.37%
HT120_20	87	83	76	246	72.88%
HT200_10	81	78	78	237	72.09%

Table 4.3: Precisions of positions #1, #2, #3 and 11-point-average

document in the dataset to find the relation with k most similar pages. After that, for every terms in this set, the probability that this term is good for enriching the target page is calculated to find the set of best terms r. This process takes a considerable computational time while the number of web pages and ads in real application is very large. For Hidden Topic method, although estimating the universal dataset would take a long time, once it is estimated, the model can be used for topic inference for web pages and ads. This process is very fast and only takes several seconds to do topic inference for thousands of short documents. This is the main advantage of Hidden Topic model in comparison with Impedance Coupling.

Finally, we also quantified the effect of the number of topics and its added amount to each Web page and ad by testing with different topic models and adjusting the scale values. As indicated in Table 4.3, the performance of 120 and 200-topic models yields a better result than 60-topic model. However, there is no considerable change between 120-topic and 200-topic models, also in

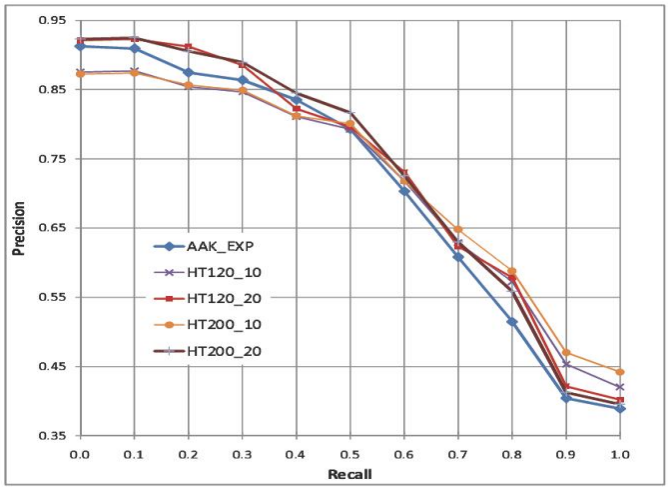


Figure 4.9: Precision-recall curves of the Impedance Coupling method and the Hidden Topics method

the quantities of added topics to each page and ad. It can therefore conclude that the number of topics should be large enough to discriminate the difference of terms to better analyze topics for Web pages and ads. However, when the number of topics is large enough, the performance of the overall system becomes more stable. The framework has shown its efficiency through a variety of experiments against the basic method using syntactic information only and the method adding terms from additional web pages. In practice, the results record an error reduction of 21% in the method using 200-topic model over the normal matching strategy without hidden topics. This indicates that this high quality contextual advertising framework is easy to implement and practical in reality.

4.7 Conclusions

This chapter reviews the problem of contextual advertisement in the general picture of online advertising. The mismatching between Web pages and ad messages prevents us from placing "right" ad messages to "right" Web pages. As a result, it reduces benefits of advertisers, Web publishers, and brings unexpected problems to users.

Follow the study in Chapter 3, we adapted the framework in Chapter 3 to another application - the problem of contextual advertising. Here, the framework ranks most relevant ads for a Web page by taking advantage of hidden topics discovered from a large data collection. This helps overcome the problem of mismatching by capturing the semantic information and reducing the sparseness in the vocabularies of both Web pages and ads. The framework has shown it efficiency through a variety of experiments against the basic method using lexical information only. In practice, the results record an error reduction of 22.9% in the method using 200 topics over the normal matching strategy without hidden topics. Along with the results from Chapter 3, we demonstrated that our

approach is adaptable to different domains, applications.

Not only be useful toward applications in text retrieval, but also topic modeling helps reduce semantic gaps in image retrieval. Next chapter will gives more details about how topic models, which origins from text modeling, can be adapted to apply to image annotation and retrieval.

Chapter 5

Feature-Word-Topic Model for Image Annotation and Retrieval

5.1 Introduction

As high-resolution digital cameras become more affordable and widespread, the use of digital images is growing rapidly. At the same time, online photo-sharing websites (Flickr, Picasaweb, Photobucket, etc.) hosting hundreds of millions of pictures have quickly become an integral part of the Internet only after a couple of years. As a result, the need for better understanding of image data and multimedia data become increasingly important in order to make the Web more well-organized and accessible. Current commercial image retrieval systems are mostly based on text surrounding of images such as Google and Yahoo image search engines. Since they ignore visual representation of images, the search engines often return inappropriate images. Moreover, this approach cannot deal with images that are not accompanied with texts.

Content-based image retrieval, as a result, has become an active research topic over the last few years [21]. While early systems were based on the query-by-example schema, which formalizes the task as search for best matches to example-images provided by users, the attention now moves to query-by-semantic schema in which queries are provided in natural language. This approach, however, needs a huge image database annotated with semantic labels. Due to the enormous number of photos taken every day, manual labeling becomes an extremely time-consuming and expensive task. As a result, automatic image annotation receives significant interest in image retrieval and multimedia mining.

Image annotation is a difficult task due to three problems namely *semantic gap*, *weakly labeling*, and *scalability*. The typical "semantic gap" problem [21] is between low level features and higher level concepts. It means that extracting semantically meaningful concepts is hard when using only low level image features such as color or textures. The second problem, "weakly labeling" [17], comes from the fact that exact mapping from keywords to image regions is usually unavailable. In other words, a label is given to an image without indications of which part of the image corresponds to that label. Since image annotation is served directly for image retrieval, "scalability" is also an important requirement and a problematic issue of image annotation.

SML: mast, bush, hill, trees, *sailboat*, overcast
Our method: bush, trees, overcast, sidewalk, hill

SML: zebra, cheetah, grass, *ocean, house*
Our method: zebra, cheetah, grass, sky, trees

Figure 5.1: Example of annotations in SML and our method

There has been a lot of effort to design automatic image annotation systems. Generally, we can categorize recent methods into two main approaches: (1) statistical generative models; (2) multi-instance learning. Statistical generative models [12, 28, 57, 70] introduce a joint distribution of visual features and labels by making use of common latent variables. In general, this approach is scalable in database size and the number of concepts of interest. However, since they do not explicitly treat semantics as image classes, what they optimize does not directly imply the quality of annotation [17]. Recently, image annotation based on multi-instance learning has become an emerging approach [17, 44, 116]. Multiple instance learning (MIL) is a variation of supervised learning for problems with incomplete knowledge about labels of training examples. In MIL, labels are assigned to "bags of instance". Here, a bag has one word as its label if at least one instance in that bag is associated with that word. Applying to image annotation, "bags" can be formalized as regions from images with the same label [17] or regions of one image [116]. This method can be seen as a potential solution to the problem of "weakly labeling" stated above. Among these methods, Supervised Multiclass Labeling model (SML) [17] is a state-of-the-art in image annotation and retrieval, and can be implemented with algorithms that are conceptually simple and computationally efficient. One disadvantage of SML is the absence of label relationships in annotation. The point is word associations such as {beach, sand}, or {ocean, fish} should be considered to reduce annotation error (thus, improve performance).

This chapter presents a novel method for image annotation, which is based on feature-word and word-topic distributions. The main idea is to guess the scene settings or the story of the picture for image annotation. Take the leftmost picture in Figure 5.1 as an example, if we (human) see this picture, we first obtain the story of the picture such as "a scene of forest with a lot of trees and a narrow path, in dark". Next, we can select "keywords" as "labels" based on it. Unfortunately, only based on "visual features", SML selects "masts" for the best keywords since it has several small white parts, which resembles to sails. Here, branches are confused with "mast" learned from images with sea scene in the databases. If, somehow, we can guess scene settings of the picture, we can avoid such confusion. We successfully resolved it and our annotations given in Figure 5.1 capture the scene better.

More specifically, we learn two models from the training dataset: 1) a model of feature-word distributions based on multiple instance learning and mixture hierarchies, which is like SML; 2) a model of word-topic distributions (topic model) estimated using probabilistic latent semantic analysis (pLSA). The models are concatenated to form feature-word-topic model for annotation, in which only words with highest values of feature-word distributions are used to infer latent topics of the image (based on word-topic distributions). The estimated topics are then exploited to re-rank words for annotation. As a result, the proposed model provides some advantages as follows:

- The model inherits the advantages of SML. In the other words, it is able to deal with the "weakly labeling" problem and optimize feature-word distributions. Moreover, since feature-word distributions for two different words can be estimated in a parallel manner, it is convenient to apply in real-world applications where the dataset is dynamically updated.
- Hidden topic analysis, which has shown the effectiveness in enriching semantic in text retrieval [75, 78, 80], is exploited to infer scene settings for image annotation. By doing so, we do not need to directly model word-to-word relationships and consider all possible word combinations, which could be very large, to obtain topic-consistent annotation. As a result, we can extend vocabulary while avoiding combinational explosion.
- Unlike previous generative models, the latent variable is not used to capture joint distributions among features and words, but among words only. The separation of topic modeling (via words only) and low-level image representation makes the annotation model more adaptable to different feature selection methods, or topic modeling.

The rest of this chapter is organized in seven sections. Section 5.2 reviews some noticeable approaches to image annotation and related problems. The general learning framework is described in Section 5.3. The main parts of the proposed framework will be given in Section 5.4, 5.5 and 5.6. Sections 6 gives the discussion about the relationships of our annotation model with related works as well as the time complexity analysis. Section 5.7 shows our experiments and result analysis on two datasets. Finally, some concluding remarks are given in Section 5.8

5.2 Previous Work

Image annotation has been an active topic for more than a decade and led to several noticeable methods. Those methods are classified by the approaches to the problem into (1) Standard binary classification, (2) Statistical generative model; and (3) Multiple instance learning.

The early effort in the area is to formalize image annotation as the standard classification in one-vs-all (OVA) mode, in which one classifier is trained corresponding to one concept/label versus everything else. Some examples are to classify images into "indoor" or "outdoor" [94]; "city" or "landscape" [99]. The disadvantage of this approach is the training complexity is dominated by the large number of negative examples of one concept when the dataset is large [17].

As mentioned earlier, statistical generative models introduce a set of latent variables to define a joint distribution between visual features and labels. This joint distribution is used to infer

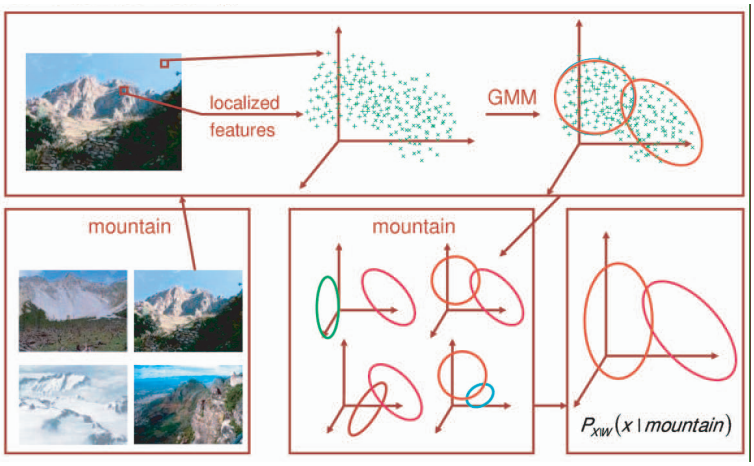


Figure 5.2: An bag of examples for word "mountain" which is used for learning $p(\mathbf{x}|mountain)$. This figure is from [17].

conditional distribution of labels given visual features that are extracted from new images. Jeon et al. [2003] proposed Cross-Media Relevance Model (CMRM) for image annotation. The work relies on normalized cuts to segment images into regions then build blobs. Here, they consider blobs as quantized features or visual terms. The model uses training images as latent variables to estimate the joint distribution between blobs and words. Continuous relevance model (CRM) [57] is also a relevance model like CMRM, but different from CMRM by the fact that it models directly the joint distribution between words and continuous visual features using non-parametric kernel density estimate. As a result, it is not as sensitive to quantization errors as CMRM. Multiple Bernoulli Relevance Model (MBRM) [28] is similar to CRM except that it is based on another statistical assumption for generating words from images (multiple Bernoulli instead of multinomial distribution). These methods (CMRM, CRM, and MBRM) are also mentioned as keyword propagation methods since they transfer keywords of the nearest neighbors (in the training dataset) to the given new image. One disadvantage of the propagation methods is that the annotation time depends linearly on the number of training set, thus have the scalable limitation [17]. Topic model-based methods [12, 69, 70], on the other hand, do not use training images but hidden topics (concepts/aspects) as latent variables. The methods also exploit either quantized features [70] or continuous variables [12]. The main advantages of the topic model-based methods are the ability to encode scene settings (via topics) [60] and to deal with synonyms and homonyms in annotation.

Multiple Instance Learning (MIL) addresses a special type of learning problems in which there are ambiguities involved during the training [117]. Supervised multiclass labeling (SML) [17] is based on MIL and density estimation to measure the conditional distribution of features given a specific word. SML uses a bag of image examples annotated by particular word (say "mountain"), and estimates the distribution of image features extracted from the bag of images (see Figure 5.2). The distribution is fitted by mixture Gaussian distribution in a hierarchical manner. Since SML only uses positive examples for each concept, the training complexity reduces considerably in

comparison with OVA classification. Stathopoulos et al. [2009] focused on the problem of density estimation and proposed a novel Bayesian hierarchical method for estimating models of Gaussian components. Zhang et. al. [2009] presented a framework on multimodal image retrieval and annotation based on MIL in which they considered instances as blocks in images. Hu et al. [2009] also partitioned images into regions and formulated the problem as semi-supervised learning under multi-instance learning framework by introducing the adaptive geometric relationship between two bags of instances (two images). Other multiple-instance learning based methods extend Support Vector Machine (SVM) [2, 15] to reduce the influence of noise in learning. MIL is suitable to cope with the "weakly labeling" problem in image annotation, but the disadvantage of current MIL-based methods is that they consider words in isolation while context plays important role in reducing annotation error.

5.3 The Proposed Method

5.3.1 Problem Formalization and Notations

Image annotation is an automatic process of finding appropriate semantic labels for images from a predefined vocabulary. This problem can be formalized as a machine learning problem with the notations as follows:

- $V = \{w_1, w_2, \ldots, w_{|V|}\}$ is a predefined vocabulary of words.
- An image I is represented by a set of feature vectors $X_I = \{\vec{x}_{I1}, \ldots, \vec{x}_{IB_I}\}$, in which B_I denotes the number of feature vectors of I and $\vec{x}_{Ij}$ is a feature vector.
- Image I should be annotated by a set of words $W_I = \{w_{I1}, \ldots, w_{IT_I}\}$. Here, T_I is the number of words assigned to image I, and w_{Ij} is the j-th word of image I selected from V.
- A training dataset $D = \{I_1, I_2, \ldots, I_N\}$ is a collection of annotated images. That means every I_n has manually assigned to a word set W_{I_n}. On the other hand, I_n is also represented by a set of feature vectors X_{I_n}. For simplicity, we often use $W_n = W_{I_n}$ and $X_n = X_{I_n}$ to indicate word set and feature set of image I_n in the training dataset.

Based on V and the training dataset D, the objective is to learn a model that automatically annotates new images with words (in V).

5.3.2 The General Framework

The overview of our method is summarized in Figure 5.3.2. As we can see from the figure, the training step consists of two stages:

1. Estimating feature-word distribution: Feature vectors of images along with their captions in the training dataset will be exploited to learn feature-word distributions $p(X|w)$ for words in the vocabulary. Depending on learning method, we may obtain $p(X, w)$ (with generative

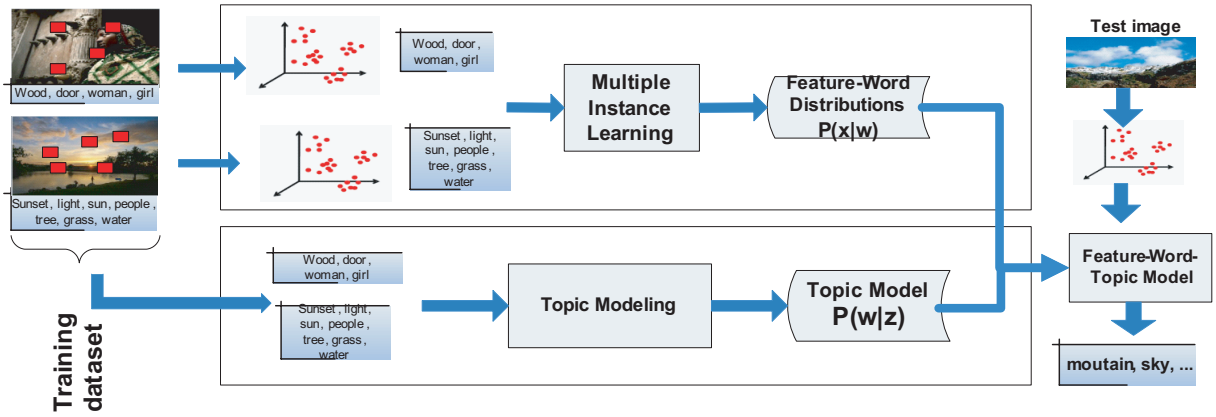


Figure 5.3: Overview of the proposed method for image annotation

model) or $p(w|X)$ (with discriminative model) instead of $p(X|w)$. In either cases, we are able to apply Bayes rule to derive $p(X|w)$:

$$p(X|w) = \frac{p(X,w)}{p(w)} = \frac{p(w|X) \times p(X)}{p(w)} \tag{5.1}$$

2. Estimating word-topic distribution: Word sets associated with images in the training dataset are considered as textual documents and used to build a topic model, that are represented by word-topic distributions. We use that topic model to obtain reasonable combinations of words to form scenes (in form of topics).

In the annotation step, two types of distributions are merged to form a feature-word-topic model for image annotation, in which feature-word distributions are used to define weights of words for topic inference. If feature-word distributions are not obtained directly, we have to apply Bayes rule as in Equation 5.1. In this case, the feature-word distributions are proportional to the output of the learned model ($p(w|X)$ or $p(X,w)$) and reversely proportional to $p(w)$. This is reasonable because we want words with higher confidence values, which are obtained from multiple instance classifiers, contribute more to topic inference and common words (such as "sky", "indoor", etc.), which occurs in many scenes, have less contribution.

In general, we can apply any MIL method and topic model to estimate two types of distributions. For simplicity, we exploited Gaussian Mixture hierarchy [100, 17], which can obtain $p(X|w)$ directly, and pLSA in our implementation of the framework.

5.4 Estimation of Feature-Word Distribution

5.4.1 Feature Extraction

In order to obtain features for annotation and retrieval, we can apply a lot of feature extraction methods [62, 24]. For comparison purpose, we made use of a similar method as SML [17] and pLSA-

based annotation and retrieval [37]. For each image I, a set $\mathbf{X}_I$ of feature vectors are extracted as follows:

1. An image I is represented in YBrCr color space. A set of B_I overlapping 8x8 regions are extracted from I using a sliding window. Note that, one region has three planes, each of which is a square of size 8x8 and in one of three color channels (Y, Br or Cr).

2. For each region $r \in \{1, 2, \ldots, B_I\}$, we applied discrete cosine transform to each of its color channels and kept lower frequencies to obtain 22 coefficients (for Y channel) or 21 coefficients (for Br, Cr channels). We then concatenate all the coefficients to obtain a feature vector $\mathbf{x}_r$ of 64 dimensions.

3. Applying step 2 for all B_I regions of I, we obtain a set $X_I = \{\vec{x}_1, \ldots, \vec{x}_{B_I}\}$ of feature vectors representing I.

5.4.2 Mixture Hierarchies Estimation with MIL approach

The feature-word distribution is estimated based on Mixture Hierarchies [100] and MIL, which is the same as in SML [17]. Here, the model considers feature vectors are generated according to word-conditional distribution $P(\vec{x}|w)$. Given a training dataset, we want to estimate these distributions.

As we have mentioned, Carneiro et al. [17] considered this problem from MIL perspective, which is to learn models from bags of ambiguous examples. A bag is a collection of examples and is considered positive to one label if at least one of those examples is assigned to that label. Otherwise, the bag is negative to that label. The basic idea is that the positive examples are much more likely to be concentrated within a small region of the feature space in spite of the occurrence of negative examples in positive bags. As a result, we can approximate the empirical distributions of positive bags by a mixture of two components: a uniform component of negative examples, and the distribution of positive examples. The consistent appearance of the word-related visual features makes the distribution of positive examples dominate over the entire positive bag (the uniform component has small amplitude).

Hierarchical Model

Let D_w be the subset of D containing all the images labeled with w, the distribution $P(\vec{x}|w)$ is estimated from D_w in a two-stage procedure:

- For each image I in D_w, we estimate a Gaussian mixture $\{\pi_j^I, \vec{\mu}_j^I, \Sigma_j^I | j = 1, \ldots, C\}$. We thus obtain a set of $|D_w|C$ image-level components. The mixing parameters π^I are summed and normalized among $|D_w|C$ components to obtain $\mathcal{M}^{im} = \{\pi_j^{im}, \vec{\mu}_j^{im}, \Sigma_j^{im} | j = 1, \ldots, |D_w|C\}$ - a collection of image-level densities where $\sum_j \pi_j = 1$.

- On the second stage, we would like to cluster the image-level densities into a mixture of L components at word-level $\mathcal{M}^w = \{\pi_i^w, \vec{\mu}_i^w, \Sigma_i^w | i = 1, \ldots, L\}$, where L is the number of components desired at the word-level.

Here, $\vec{x}, \vec{\mu}_j^{im}$ and $\vec{\mu}_j^w$ are vectors in the feature space of 64 dimensions as described in previous section. Covariance matrices $\Sigma_j^{im}, \Sigma_i^w$ are of size 64×64. The probability of drawing $\vec{x}$ from $\mathcal{M}^w$ is expressed as follows:

$$p(\vec{x}|\mathcal{M}^w) = \sum_{i=1}^{L} \pi_i^w p(\vec{x}|z^w = e_i^w, \mathcal{M}_i^w). \tag{5.2}$$

where e_i^w is a canonical basic of R^L and z^w is an indicator vector that $z^w = e_i^w$ if $\vec{x}$ is sampled from $i-th$ component of $\mathcal{M}^w$. Similarly, we have the probability of drawing $\vec{x}$ from $\mathcal{M}^{im}$ is calculated as follows:

$$p(\vec{x}|\mathcal{M}^{im}) = \sum_{i=1}^{|D_w|C} \pi_i^{im} p(\vec{x}|z^{im} = e_i^{im}, \mathcal{M}_i^{im}). \tag{5.3}$$

where e_i^{im} is a canonical basic of $R^{|D_w|C}$ and z^{im} is an indicator vector with one element equals 1, which corresponds to one of $|D_w|C$ components of $\mathcal{M}^{im}$, and the other elements equal 0.

Models of 2 levels are related by a *permutation matrix* P (of size $|D^w|C \times L$) such that $z^{im} = P \times z^w$. Here, $p_{ij} = 1$ if component i-th of $\mathcal{M}^{im}$, i.e. $(\pi_i^{im}, \vec{\mu}_i^{im}, \Sigma_i^{im})$, is a copy of j-th component of $\mathcal{M}^w$, i.e. $(\pi_j^w, \vec{\mu}_j^w, \Sigma_j^w)$.

$$p(\vec{x}|z^{im} = e_i^{im}, p_{ij} = 1, \mathcal{M}_i^{im}) = p(\vec{x}|z^w = e_j^w, \mathcal{M}_j^w) \tag{5.4}$$

This condition is sufficient enough to guarantee the consistency of the hierarchical representation [100].

Propagating parameters

Given definitions and notations above, we are able to draw independent samples $\{(\vec{x}_m, z_m^{im})\}_{m=1}^N$ from $p(x|\mathcal{M}^{im})$. These samples can be grouped into the sequence $\{(\hat{X}_i, e_i^{im})\}_{i=1}^{|D_w|C}$ where $\hat{X}_i = \{(\vec{x}_m, z_m^{im} = e_i^{im})\}_{m=1}^{N_i}$ are samples drawn from component i-th of $\mathcal{M}^{im}$. We evaluate parameters of word-level by maximizing the likelihood of the samples $\hat{X} = \{\hat{X}_1, \ldots, \hat{X}_{|D_w|C}\}$ under the model of word level:

$$p(\hat{X}|\mathcal{M}^w) = \prod_{i=1}^{|D_w|C} p(\hat{X}_i|\mathcal{M}^w) \tag{5.5}$$

We use Expectation-Maximization algorithm, which iterates between E-step and M-step, for this problem. The E-step computes the assignment of $\hat{X}_i$ which are samples from $\mathcal{M}_i^{im} = (\pi_i^{im}, \vec{\mu}_i^{im}, \Sigma_i^{im})$ to $\mathcal{M}_j^w = (\pi_j^w, \vec{\mu}_j^w, \Sigma_j^w)$:

$$\begin{aligned} h_{ij} &= p(z^w = e_j^w|\hat{X}_i, z^{im} = e_i^{im}, \mathcal{M}^w) \\ &= \frac{p(\hat{X}_i|z^w = e_j^w, \mathcal{M}^w)\pi_j^w}{\sum_k p(\hat{X}_i|z^w = e_k^w, \mathcal{M}^w)\pi_k^w} \end{aligned}$$

The inference in [100] leads to the following update in E-step:

$$h_{ij} = \frac{\left[\mathcal{G}(\vec{\mu}_j^{im}, \vec{\mu}_i^w, \Sigma_i^w) \exp(-1/2 \text{ trace}\{(\Sigma_i^w)^{-1} \Sigma_j^{im}\})\right]^{\pi_j^{im}\mathcal{N}_j} \pi_i^w}{\sum_{k=1}^K \left[\mathcal{G}(\vec{\mu}_j^{im}, \vec{\mu}_k^w, \Sigma_k^w) \exp(-1/2 \text{ trace}\{(\Sigma_k^w)^{-1} \Sigma_i^{im}\})\right]^{\pi_j^{im}\mathcal{N}_j} \pi_k^w}$$

where $\mathcal{G}(\vec{x}, \vec{\mu}, \Sigma)$ is a Gaussian with mean μ and covariance Σ, and $\mathcal{N}_j$ is the number of pseudo-sample drawn from each image-level component, which is set to 1 as in [17]. For M-step, we maximizes:

$$Q = \sum_{i=1}^{L} \sum_{j=1}^{|D^w|C} h_{ij} log(\pi_j^w p(\hat{X}_i | z^w = e_j^w, \mathcal{M}^w) \tag{5.6}$$

subject to the constraint $\sum_j \pi_j^w = 1$. The Gaussian case leads the following parameters update [100]:

$$\pi_i^w = \frac{\sum_j h_{ij}}{|D_w|C}$$

$$\vec{\mu}_i^w = \sum_j \lambda_{ij} \vec{\mu}_j^{im}, \text{ where } \lambda_{ij} = \frac{h_{ij}\pi_j^{im}}{\sum_j h_{ij}\pi_j^{im}}$$

$$\Sigma_i^w = \sum_j \lambda_{ij} \left[\Sigma_j^{im} + (\vec{\mu}_j^{im} - \vec{\mu}_i^w)(\vec{\mu}_j^{im} - \vec{\mu}_i^w)^T\right]$$

5.5 Estimation of Word-Topic Distribution

Considering word sets of images as small documents, we use pLSA to analyze the combination of words to form scenes. Like pLSA [40, 70] for textual documents, we assume the existence of a latent aspect (topic) z_k ($k \in 1, ..., K$) in the generative process of each word w_j ($w_j \in V$) associated with an image I_n ($n \in 1, ..., N$). Each occurrence of w_j is independent from the image it belongs to. Given K and label sets of images, we want to automatically estimate $Z = \{z_1, z_2, \ldots, z_K\}$. Note that, we only care about annotated words, not visual features in this latent semantic analysis. The conditional independence of pLSA is shown in the graphical model in Figure 5.4b.

- First, an image I_n is sampled with the probability $p(I_n)$, which is proportional to the number of labels of the image.
- Next, an aspect z_k is selected according to the conditional probability distribution $p(z|I_n)$.
- Given the aspect z_k, a word w_j is sampled from the conditional distribution $p(w|z_k)$.

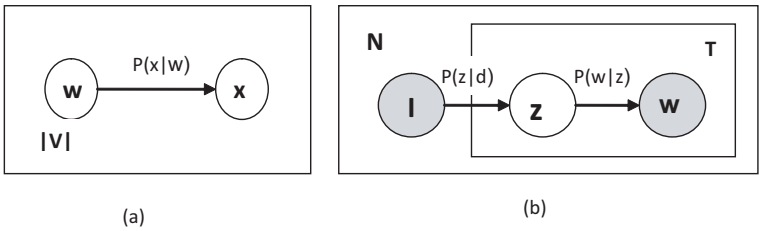


Figure 5.4: (a) Feature-Word model (b) pLSA

We would like to estimate the conditional probability distributions $p(w|z_k)$ and $p(z|I_n)$, which are multinomial distributions and can be considered as parameters of pLSA. We can obtain the distributions by using EM algorithm [70], which is derived from the maximization of the likelihood $\mathcal{L}$ of the observed data

$$\mathcal{L} = \prod_{n=1}^{N} \prod_{j=1}^{|V|} \{p(I_n) \sum_{k=1}^{K} p(z_k|I_n) p(w_j|z_k)\}^{\mathcal{N}(I_n, w_j)} \tag{5.7}$$

where $\mathcal{N}(I_n, w_j)$ is the count of element w_j assigned to image I_n. The two steps of the EM algorithm are described as follows [40]:

- E-step. The conditional probability distribution of the latent aspect z_k given the observation pair (I_n, w_j) is updated to a new value from the previous estimate of the model parameters:

$$p(z_k|I_n, w_j) \leftarrow \frac{p(w_j|z_k)p(z_k|I_n)}{\sum_{k'=1}^{K} p(w_j|z_{k'})p(z_{k'}|I_n)} \tag{5.8}$$

- M-step. The parameters of the multinomial distribution $p(w|z)$ and $p(z|I_n)$ are updated with the new expected values $p(z|I, w)$:

$$p(w_j|z_k) \leftarrow \frac{\sum_{n=1}^{N} \mathcal{N}(I_n, w_j) p(z_k|I_n, w_j)}{\sum_{m=1}^{|V|} \sum_{i=1}^{N} \mathcal{N}(I_i, w_m) p(z_k|I_i, w_m)}, \tag{5.9}$$

$$p(z_k|I_n) \leftarrow \frac{\sum_{j=1}^{|V|} n(I_n, w_j) p(z_k|I_n, w_j)}{\mathcal{N}(I_n)} \tag{5.10}$$

Here, $\mathcal{N}(I_n)$ is the total number of words assigned to I_n.

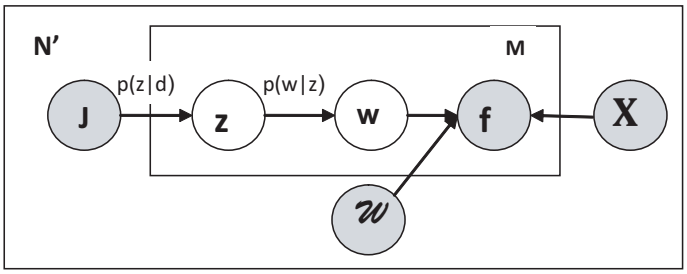


Figure 5.5: Feature-Word-Topic Model for Image Annotation. Here, N' is the number of images in the testing dataset

5.6 Feature-Word-Topic Model for Image Annotation and Retrieval

5.6.1 Feature-Word-Topic Model

Generative Model

The Feature-Word-Topic model (FWT) for annotation is represented in Figure 5.5. Suppose that there exist a set of (distinguishable) visual representations $\{g_1, g_2, \ldots, g_{|V|}\}$ determined by the occurrences of words $\{w_1, w_2, \ldots, w_{|V|}\}$ in the vocabulary V. However, for any given image, because of the feature extraction method and a set of likely co-occurred labels $\mathcal{W}$ ($\mathcal{W} \in V$), we only observe noisy occurrences of g_w. The generative model is described in the following:

- For each image, a set $\mathcal{W}$ of M most likely words are sampled according to topic distribution of image J.
- For each word w of M most likely words, generate the (noisy) visual observation f_w (of g_w). Here, we consider each f simply as one copy of X. However, if J is divided into regions, we can consider f as the part of X corresponding to one specific region in the image.

The observation/hidden states of variables of the graphical model for the training and the testing (or new) images are described as follows:

- The training images have w, J, X, $\mathcal{W}$, and f observed. From the model, we see that the observed w blocks the way from z to f. In the other words, the topic part is independent of the feature part given word. By ignoring the feature part, the word-topic distributions are estimated as in Section 5.5 to obtain $p(w|z)$. For the feature-part, since each f is one copy of X, we define the assumption as follows:

$$p(f_{1:M}|w, X, \mathcal{W}) = p(f_i|w, X, \mathcal{W}) = \begin{cases} \psi(X, w, \mathcal{W}) & w \in \mathcal{W} \\ 0 & otherwise \end{cases} \tag{5.11}$$

This definition reflects the noises caused by the multiple-instance nature of images. Here, $\psi(X, w, \mathcal{W})$ is a weighting function of $p(X|w)$, which is estimated as in Section 5.4, and $\mathcal{W}$.

$$\psi(X, w, \mathcal{W}) \propto p(X|w) - min\{p(X|w_m)|w_m \in \mathcal{W}\} \tag{5.12}$$

This weighting function ψ preserves the ranking order of candidates in $\mathcal{W}$ while emphasizing higher ranking words in topic inference, ψ is normalized to make $\sum_{w\in\mathcal{W}} \psi(X, w, \mathcal{W}) = 1$.

- In the testing images, we have J, X, f observed. $\mathcal{W}$ is indeed unobserved, but we make it observed by selecting a set $\mathcal{W}$ of M candidate words with highest values of $p(X|w) = \prod_{i=1}^{B_J} p(\vec{x}_i|w)$ where $\vec{x}_i$ is a feature vector of image J. In this paper, we fixed M to 20. We now want to infer hidden variables w given the observed variables for image annotation.

In testing, the definition in Equation 5.11 ensures that we only select words w from $\mathcal{W}$ instead of the whole vocabulary V. Here, each w is one word index (from $\mathcal{W}$), and the model works as we sample M times from a multinominal distribution, which is parameterized with $\psi(X, w, \mathcal{W})$, over $\mathcal{W}$ but the selection of w also be controlled by the topic distribution of the whole image J.

Inference

Given the model defined in Figure 5.5 and a new image J while fixing $p(w|z)$ and $p(X|w)$ from word-topic estimation and feature-word estimation, an EM algorithm is used to obtain $p(z_k|J)$ for $k = 1, 2, \ldots, K$. Since each f is one copy of X, we can replace each f_m by X . The EM starts with an initiation and iteratively run through E-step and M-step until convergence.

- E-step updates posterior distributions:

$$p(z_k, w_m|J, X, \mathcal{W}) \leftarrow \frac{p(z_k|J) \times p(w_m|z_m) \times \psi(X, w_m, \mathcal{W})}{Z} \tag{5.13}$$

where $Z = \sum_{k'} \sum_{w'_m\in\mathcal{W}} p(z'_k|J)p(w'_m|z'_k)\psi(w'_m, X, \mathcal{W})$

- M-step maximizes the expectation of complete log likelihood $\mathcal{L}^c$ with respect to posterior distribution (from E-step). Denote $\mathcal{E} = E_{p(w,z|J,X,\mathcal{W})} \log \mathcal{L}^c$, we have:

$$\begin{aligned}
\mathcal{E} &= \sum_{z_k} \sum_{w_m\in\mathcal{W}} p(z_k, w_m|J, X, \mathcal{W}) \log p(J, z_k, w_m, \mathcal{W}, X) \\
&\propto \sum_{z_k} \sum_{w_m\in\mathcal{W}} p(z_k, w_m|J, X, \mathcal{W}) \times \\
&\qquad \{\log p(z_k|J) + \log p(w_m|z_k) + \log \psi(X, w_m, \mathcal{W})\}
\end{aligned}$$

Maximizing $\mathcal{E}$ with the constraint that $\sum_{z_k=1}^{K} p(z_k|J) = 1$, we obtain:

$$p(z_k|J) \leftarrow \sum_{w_m \in \mathcal{W}} p(w_m, z_k|J, X, \mathcal{W}) \tag{5.14}$$

After EM algorithm converged, we obtain $p(z_k|J)$ $(k = 1, \ldots, K)$ for image J. For each $w \in \mathcal{W}$, we calculate:

$$\begin{aligned} p(w|J, X, \mathcal{W}) &\propto \sum_{z_k} p(w, z_k, X, J, \mathcal{W}) \\ &= \sum_{z_k} p(z_k|J) \times p(w|z_k) \times \psi(X, w_m, \mathcal{W}) \\ &= \psi(X, w, \mathcal{W}) \sum_{z_k} p(z_k|J) \times p(w|z_k) \end{aligned} \tag{5.15}$$

We then rank w in $\mathcal{W}$ by $p(w|J, X, \mathcal{W})$ for image annotation. As you can see from Equation 5.15, words with higher feature-word probabilities via $\psi(X, w, \mathcal{W})$ and highly contribute to emerging topics of the scene will result in higher ranks in the new ranking list.

5.6.2 Time Analysis

We compare time complexity of our proposed method with SML, which is based on the same feature-word distributions but does not consider topic modeling. For annotating one image, SML requires $O(BL|V|)$ in which B, L and $|V|$ are the number of feature vectors (of the given image), the number of Gaussian components at word-level and the vocabulary size. Our method needs $O(BL|V| + MKe)$ (e is the number of EM iterations in Section 5.7, and K is the number of topics). In real-world dataset, since $BL|V|$ is usually much larger than MKe , the extra time for topic inference is relatively small. For example, $BL|V| \approx 6,000 \times 32 \times 292$ and $MKe \approx 20 \times 100 \times 30$ in our experiments given in Section 5.7.

5.6.3 Comparison with Related Approaches

Supervised Multiclass Labeling

As mentioned earlier, our method estimates feature-word distribution based on mixture hierarchies and MIL, which is the same as SML [17]. The difference of our approach compared with SML is the introduction of latent topics in the annotation. For annotating a new image J with SML, words are selected based on $p(w|X)$ calculated as follows:

$$p(w|X) \propto p(X|w) \times p(w) \tag{5.16}$$

From the equations 5.15 and 5.16, we see that SML only integrates word frequencies (from training dataset) into image annotation but our method considers word relationships (via topics).

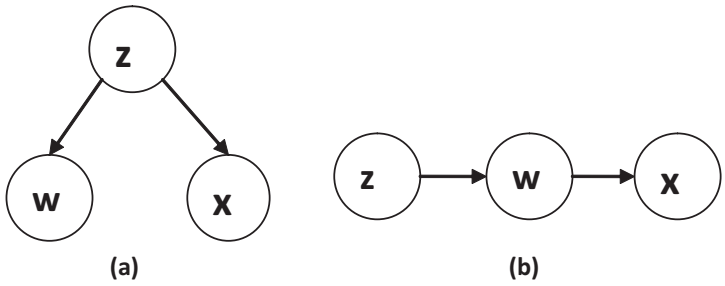


Figure 5.6: The difference of our method in comparison with other topic-based approaches: (a) Other approaches; (b) Our method

Topic Models for Image Annotation

There were a lot of applications of topic models, which are originated from text mining, in image-related problems. Most of the current approaches model directly topic-feature distributions [12, 41, 42, 69, 70, 60, 106]. If continuous features are used [12, 42], topic estimation becomes very complicated and expensive (in terms of time complexity) since the feature space is very large in comparison with word space. If features are clustered to form discrete visual-words [41, 60, 69, 106], the clustering step on a large dataset of images is also very expensive and may reduce the annotation performance [49]. Moreover, the indirect modeling of visual features and labels make it harder to guarantee annotation optimization [17]. Topics of features are also hard to interpret than topics of words.

The difference of our method from previous approaches is that we model topics via words, not words and features (see Figure 5.6). As a result, we do not need to modify topic models for training, where captions are available. To infer topics for an unannotated image, we only need to consider weights based on $p(\mathbf{x}|w)$ instead of word occurrence in the original models. Since feature-word distribution for a concept is estimated using a subset of the training dataset, it is more practical in comparison with visual-word construction. Moreover, the separation of feature-word distribution and word-topic distribution makes it easier to optimize the performance. For example, if we already have good models for recognizing some of the concepts in the vocabulary such as "tigers", "face", we can replace those models to obtain more-confident $p(X|\text{“}tigers\text{''})$ or $p(X|\text{“}face\text{''})$, which improves the final ranking in Equation 5.15. Similarly, we can construct more suitable topic model, which deals with the sparseness of words per scene, but still be able to reuse the whole feature-word distributions.

Modeling Word Relationships for Annotation

In order to incorporate the relationships between labels to reduce annotation error, most of previous works are based on word-to-word correlations [61, 50, 84, 72] or fixed semantic structures such as Wordnet [51, 108]. Among the methods, Coherent Language Model [50] is noticeable because it enables us to estimate the length of annotation. These methods can also be roughly categorized into two classes: 1) Post-processing or annotation refinement [61, 108, 72] in which word-to-word

relationships are used to refine candidates generated from a base annotation method; and 2) Correlative labeling [50, 84] in which word-to-word relationships are integrated to annotate images in a single step. The disadvantage of the refinement approach is that the errors incurred in the first step can propagate to the second fusion step [84]. On the other hand, the correlative labeling approach is more expensive because the number of word combination is exponential to the size of the vocabulary. As a result, it limits the extension of the annotation vocabulary.

Among the approaches that make use of word relationships in annotation, our method falls into the refinement category. The difference of our method is that we make use of topic model to capture word relationship rather than word-to-word correlations or fine-constructed semantic structures like Wordnet. As a result, we are able to extend the vocabulary easier and take the current advances of topic modeling in text. Even that we use pLSA for topic estimation, other topic models can be used to capture stronger relationships such as Correlative Topic Model (CTM) [10], in which the presence of one topic {sand, ocean, sky, dune} may lower the probability of another topic like {sand, desert, dune, sky}.

5.7 Experiments

5.7.1 Experimental Dataset

UWDB Dataset

UWDB is a freely available benchmark dataset for image retrieval, which is maintained at University of Washington[1]. This dataset contains 1109 images that are classified into categories like "spring flowers", "Barcelona" and "Iran". They also provide an uncategorized dataset containing 636 landscape images. All of those images are annotated with captions. For experiments in this paper, we obtain color images from UWDB and resize them to 40% of original size, which results in images with size 300×200. For image labels, we performed a small preprocessing by spell checking, tense transformation (such as "trees" to "tree"). Finally, we obtained 1490 images (of size 300×200) annotated with 292 unique words for experiments. The maximum of words per image is 22, and the minimum is 1. On average, we have 4.32 captions per image.

Corel5k Dataset

The Corel5k benchmark is obtained from Corel image database and commonly used for image annotation [27, 17, 37]. It contains 5,000 images from 50 Corel Stock Photo CDs and were divided into a training set of 4,000 images, a validation set of 500 images, and a test set of 500 images. The validation set can be used to tune parameters such as the number of topics K, after which, the training and validation sets can be merged to form a new training set. Each image is labeled with 1 to 5 captions from a vocabulary of 374 distinct words. On average, one image has 3.22 captions.

[1] www-i6.informatik.rwth-aachen.de/~deselaers/uwdb

5.7.2 Experimental Settings

We performed training as described in Section 5.3, 5.4, and 5.5. Here, we set $C = 4; L = 32$ and K varies from 10 to 250. Image annotation for test dataset is performed as described in Section 5.6, where we set $M = 20$. We compared our method with SML that is based on feature-word distributions given in Section 5.4 with the same values of C and L.

5.7.3 Evaluation Methods

The annotation performance is measured using mean Average Precision (mAP). Here, for one image, we compare the ranking list of words, which are automatically generated, with the truth manually assigned by human. The main idea is that a relevant word at higher rank will give more credits than lower ranks. More specifically, we calculate the average precision (AP) for one image as follows:

$$AP = \frac{\sum_{r=1}^{M} P(r) \times rel(r)}{\text{Number of manual labels of the image}}$$

where r is a rank, $M = 20$ is the cutoff threshold, $rel(r)$ is a binary function to check the word at r is in the manual list of words or not, and $P(r)$ is the precision at r. Note that, the denominator of AP is independent with the cutoff threshold. Finally, mAP is obtained by averaging APs over all images in the testing dataset.

Beside annotation evaluation, we also performed retrieval evaluation by making use of the label-based mAP, which is similar to [17, 28, 37, 70]. For each image, top words are indexed using probabilities of those words generated by image annotation. Given a single-word query, the system returns all images annotated with that word, ordered by probabilities. Label-based mAP is defined as the average precision over all queries, at the ranks, where recall change.

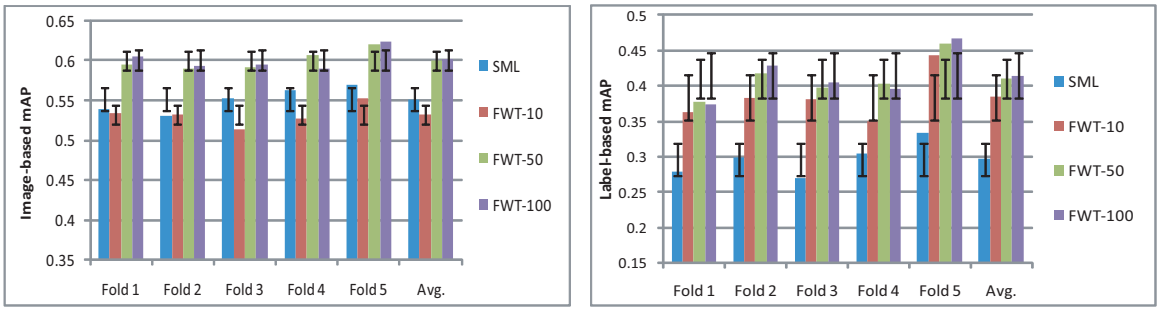
5.7.4 Experimental Results and Analysis

UWDB Dataset

UWDB dataset is evaluated using 5-fold-cross validation. That is we divide dataset into 5 folds, each of which contains 149 images, and in turn take one fold for testing and the rest for training.

The results from 5 folds are reported in Figure 5.7(a) and Figure 5.7(b), where the error bars imply the standard variations over 5 folds. Figure 5.7(a) shows image-based mAPs on 5 folds of UWDB. It can be observed that FWT-50 and FWT-100 outperform SML, in which FWT-100 increases 9.8% of image-based mAP on average. Regarding FWT-100, the improvement of image-based mAP varies from 5.6% on fold 4 to 13.1% on fold 1. The values of label-based mAP are shown in Figure 5.7(b). Here, we indexed top 20 words for each image based on word-image probabilities for label-based mAP evaluation. It can be seen from Figure 5.7(b) that FWT-K models achieve gains from 29.4% (K=10) to 39.6% (K=100) on average.

On the other hand, Figure 5.7(a) and Figure 5.7(b) lead to an interesting observation that although FWT-10 is a little worse than SML according to image-based mAP, it improves SML considerably (29%) with respect to label-based evaluation. This might be due to the ambiguity



(a) Image-based mAP (b) Label-based mAP

Figure 5.7: (a) Image-based mAPs on 5 folds of UWDB evaluated with SML, and FWT with different number of topics (K=10, 50, 100). (b) Label-based mAPs on 5 folds of UWDB evaluated with SML, FWT with different number of topics (K=10, 50, 100)

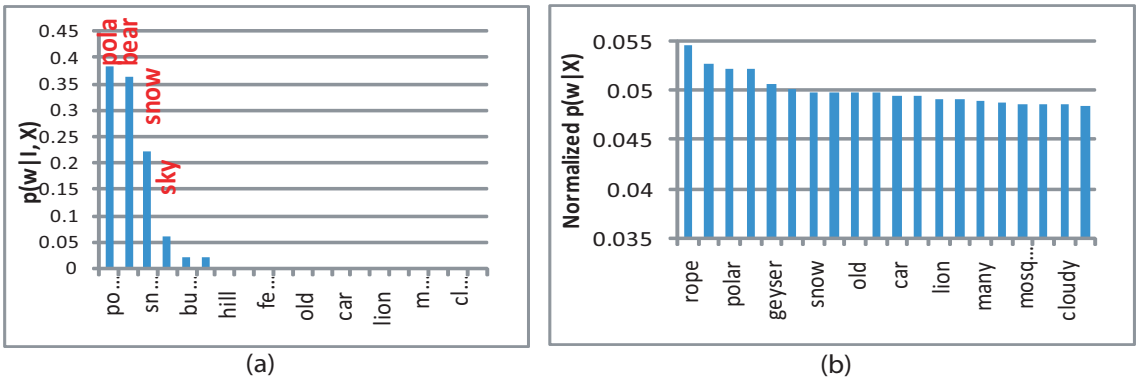


Figure 5.8: The conditional probability distributions top 20 words inferred from the image of "polar bear" in Figure 5.9 in (a) FWT-10; and (b) SML

caused by negative examples in positive bags in learning feature-word distributions. Since SML excludes negative bags in learning feature-word distributions, the discriminative power of SML is lower than other MIL methods. Consequently, probabilities of top 20 words generated by SML for an image forms a near-uniform distribution (see Figure 5.8). When we index images based on top 20 words, the ambiguity becomes more severe across images with SML. In contrast to that, top topic-consistent words, which are generated by FWT, appear to receive much larger probabilities than the rest of words in top 20 candidates (Figure 5.8(b)). This observation suggests that we are able to automatically determine the length of annotation with FWT. Some demonstrative examples of annotation results on UWDB dataset are shown in Figure 5.9. These examples show that our method is able to annotate images with more topic-consistent words.

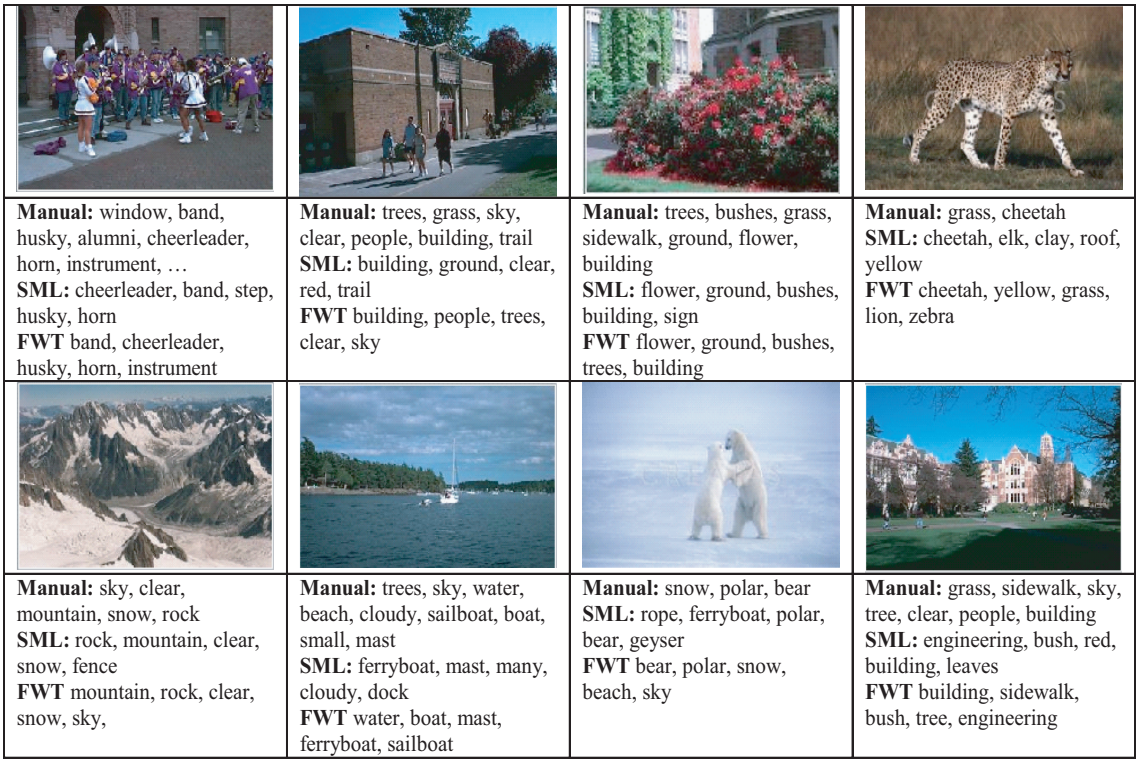
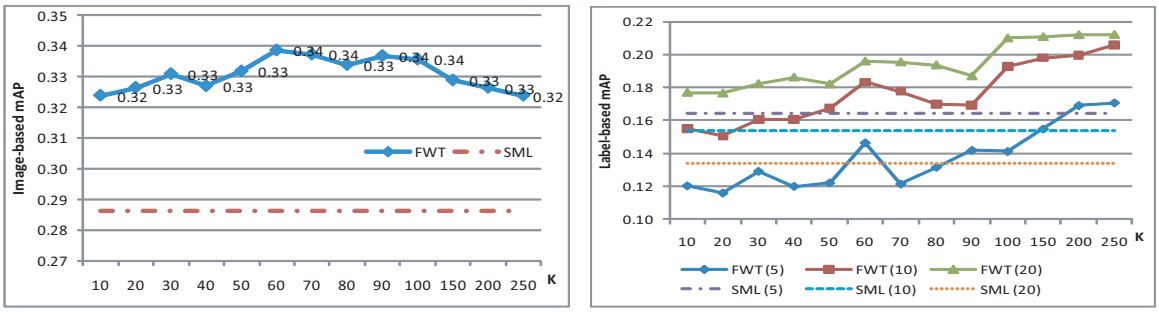


Figure 5.9: UWDB- Examples of Image Annotation with SML and FWT (K=100)

Corel5k Dataset

We evaluated performance of FWT when changing K in Figure 5.10(a) and Figure 5.10(b). Here, we fixed feature-word distributions and trained a number of topic models to annotate with FWT. For each K, due to the random nature of EM algorithm, 10 attempts were conducted and the avarage mAP were obtained and shown in the figures. Since the standard deviations are small (less than 10^{-4}), they have not been shown in the figures.

Figure 5.10(a) demonstrates performance of different FWT models on Corel5k when changing the number of topics K. Overall, FWT models obtain better image-based mAPs than SML on all settings. However, we can observe that the improvement on image-based mAP is bounded by a threshold, which is certainly the best solution to rerank top 20 candidates generated from feature-word distribution. Figure 5.10(b) presents label-based mAPs of SML and FWT models on Corel dataset. Although image-based mAPs are bounded, the number of topics affect indexing quality and thus retrieval performance. It is observable that the larger the number of topics is, the better result we can obtain with FWT. We investigated how the number of top words being indexed for each image affect the retrieval performance of SML and FWT. Since probabilities of top words assigned by SML are not much different from each other as analyzed in the previous section, the smaller number of words being indexed per image provide the better retrieval results for SML. FWT, on



(a) Image-based mAP

(b) Label-based mAP

Figure 5.10: (a).Image-based mAPs on Corel dataset evaluated with SML, FWT with different number of topics K, (b). Label-based mAPs on Corel dataset evaluated with SML, FWT with different number of topics K. The numbers inside brackets indicate the number of indexed words per image. For example: SML(5) means we obtained top 5 words per image for indexing.

Method	Label-based mAP
SML (our implementation)	0.164
pLSA-mixed [37]	0.141
pLSA-words [37]	0.162
FWT (K=250)	0.212

Table 5.1: Retrieval Results of some r elated methods and FWT reported on Corel5K

the other hand, has better results when the number of words being indexed per image is larger. Noticeably, FWT(5) is worse than SML(5) except when the number of topics is large (K=250). This is because the small number of topics brings more bias towards popular labels such as “sky”, “cloud”, or “water”. Those popular words are so obvious that they are not included as captions in some cases. For example, a lot of images with “pool” caption do not contain “water” as their captions even “pool” and “water” are topic-consistent. Because the number of popular words is much smaller than the number of less popular words, FWT(5) is consequently worse than SML(5). When we increase the number of top words to be indexed, the less popular words have chance to be selected with FWT, hence the label-based mAPs of FWT(10) and FWT(20) are higher than FWT(5) and even SML(5). A better strategy to weigh less popular words more than popular ones in topic estimation and inference can help to overcome this situation. For FWT, further studies can be conducted to estimate the length of topic-consistent annotation instead of fixed annotation length in most of current studies.

Table 5.1 summarizes significant results obtained in our implementation of SML, FWT-K models, and two models of joint distribution of words and features based on pLSA in [37]. Note that these methods used DCT-based feature selection and were tested on Corel5k. In comparison with these baselines, FWT shows promising improvement. Although optimizing feature-word distributions is not the main topic of this paper, the better implementation of feature-word distributions

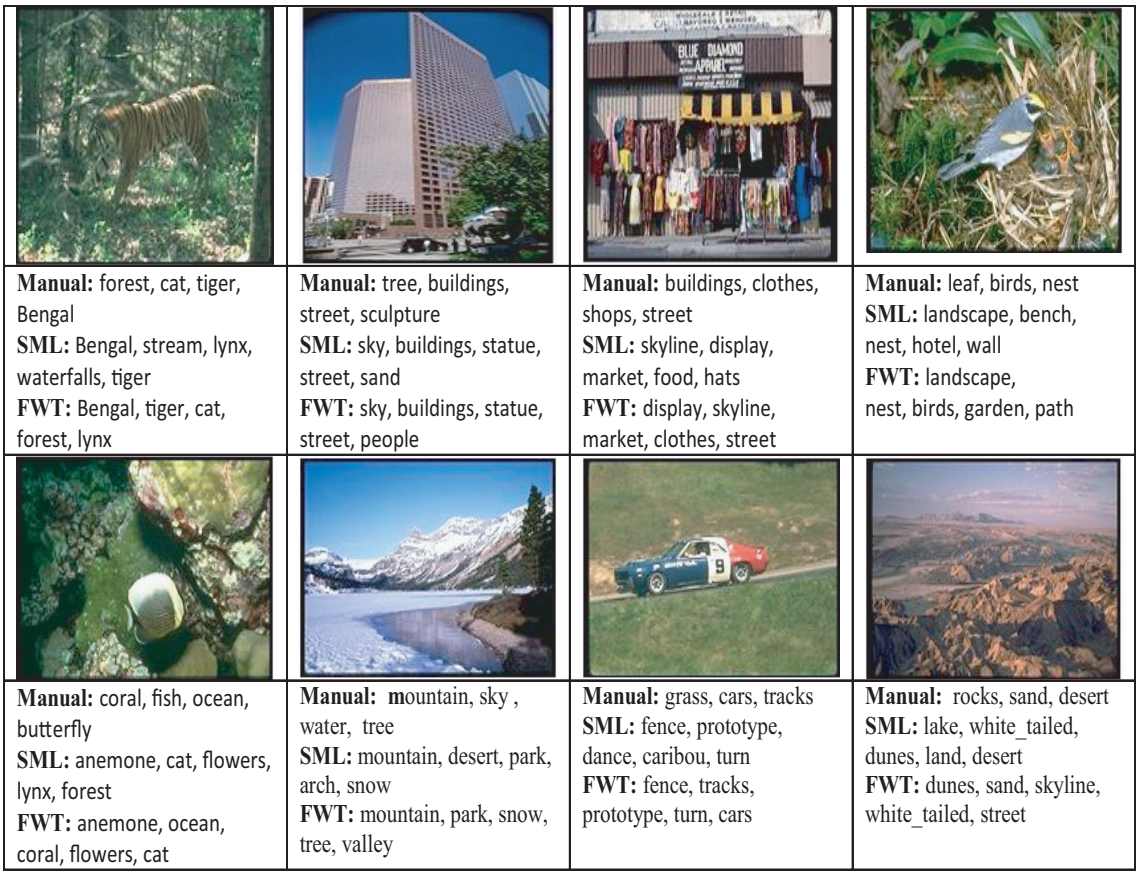


Figure 5.11: Corel5K - Examples of Image Annotation with SML and FWT (K=250)

based on MIL & mixture hierarchies, which leads to better results for SML in [17], is also expected to improve the performance of FWT. Some demonstrative examples of annotation results on Corel5k dataset are shown in Figure 5.11.

5.7.5 How Topics Can Help to Reduce the Semantic Gap

Figure 5.12(a) demonstrates sample topics estimated from the dataset of the training and validation parts of Corel5k, and Figure 5.12(b) shows how topics can be used to improve annotation performance. Based on feature-word distributions, the top 20 candidates are selected and shown in the figure. It can be seen that the visual representation gives some wrong interpretation of the picture, which results in words like “arch”, “guard” or “elephant” at higher positions than more reasonable word like “prototype”. The reasonable interpretation of the picture, however, makes the topic describing the scene (topic 131 in Figure 5.12(a)) surpass the other topics. By taking topics into account, the more reasonable words can be at higher ranking positions than only based

Topic 131	Topic 165	Topic 164	Topic 123	Topic 198	Topic 147	Topic 224	Topic 8
Tracks	Pool	Pyramid	Cat	Sponges	Train	Shops	Log
Cars	People	Stone	Tiger	Coral	Locomotive	Street	Reptile
Turn	Canoe	Tomb	Forest	Ocean	Sky	City	Lizard
Prototype	Race	Ruins	Bengal	Sea	Mountain	People	Snake
Sun	Swimmers	Sun	Sun	Anemone	Railroad	Crafts	Sun
Formula	Water	Sand	Grass	Basket	Vehicle	Sign	Tree

(a) Topics estimated from Corel5k (K=250)

SML: tracks, formula, cars, arch, guard

FWT: cars, tracks, formula, straightaway, prototype

Emerging topic: Cars, turn, tracks, formula, straightaway, prototype, sky, grass

Candidates: tracks formula cars arch guard seals elephant dock wall ice bulls elk steps rock-face prototype baby straightaway snow mist boats

After Re-ranking with topics: cars tracks formula straightaway prototype arch wall elk bulls snow boats steps seals ice city mountain sky sun water

(b) Refinement with Topics

Figure 5.12: Demonstration of topics and their influence on image refinement

on features. Due to the "semantic gap", the visual representation is not good enough for image annotation. We need more "semantic" from scene settings to infer reasonable labels. More examples of annotation on Corel5k given in Figure 5.11 show that we are able to make the "semantic gap" smaller.

5.7.6 When Topics Are Not Much Helpful

The performance of FWT depends on the quality of feature-word distributions and topic models. Here, we weight words with higher ranks from feature-word distributions more than the lower rank ones for topic inference. Certainly, when the top words from feature-word distributions are far from correctness as in Figure 5.13, the estimated topics that only depends on feature-word distributions can not be helpful. Fortunately, images in the Internet usually come with surrounding texts. If we consider the surrounding texts as some types of features, we can have great chance to infer reasonable topics from surrounding text beside those from visual representation. Because image

Manual: city, hills, buildings, landscape
SML: reptile, log, snake, slope, den
FWT: log, reptile, snake, Hawaii, slope

Manual: buildings, reflection, window, statue
SML: sky, mountain, jet, bridge, street
FWT: mountain, sky, bridge, jet, plane

Figure 5.13: Examples from Corel5K when topics based on visual features are not helpful

annotation as well as object recognition are still difficult problems to cope with, combining multiple modalities (surrounding texts, and visual) is a promising solution for image annotation and retrieval.

5.8 Concluding Remarks

The proposed approach is simple and can be extended in several ways. First of all, we can easily adapt to a different topic model to obtain richer word relationships for annotation (such as correlated models, hierarchies models), or a different MIL method thanks to the separation of topic modeling from low-level feature representation. This property brings us a lot of benefits from the recent development of text modeling and MIL approach in image annotation.

Second of all, we can modify the approach to include many types of feature extraction, which has been shown effectively in improving annotation performance as well as object detection [62, 96]. One feature representation can be considered as one view of an image, different views of an image can be used to obtain better annotation. For example, we can train one model of $p_1(\mathbf{x}|w)$ for local feature descriptors (such as SIFT, DCT, and so on), one model $p_2(\mathbf{y}|w)$ for global features (such as contour, shapes). Weighted candidates from different views can be selected, merged and refined for annotation using topics. Considering an image in different views not only help to improve annotation performance but also reduce the time complexity to estimate feature-word distribution. Instead of using feature vectors with large dimension, we can divide them to several types of feature vectors, each of which has smaller dimension.

Finally, we can perform topic modeling using a larger vocabulary, which includes both annotation words and surrounding texts (or name of image files). Since we consider only M selected candidates, which are in the annotation vocabulary, the annotation step works exactly the same as described. Due to computation complexity and the dynamic of human language, the annotation vocabulary is usually limited. By modeling topics for a larger vocabulary, we are able to infer topics based on surrounding texts and features (via feature-word distributions). In fact, the surrounding text may be not enough for searching but can be used as a hint for annotation refined by topics. For example, suppose that we model topics with an extended vocabulary containing “Eiffel”, an

image file name “Eiffel” should increase the probabilities for topics related to “tower”, “city” even if “Eiffel” is not in the annotation vocabulary. This property also allows us to search with queries that are not in the annotation vocabulary.

In the next chapter, we will discuss more about the role of context via global features in image annotation. Also, we will present more carefully significant points to consider when modeling topics for images.

Chapter 6

Cascade of Multi-level Multi-instance Classifiers for Image Annotation

6.1 Introduction

As introduced in Chapter 5, image annotation is an important task to bridge the semantic gap in image retrieval. Although *image classification* and *object recognition* also assign meta data to images, the difference of image annotation from classification and recognition defines its typical challenging issues. In general, the number of labels (classes/objects) is usually larger in image annotation compared to classification and recognition. Because of the dominating number of negative examples, both the one-vs-one and one-vs-all schemes in multi-class supervised learning do not scale very well for image annotation. Unlike object recognition, image annotation is "weakly labeling" [17], that is a label is assigned to one image without indication of the region corresponding to that label. Moreover, scalability requirement prevents researchers investigating feature extraction for every label in image annotation. This, however, can be performed with a limited number of objects in object recognition . On the other hand, the variety of visual representations of objects suggests that we should not depend on one feature extraction method to work well with a large number of labels [1, 63].

Motivated by the aforementioned issues, we propose a new learning method - a cascade of multilevel multi-instance classifiers (CMLMI) for image annotation. The idea behind our approach is that global features best describe the scene and common concepts such as "forest, building, mountain", while finer levels bring useful information to specific objects such as "tiger, cars, bear". Given an object, the cascade method ensures that we first detect the object's related scene, then focus on the "likely" scene to further recognize the object in that context. Formally, cascading means that learning classifiers at finer levels is dependent on classifiers at coarser levels (learning from coarse-to-fine). By so doing, when learning classifiers for specific objects at finer levels, we can ignore (negative) samples of non-related scenes, thus reduce training time. Since negative examples are those of the same scene without the considered object, there is more chance for us to separate the object from the background. For instance, since a "tiger" usually appears in a forest, the negative examples of forest background, which do not contain "tiger", helps recognize "negative"

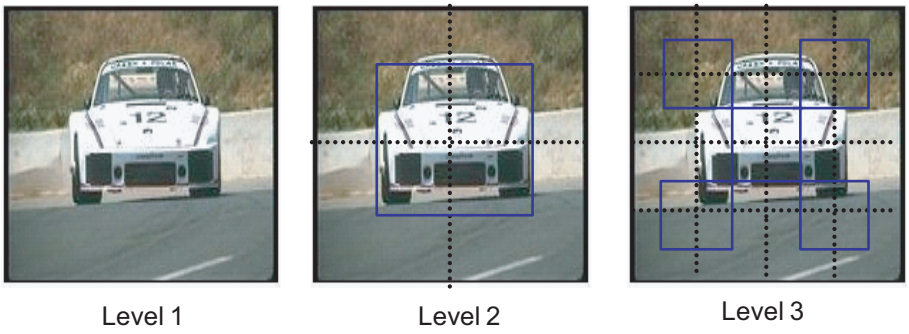


Figure 6.1: Level 1: the whole image; Level 2: 2x2 grid + 1 subregion in the center; Level 3: 4x4 grid + 5 overlapping subregions (blue border rectangles)

regions (forest regions) in the positive examples of "tiger". As a result, it improves the selection of regions corresponding to "tiger", and reduces the ambiguity of "weakly labeling".

Specifically, our propose contains two main parts: 1) multi-level feature extraction; and 2) cascade of multi-instance classifiers over multiple levels. Multilevel means we divide images into different levels of granularity from the coarsest one (the whole image) to increasingly fine (overlapping) subregions (Figure 6.1). Several feature extraction algorithms are performed at each level, each algorithm produces a set of feature vectors corresponding to subregions of the image. Given a label, a cascade of multi-level multi-instance classifiers is then built across levels, from cheapest (coarsest) features to the most expensive (finest) features.

In the literature, cascade of classifiers were successfully used to design fast object detectors [101] and reduce detecting time [87], while multi-level of features were applied to image classification [58] and object recognition [96]. To the best of our knowledge, however, this is one of the first attempts that adopts the hierarchy of multi-level feature extraction to group features according to acquisition cost so as to develop a cascade learning algorithm for image annotation. In comparison with previous cascading algorithms, we take into account the "weakly labeling" problem by using MIL and make the cascading algorithm suitable to image annotation. In addition, our approach is more robust than previous MIL methods because we consider multi-level feature extraction which allows us to cope with the variety in visual representation among labels. The advantages, thus, lie in threefold: 1) reducing training time by a cascade learning algorithm; 2) relaxing the ambiguity of "weakly labeling" problem of image annotation; and 3) obtaining strong classifiers, which are robust to multiple resolution.

The rest of this chapter is organized in 6 sections. Section 6.2 summarizes typical approaches to image annotation and related tasks, which gives a broader view than Chapter 5. Multi-level feature extraction and multi-instance learning are presented in Section 6.3 and Section 6.4. Our proposed method for image annotation is given in details in Section 6.5. Based on the ideas from Chapter 3, we propose a refinement algorithm on label candidates obtained from CMLMI in 6.6 by using Latent Dirichlet Allocation 2 in order to obtain topic-oriented annotation. Experiments are shown in Section 6.7. Finally, Section 6.8 concludes the important remarks of this chapter.

6.2 Related Work

Image annotation and related tasks (object recognition, image retrieval, image classification) have been the active topics for more than a decade and led to several noticeable methods. In the following, we present an overview of typical approaches, which are categorized into 1) classification-based methods; 2) joint-distributions based methods; and 3) ranking approaches.

6.2.1 Classification-based Approach

The early effort in the area is to formalize image annotation as the task of binary classification. Some examples are to classify images into "indoor" or "outdoor" [94]; "city" or "landscape" [99]. In object recognition, Viola and Jones [101] proposed a method for face detection (face/non face classification) using Adaboost, which is very fast in dropping negative windows (non face) in images, thus results in fast face detectors.

For image retrieval, the two-class formalization is not enough to meet searching requirements. Lyndon et al. [54] used a reranking method to combine binary classifiers. Akbas et al. [1] fused binary classifiers by learning a new meta classifier from category-membered vectors, which are generated from the binary classifiers. Wang et al. [107] considered the problem as multilabel classification, in which they first cluster labels into topics, and learn one classifier for one topic. The model then transfers labels of a topic to images at the same time.

In order to apply classification approach to image annotation, we need to take the "weakly labeling" problem into account. Typically, this can be done by adopting multi-instance learning (MIL) instead of single-instance learning. Ansdrew et al. [2] adapted single-instance learning version of Support Vector Machine (SVM) to multi-instance learning versions namely MI-SVM and mi-SVM and applied to image annotation with 3 classes (tiger, fox, elephant). On the other attempt, Yang et al. [111] introduced Asymmetric SVM (ASVM) to pose different loss functions to 2 types of error (false positive and false negative) to improve annotation. ASVM has been applied to 70 common labels of Corel5K, which is the common benchmark for image annotation, and shown comparative results. Also following the idea of MIL but supervised multiclass labeling (SML) [5] does not consider negative examples in learning binary classifiers. Given a label, SML is based on hierarchical Gaussian mixture to train a binary classifier using only positive examples. Since only global features are used in SML, it is not clear whether SML works well for specific objects or not although on average it showed state-of-the-art performance on Corel5K. All in all, MIL-based image annotation systems do not exploit the benefit of combining global and region-based features.

6.2.2 Joint Distribution-based Approach

Statistical generative models introduce a set of latent variables to define a joint distribution between visual features and labels for image annotation. This joint distribution is used to infer conditional distribution of labels given visual features that are extracted from new images. Jeon et al. [48] proposed Cross-Media Relevance Model (CMRM) for image annotation. This work relies on normalized cuts to segment images into regions then clusters visual descriptors of segments to build blobs. CMRM uses training images as latent variables to estimate the joint distribution between

blobs and words. Continuous Relevance Model (CRM) [57] is another relevance model but different from CMRM by the fact that it models directly the joint distribution between words and continuous visual features using non-parametric kernel density estimate. As a result, it is not as sensitive to quantization errors as CMRM. Multiple Bernoulli Relevance Model (MBRM) [28] is similar to CRM except that it relies on another statistical assumption for generating words from images (multiple Bernoulli instead of multinomial distribution). These methods (CMRM, CRM, and MBRM) are also referred as keyword propagation methods since they transfer keywords of the nearest neighbors (in the training dataset) to the given new image. The drawback of those methods is that the annotation time depends linearly on the number of training set, thus have the scalable limitation [17].

Following this approach, topic model-based methods [12, 69, 71, 70] do not use training images but hidden topics (concepts/aspects) as latent variables. These methods also rely on either quantized features [70] or continuous variables [12]. The main advantages of the topic model-based approach lies in two points: 1) the better scalability in comparison with propagation methods; and 2) the ability to encode scene settings (via topics) into image annotation. The disadvantage is its lack of direct modeling between visual features and labels, which makes it difficult to optimize annotation performance.

6.2.3 Ranking Approach

Recently, Jing et al. [53, 52] proposed an algorithm which is similar to pagerank in text retrieval but applied to visual-based similarity graph of images. This approach improves current tag-based image search by providing a better rank and organization of returned images. Since building a similarity graph for the whole database is expensive, the authors partly build similarity graph in a query dependent method that is based on top images returned from an image search engine. For images without tags, solutions have not been addressed in [53, 52]. David et al. [33] proposed a ranking-based image retrieval, which is able to deal with images without tags. For a new image, the system is based on picture kernels, which maps a pair of images (a new image and one manually tagged image in the training set) into R. Although image annotation is not performed in this method, the main principle is similar to propagation methods such as CMRM, CRM, in which they rely on the assumption that similarity in visual representation leads to similarity in label descriptions.

6.3 Multi-level Feature Extraction

As stated previously, our method consists of 2 main parts: 1)multi-level feature extraction; and 2) cascade of multi-instance classifiers over levels. This section reviews some noticeable methods to extract visual descriptors for image annotation, classification and retrieval as the foundation for our multi-level feature extraction described later. We distinguish 3 types of visual descriptors, which are global features, region-based features, and hybrid.

Global feature extraction: an image is not divided into subregions. As a result, we obtain only one feature vector for each image. Many low-level features can be extracted and concatenated

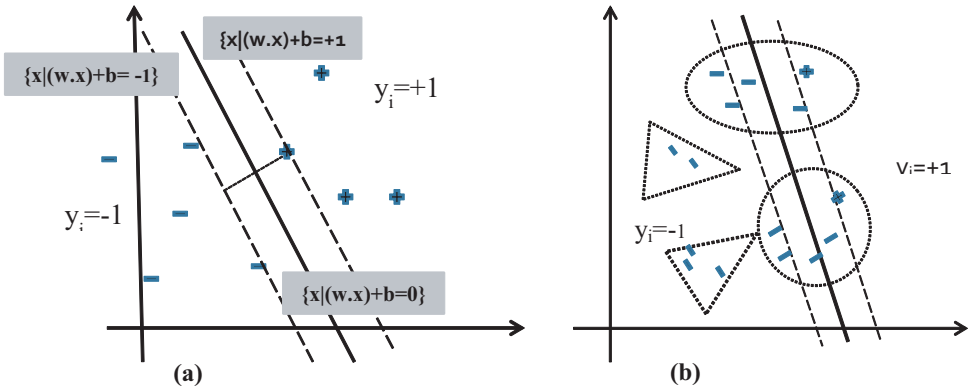


Figure 6.2: Support Vector Machines: (a) Single Instance Learning; (b) Multiple Instance Learning: positive and negative bags are denoted by circles and triangles respectively

from the whole image such as color histogram, texture, or edge histogram [24, 63, 26, 47, 17, 1]. Bag-of-feature [39, 24] obtained by quantizing features at interest points can also be classified to this category because one image is not divided into smaller regions, and an image has only one histogram feature vector. Recent baseline in image annotation [63] also relied on global feature extraction. However, they did not concatenate feature vectors but combined similarities from different feature types to measure similarity between images for K-nearest-neighbor based image annotation.

Local feature extraction: an image is divided into smaller regions using image segmentation [8, 27, 48] or equal division. A feature vector is then extracted from each subregion [28]. As a result, one image has several feature vectors, one corresponds to one subregion. Since image segmentation is still a difficult task, many of current works avoid this task and divide images into grids instead. Some previous studies [28, 49] have shown that equal division can obtain better results than segmentation.

Hybrid method: Spatial pyramid method [58] can be considered as a hybrid of local and global representations. Informally, an image is divided to increasingly coarser grids and concatenate weighted histograms of all cells (in the grid) into a long vector. This method has been applied successfully to scene classification and image classification with little ambiguity, which does not have "weakly labeling" as in image annotation. Even our approach also divides images into different coarse grid (coarse levels) and extract features from levels, the difference is that we do not concatenate the feature vectors from different levels but exploit the hierarchy to group feature sets according to acquisition cost. As a result, we are able to develop a cascade algorithm for image annotation.

6.4 Multi-instance Learning with Support Vector Machines

Multi-instance learning is essential in our propose. This section begins with standard supervised learning with Support Vector Machine (SVM), which is single instance learning, then presents one extension to turn SVM into multi-instance learning (multiple instance SVM).

In standard supervised learning, it is often the case that we are given a training set of labeled instances (samples) $D = \{(\vec{x}_i, y_i)|i = 1, \ldots, N; \vec{x}_i \in R^d; y_i \in \mathcal{Y} = \{+1, -1\}\}$ and the objective is

to learn a classifier, i.e., a function from instances to labels: $h : R^d \rightarrow \mathcal{Y}$. This class of supervised learning belongs to single-instance learning, where Support Vector Machine (SVM) [90] is one of the most successful methods.

Multiple Instance Learning (MIL) generalizes the single instance learning to cope with the ambiguity in training dataset. Instead of receiving a set of labeled instances, we are given a set of negative/ positive bags, each contains many instances. A negative bag contains all negative instances, while a positive bag has at least one positive instance but we do not know which one it is. The formalization of MIL naturally fits the "weakly labeling" in image annotation where a positive bag (w.r.t a label) corresponds to an image annotated with that label. There were several methods for MIL . For simplicity, we will discuss one simple formalization to apply SVM for multiple instance learning namely MI-SVM [2].

6.4.1 Support Vector Machines

In Support Vector Machines [90], a class of hyperplanes that separate negative and positive patterns (Figure 6.2) is considered. For separable case, the hyperplane represented by a pair $(\vec{a}, b)$ ($\vec{a} \in R^N$ and $b \in R$) satisfies:

$$\begin{cases} \vec{a}\vec{x} + b \geq +1 & \text{if } y_i = +1 \\ \vec{a}\vec{x} + b \leq -1 & \text{if } y_i = -1 \end{cases}$$

The corresponding decision function becomes $f(\vec{x}) = sgn(\vec{a}\vec{x} + b)$. Among the hyperplanes that are able to separate positive and negative patterns, the optimal hyperplane is the one with maximum margin and most likely to have minimum test error [90]. It has been proved that the margin of a hyperplane is reversely proportional to $||\vec{a}||$. In practice, a separating hyperplane may not exist, i.e. data is non-separable, slack (positive) variables are introduced to allow misclassified examples. The optimization turns into:

$$\text{minimize: } \frac{1}{2}||\vec{a}|| + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to: } y_i(\vec{a}\vec{x} + b) \geq 1 - \xi_i, i = 1, \ldots, N$$

where C is the constant determining the trade-off. SVMs also can carry out the nonlinear classification by using kernel functions that embed the data into a feature space where the nonlinear pattern now appears linear. Though, we omit the details here, the key aspect of kernel functions is that they preserve the pairwise inner products while relaxing the constraints of coordinates of the embedded points.

6.4.2 Multiple Instance Support Vector Machines

Let $D_w = \{(X_i, Y_i)|i = 1, \ldots, N, X_i = \{\vec{x}_j\}; Y_i = \{+1, -1\}\}$ be a set of images (bags) with/without word w, where a bag X_i of instances $(\vec{x}_j)$ is positive ($Y_i = 1$) if at least one instance $\vec{x}_j \in X_i$ has its label y_j positive (the subregion in the image corresponds to word w). As shown in Figure 6.2b,

positive bags are denoted by circles and negative bags are marked as triangles. The relationship between instance labels and bag labels can be compressed as $Y_i = \max(y_j), j = 1, \ldots, |X_i|$.

MI-SVM [2] extends the notion of the margin from an individual instance to a set of instances (Figure 6.2b). The functional margin of a bag with respect to a hyperplane is defined in [2] as follows:

$$Y_i \max_{\vec{x}_j \in X_i} (\vec{a}\vec{x}_j + b)$$

The prediction is then have the form $Y_i = sgnmax_{\vec{x}_j \in X_i}(\vec{a}\vec{x}_j + b)$. For a positive bag, the margin is the margin of the most positive instance, while the margin of a negative bag is defined as the "least negative" instance. Keeping the definition of bag margin in mind, the Multiple Instance SVM (MI-SVM) is defined as following:

$$\text{minimize: } \frac{1}{2}||\vec{a}|| + C\sum_{i=1}^{N} \xi_i$$

$$\text{subject to: } Y_i \max_{\vec{x}_j \in X_i} (\vec{a}\vec{x}_j + b) \geq 1 - \xi_i, i = 1, \ldots, N, \xi_i \geq 0$$

This optimization can be casted as a mixed-integer program [2]. By introducing to each positive bag X_i a selector variable s_i which denotes the instance selected as the positive "witness" of the positive bag, Andrews et al. has derived an optimization heuristics. The general scheme of optimization heuristics alternates two steps: 1) for given values of selector variables, train SVMs based on selected positive instances and all negative instances; 2) based on current trained SVMs, updates new values of selector variables. The process finishes when no change in selector variables.

6.5 Cascade of Multi-level Multi-instance Classifiers

6.5.1 Notation and Learning Algorithm

Let $\mathcal{D} = \{(I_1, \mathbf{w_1}), \ldots, (I_N, \mathbf{w_N})\}$ be a training dataset, in which $\mathbf{w_n}$ is a set of words associated with image I_n and sampled from a vocabulary $V = \{w_1, w_2, \ldots, w_{|V|}\}$. The objective is to learn a mapping function from visual space to word space so that we can index and rank new images for text-based retrieval. The two main components of our propose are described in the following:

- **Extracting multi-level features**: we divide each image in T different levels then perform M feature extraction algorithms $\mathcal{F}_m$ as in Figure 6.3. Here, we can choose any suitable feature extraction such as color, texture, shape description, gist, etc. for $\mathcal{F}_m$. Let $\mathcal{M}(l)$ $(l = 1, \ldots, T)$ be feature indexes at level l, e.g. $\mathcal{M}(1) = 1, 2; \mathcal{M}(2) = 3, 4, 5$ (Figure 6.3). From this notation, we have $\sum_{l=1}^{T} |\mathcal{M}(l)| = M$. Also, we can infer that all the feature extraction algorithms at previous levels of the level l are indexed from 1 to $\min\{\mathcal{M}(l)\} - 1$.

- **Cascade of multi-instance classifiers over levels**: given a label w, $D_w = \{B^+, B^-\}$ denotes a training dataset where B^+ (B^-) is the set of images with (without) w. Let Y be

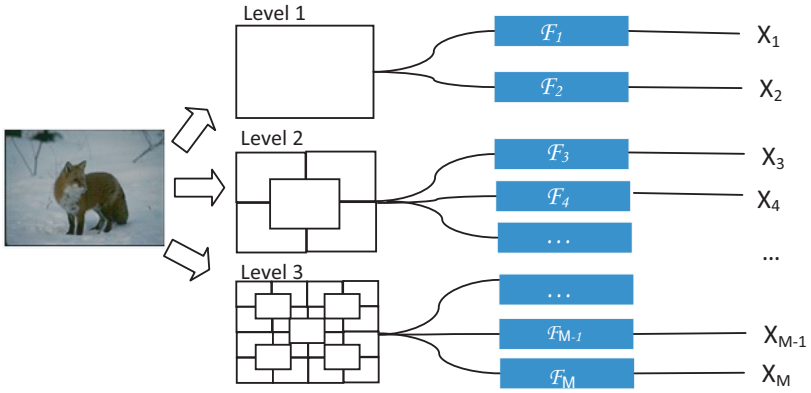


Figure 6.3: An image is divided into different levels of granularity. For each level, we perform one or more feature extraction methods. In total, we obtain M feature extraction methods.

a vector of corresponding classes of images in D_w, i.e. $Y_n = 1$ if $I_n \in B^+$ and $Y_n = -1$ otherwise. Let $\vec{score}$ be the output (confidence) vector generated by machines (classifiers), where $score_n > 0$ (or absolute value of $score_n$ (< 0)) is the confidence of assigning (not assigning) w to $I_n \in D_w$. We denote h_m the weak classifier, which maps from feature space X_m of feature extraction algorithm $\mathcal{F}_m$ to $\{-1, 1\}$. The confidence score posed by h_m on the image I is denoted as $h_m(\mathcal{F}_m(I))$, in which we apply h_m on feature vectors obtained by conducting $\mathcal{F}_m$ on I. Based on these notations, CMLMI is presented in Algorithm 3. Note that multi-instance learning turns into single-instance learning at the coarsest level when global feature vector is in use.

For global feature extractions at level $l = 1$, an image has one instance (one feature vector), the problem turns into normal supervised learning. We applied SVM for this case. At finer level ($l > 1$), one image has a set of instances, one corresponds to one subregion. Due to weakly labeling, we do not know which instance best represents the given label. The multiple-instance version of SVM (MI-SVM) (see Section 6.4) is used to address this ambiguity.

We update scores of images in D_w at level l using the following recursion:

$$score = H_l = \gamma_l * H_{l-1} + \sum_{m \in \mathcal{M}(l)} \lambda_m * h_m + c_l$$

Since we have the constraint that $\gamma_l > 0$, the ranking of images is based on previous ranking (H_{l-1}) but modified by the additional classifiers of current level (the second term). The constant term c_l is used as the constant threshold for level l. We then find coefficients for classifiers of level l using linear regression that is minimizing square error $||H - Y||_2$ (lines from 10 to 11 in Algorithm 3). Here, scores for images in D_w are accumulated from level 1 to level $l - 1$ and stored in $\vec{score}$.

Unlike previous boosting methods, the sampling distribution θ on B^- is updated based on the ranking positions of negative samples on the sorted $\vec{score}$ instead of the $\vec{score}$ itself (line 18,19). As a result, a negative example at higher rank will be weighted more than negative examples at

Algorithm 3: A Cascade of Multi-Level Multi-Instance Classifiers

Input : A set $D_w = \{B^+, B^-\}$ of positive and negative examples for word w.
Output: A strong classifier H_w for w

Initialize $score_n = 0$, $\theta_i = 1/|B^-|$, $c = 0$, and $\alpha_m = 0$ for $n = 1, \ldots, |B|$, $i = 1, \ldots, |B^-|$;, and $m = 1, \ldots, M$.

//Learning weak classifiers over T levels
for $l \leftarrow 1$ **to** T **do**
  if $l == 1$ **then**
    Learn classifiers h_m using SVM from D_w for all $m \in \mathcal{M}(l)$
  if $l > 1$ **then**
    Sample a smaller set SB^- from B^- according to θ
    Learn classifiers h_m using MI-SVM from $SD_w = \{B^+, SB^-\}$ for all $m \in \mathcal{M}(l)$
  end
  //Update *score* **for all images in** D_w
  Set $score_n = \gamma_l * score_n + \sum_{m \in \mathcal{M}(l)} \lambda_m * h_m(\mathcal{F}_m(I_n)) + c_l$ for $n = 1, \ldots, |D_w|$
  Find coefficients $\gamma_l > 0$, λ_m and c_l to minimize $||score - Y||_2$
  //Update coefficients of classifiers in previous levels
  for $m' = 1$ **to** $\min\{\mathcal{M}(l)\} - 1$ **do**
    $\lambda_{m'} = \lambda_{m'} * \gamma_l$
  end
  Update the overall threshold $c = c * \gamma_l + c_l$
  Sort *score* in descending order, and let r_j be the ranking position of $I_j \in B^-$ in sorted *score*
  Update $\theta_j \leftarrow \theta_j * 1/r_j$ for all $j = 1, \ldots, |B^-|$ and normalize θ so that $\sum_j \theta = 1$
end
Final robust classifier:

$$H_w = \frac{\sum_{m=1}^{M} \lambda_m * h_m + c}{\sum_{m=1}^{M} \lambda_m + c}$$

lower ranks. From experiments, we see that this ranking-based scheme is better than score-based. The unbalance of negative and positive examples in training makes the number of false positives smaller. In the extrema case, if we treat the positive and negative classes the same, one can assign all images to negative class to obtain reasonably high accuracy (false positive = 0, but false negative is still small in ratio). If score-based sampling is used, little images will be selected as next-level negative examples.

6.5.2 Detailed Analysis

This section presents theoretical analysis for our algorithm, which focuses on the benefit in training time and shows that our algorithm is suitable to image annotation.

Based on cascading scheme, it is obvious that CMLMI requires less training time than learning

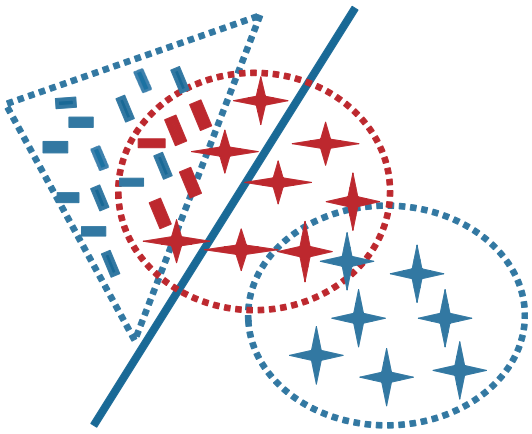
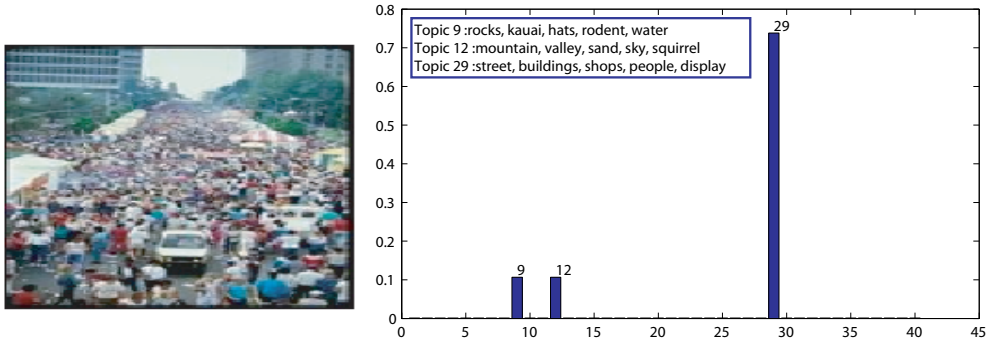


Figure 6.4: Negative bags that share common negative instances with positive bags help reduce ambiguity. Here the stars denote unknown classes of instances (either positive (+) or negative (-)

all individual classifiers independently. The training time of MI-SVM depends on $|B^+|+NR*|B^-|$, where NR is the number of subregions per image. That NR is larger on finer levels makes the domination of negative instances over positive ones even more serious. Training MI-SVM in cascade with SB_w (Line 7 in Algorithm 3) is more efficient than training an independent one with D_w.

Not only having advantage in training time, but also CMLMI is suitable to image annotation and able to reduce the ambiguity of weakly labeling. When the coarse levels are in charge of detecting related context of the given level, the finer levels are able to focus on sample images of similar scene to separate the object from the background, and reduce ambiguity caused by weakly labeling. Figure 6.4 demonstrates our idea. Here, circles still denote positive bags, in which we know positive instances are available but do not know which ones, and triangles denote negative bags, of which we have guarantee that all instances are negative. The negative bag selected here is the one with instances close to some other instances of one positive bag (the red circle). The common/similar instances correspond to subregions of the shared/similar background of the two bags. Since we have the knowledge that all instances of the negative bag are negative, we can conclude that the instances of the red circle, which are close to or even included in the negative bag, are negative. Along with the similarity among positive bags, which contain the same object, this information helps obtain better hyperplane to separate negative and positive instances.

To our best knowledge, this is one of the first attempts that makes use of the similarity between negative bags and positive bags to reduce ambiguity in MIL. Most of previous approaches in MIL only made use of similarity among positive bags to deal with the ambiguity. For example, Carneiro et al. [17] only uses positive bags to generalize a dominating distribution over positive bags. Maron et al. [66] finds regions in the instance space with instances from many different positive bags and far away from instances from negative bags. In [111, 2], negative bags are sampled randomly only to cope with the domination of negative examples over positive examples without giving notice to negative bags that share backgrounds with positive bags. Recently, Deselaers et al. [23] also follows



(a) Candidates: street, buildings, people, valley, rocks

(b) Dominant topics inferred from candidates

Figure 6.5: Refinement with Hidden Topics

the idea that positive instances are all similar whereas every negative instance is negative in its own way. This assumes that the significant portion of positive instances will result in a reasonable classifier performing better than by change. We, however, observe that some negative instances also account for significant portion, which are the instances corresponding to common backgrounds. This issue becomes more serious when more and more labels are taken into consideration as in image annotation.

6.6 Annotation Refinement with Hidden Topics

As mentioned in Chapter 5, scene plays an important role to ensure reasonable annotation. This section focuses on refining annotation, which is produced by CMLMI, using Latent Dirichlet Allocation (LDA). There are two steps to refine annotation with hidden topics:

- Scene analysis: Similar to word-topic distribution in Chapter 5, we obtain only the label part of the training dataset D to perform topic estimation. Here, topics are considered as the combinations of words to form scenes. The process is challenging compared to hidden topic modeling from normal documents because one image is only associated with a couple of labels. In the next subsection, we will give in-detail analysis about these difficulties and show our heuristic solutions.

- Annotation refinement based on topics: The scene analysis step provides us a topic model. Given a set $\tilde{D}$ of new images, we used CMLMI to generate a matrix of confidence scores $\tilde{W}$ of size $\tilde{N} \times |V|$, where $\tilde{N}$ is the number of images in $\tilde{D}$ and $|V|$ is the size of the vocabulary V. Based on $\tilde{W}$ and the topic models, we can refine annotation results to obtain more topic-oriented annotation. One example is given in Figure 6.5 where candidates are words with confidence scores greater than 0. It is obvious that the set of candidates {street, buildings, people, building, valley, rocks} is not a reasonable annotation since "valley" and "rocks" do not often come with "street" view. The topic analysis from the set of candidates show that the

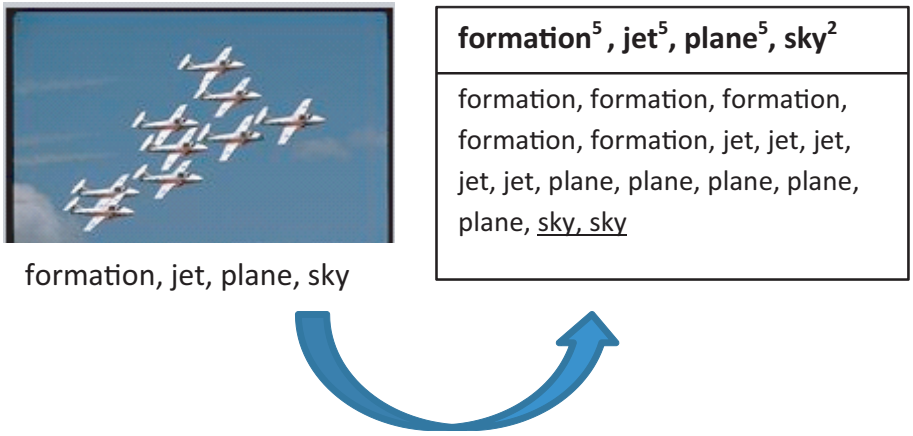


Figure 6.6: Representing label part with duplicated words, where stop words are repeated less frequently than normal words. The first row on right side of the figure is the representation in the short form, which is expressed in long form on the second row. Here, the number next to each word shows how frequent we repeat that word. For example, we repeat 5 times for each word "formation", "jet" and "plane", but only 2 times for the stop word "sky".

street view topic (topic 29) dominate other topics including those of "valley", "rocks", which were mistakenly assigned by CMLMI due to the similarity in visual features. Eliminating those insignificant topics helps remove noisy words and obtain topic-oriented annotation.

6.6.1 Scene Analysis with Latent Dirichlet Allocation

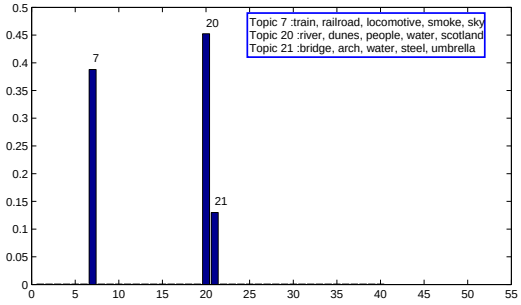
We perform scene analysis using Latent Dirichlet Allocation (see Chapter 2). In sum, given a collection of documents and a predefined topic number K, LDA consists of several processes. Firstly, a document $\vec{w}_m = \{w_{m,n}\}_{n=1}^{N_m}$ is generated by picking a distribution over topics $\vec{\vartheta}_m$ from a Dirichlet distribution ($Dir(\overrightarrow{\alpha})$), which determines topic assignment for words in that document. Then the topic assignment for each word placeholder $[m, n]$ is performed by sampling a particular topic $z_{m,n}$ from multinomial distribution $Mult(\vec{\vartheta}_m)$. And finally, a particular word $w_{m,n}$ is generated for the word placeholder $[m, n]$ by sampling from multinomial distribution $Mult(\overrightarrow{\varphi}_{z_{m,n}})$.

In comparison with normal documents, image captions are sparser, i.e. they contains from five to dozen of words. The sparsity reduces word co-occurrence, which is the basic for topic analysis. The sparsity also change the way we normally treat common words, which are those appear in many scenes and have little support for topic analysis such as "sky", "water". In topic analysis for normal documents (Chapters 3, 4), we can simply remove those common words. For image captions, however, removing common words further reduce the sparsity and leads to unexpected results. Our heuristic solution is to increase the co-occurrence statistics by repeating words in image captions. For common words, we can make them less important by repeating them less than normal words. Figure 6.6 demonstrates our idea, in which *sky* is repeated only 2 times while the other words are repeated 5 times.

Figure 6.7: The relations between α and the sparsity of topic distribution in LDA

The sparsity of image captions also relate to the sparsity of topic distribution, i.e. one image has smaller number of dominant topics. Fortunately, the hyper-parameter α of LDA allows us to control the sparsity. Even that careful investigation should be done to deal with the sparsity in image captions, it beyonds the scope of this study. We, however, performed an experimental study on selecting α for topic analysis with image captions on Corel5K, where each image consists of 3-5 labels. Fixing the number of topics K to 50, several values of hyper-parameter α of LDA were selected. We showed the corresponding topic distributions of sampled documents in Figure 6.7. It can be observable that the smaller α is, the sparser topic distributions become. From experiments, we see that the suitable number of dominant topics for one image is about 2 or 3 topics per image. This number corresponds to α around 0.01 for $K = 50$. For a general number K of topics, we chose $\alpha = 0.5/K$, $\beta = 0.001$, which produces sparser topic model in comparison with normal documents.

(a) Candidates: water, train, tree, people, sky, railroad, bridge



(b) Dominant topics inferred from the candidates

Figure 6.8:

6.6.2 Refinement with LDA

Given a topic model and a confidence score matrix $\tilde{W}$ of new images in $\tilde{D} = \{\tilde{I}_1, \tilde{I}_2, ..., \tilde{I}_{\tilde{N}}\}$, the objective is to modify $\tilde{W}$ to obtain topic-oriented annotation. We consider a specific image $\tilde{I}_n$ with its corresponding confidence score vector $\vec{s}_n = \tilde{W}(n, .)$ of size $|V|$. Because the elements of $\vec{s}_n$ are real numbers, we would like to convert them into a set of candidate words for topic inference. For that, we adopt a simple method that is based on confidence score to repeat words with higher confident more than words with lower confident. The words with confidence score less than 0 will not be included in the final set of candidates. More precisely, the frequency of j^{th} word in candidate set of image $\tilde{I}_n$ is determined as follows:

$$freq(\tilde{I}_n, w_j) = round(\tilde{W}_{n,j} * 3)$$

We then perform topic inference and obtain topic distribution $\vec{\vartheta}$ for each image n. We consider topics with probabilities $\vartheta_k > 0.05$, those topics are called dominant topics. Depending on candidate words, there are several cases of topic distribution:

- One dominant topic is much larger than the other dominant topics: This case is demonstrated in Figure 6.5. It is when CMLMI generates a set of candidate words that strongly contribute to one topic, words belonging to other topics are noisy words. We can easily eliminate the smaller dominant topics and corresponding noisy words.

- Some dominant topics are nearly equal: This case is demonstrated in Figure 6.8. In this case, we don't know which topic is more important, we simply keep all of them, just eliminate topics that significantly small. If the topics are similar in semantic, i.e. candidate words are topic-oriented annotation but we have two similar topics, what we perform conserve the results. If the topics are conflicting like those in Figure 6.8, we just cannot refine annotation with current information. Further investigation should be performed to resolve these ambiguous

Algorithm 4: Annotation Refinement with LDA

Input : confidence score $\vec{s}_n$ of image $\tilde{I}_n$, a vocabulary V, LDA topic model $\mathcal{L} = \{\phi_{k,j} | k = 1, \dots K; j = 1 \dots |V|\}$
Output: refined confidence score $\vec{s}_n$

```
// Perform topic inference
Construct a set of candidates for Ĩ_n by repeating words w_j from V with freq(Ĩ_n, w_j) > 0 times
Perform topic inference using the set of candidates and L, we obtain topic distribution ϑ⃗_n'
// Refinement with topics
Calculate ϑ⃗⃗_n from ϑ⃗_n
Calculate ERT_n for Ĩ_n
if ERT_n < 0.5 then
    Set max be the maximum values of dominant topics K in ϑ⃗⃗_n
    Set topics k with ϑ̃_{n,k} smaller than max to 0
else if ERT_n < 0.9 then
    Calculate mean m̄ of dominant topics K in ϑ⃗⃗_n
    Set topics k with ϑ⃗_{n,k} smaller than mean m̄ to 0
end
foreach word w_j that is s_{n,j} > 0 do
    δ_j = 0
    foreach dominant topic k do
        // φ(k, j) is the topic-word distribution of model L
        δ_j += ϑ̃_{n,k} * φ(k, j)
    end
    if δ_j == 0 then
        s_{n,j} = 0
end
```

topics, or more information should be added to recognize the correct dominant topic (entropy reduced).

We formalize the two cases by making use of Shannon entropy[1] in information theory. Basically, a set of events with larger entropy is more uncertain. The largest entropy posed on a set of events with uniform distribution. For example, a set of two events with distribution of $\{0.5, 0.5\}$ is more uncertain than other set with distribution of $\{0.8, 0.2\}$. Let us consider the topic distribution $\vec{\vartheta}_n$ of the image $\tilde{I}_n$, we modify $\vec{\vartheta}_n$ to obtain $\vec{\tilde{\vartheta}}_n$ in which we keep only dominant topics:

$$\tilde{\vartheta}_{n,k} = \begin{cases} \vartheta_{n,k} & \text{if } \vartheta_{n,k} > 0.05 \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

[1] http://en.wikipedia.org/wiki/Entropy_information_theory

We normalize $\vec{\vartheta}_n$ so that $\sum_{k=1}^{K} \vec{\vartheta}_{n,k} = 1$. Denote $\mathcal{K}$ is the indexing values of dominant topics in $\vec{\vartheta}_n$, we calculate how far the distribution of dominant topics is from uniform distribution by defining the entropy ratio (ERT) as follows:

$$ERT_n = \frac{-\sum_{j=1}^{|\mathcal{K}|} \tilde{\vartheta}_{n,\mathcal{K}_j} \times \log \tilde{\vartheta}_{n,\mathcal{K}_j}}{-log(1/|\mathcal{K}|)}$$

Algorithm 4 summarizes steps for refining image annotation with LDA. Line 7 corresponds to the "one dominant topic" while line 10 corresponds to the "multiple equally dominant topics" case stated above. For the first case, we just keep the maximum topic. In the second case, we eliminate topics smaller than mean of dominant topics. δ_j is a testing variable for word w_j, which is larger than 0 if w_j belongs to any dominant topics k ($\phi(k, j) > 0$). Words w_j with $\delta_j = 0$ are noisy words and eliminated in line 20-22.

6.7 Experiments

6.7.1 Corel5K Dataset

The Corel5k benchmark is obtained from Corel image database and commonly used for image annotation [27, 17, 37]. It contains 5,000 images from 50 Corel Stock Photo CDs and were divided into a training set of 4,000 images, a validation set of 500 images, and a test set of 500 images. Each image is labeled with 1 to 5 captions from a vocabulary of 374 distinct words. These images are with small sizes either of 128×192 or 192×128.

6.7.2 Evaluation

Given a testing dataset containing images with tagged labels, we can perform image annotation on the testing dataset and use typical metrics to measure the effectiveness of the algorithm. Regarding a label w, the typical measures for retrieval are precision P_w, recall R_w:

$$P_w = \frac{\text{Number of images are correctly annotated with } w}{\text{Number of images are annotated with } w}$$

$$R_w = \frac{\text{Number of images are correctly annotated with } w}{\text{Number of images are manually annotated with } w}$$

We calculate P and R, which are means of P_w and R_w over all labels. To balance the trace-off between P and R, $F_1 = 2 * P * R/(P + R)$ is usually used as another measure for evaluation.

6.7.3 Experimental Settings

For the experiments, we performed a cascade of 4 classifiers with 2 levels. Here, we worked with only 2 levels because the images of Corel5K are all in small size. Moreover, we would like to focus on the basic case to analyze the impact of global features on reducing the weakly labeling problem. At the first level, global features were extracted from the whole image. We exploited Gist [76], and

Level 1	$\mathcal{F}_1$: "gist" of scene	SVM-GIST
-	$\mathcal{F}_2$: color histogram	SVM-color
Level 2	$\mathcal{F}_3$: color histogram	MISVM-color
-	$\mathcal{F}_4$: Gabor texture	MISVM-texture

Table 6.1: Feature extractions & classifiers

(a) In comparison with other standalone MIL methods.

Method	P	R	F1
ASVM-MIL [111]	**31%**	39%	35%
mi-SVM [111]	28%	35%	31%
MISVM-Color	13.10%	**55.39%**	21.19%
MISVM-Texture	7.86%	36.16%	12.91%
CMLMI	30.5%	52.35%	**38.54%**

(b) In comparison with standalone SVM with global features

Method	P	R	F1
SVM-Color	20.86%	39.43%	27.28%
SVM-Gist	26.85%	47.40%	34.28%
CMLMI	30.5%	52.35%	38.54%

Table 6.2: CMLMI vs. various MIL methods

color histogram in RGB color space with 16 channels. For each region in the second level, we also performed color histogram extraction but with 8 channels and texture extraction using Gabor filter as in [63]. Summary of feature extraction methods and their relationship with levels are is given in Table 6.1. The numbers of dimension in corresponding feature spaces of algorithms $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$, and $\mathcal{F}_4$ are 960; 4096; 192; and 512 respectively.

We name classifiers trained on feature spaces of $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$, and $\mathcal{F}_4$ as SVM-Gist, SVM-color, MISVM-color, and MISVM-texture. Conventionally, CMLMI is used to indicate the strong classifier H_w learned according to Algorithm 3, in which classifiers of level 2 (MISVM-color, and MISVM-texture) are dependent of classifiers of level 1 (SVM-Gist, and SVM-color). In the following, we refer to, for example, MISVM-color (or standalone MISVM-color) to indicate an independent classifier trained on D_w, and MISVM-color of CMLMI to imply the MISVM-color learned in the cascade according to Algorithm 3. In other words, MISVM-color of CMLMI is the classifier trained on $SD_w = \{B^+, SB^-\}$ based on the results of level 1 (SVM-Gist and SVM-color of CMLMI).

6.7.4 Experimental Results on 70 most Common labels

Like [111], we selected 70 most common labels from Corel5K dataset for experiments. The reason is that labels with a small number of the positive examples are not efficient to train a classifier with multiple instance learning.

Table 6.1(a) shows that CMLMI outperforms other MIL methods. As observable from the

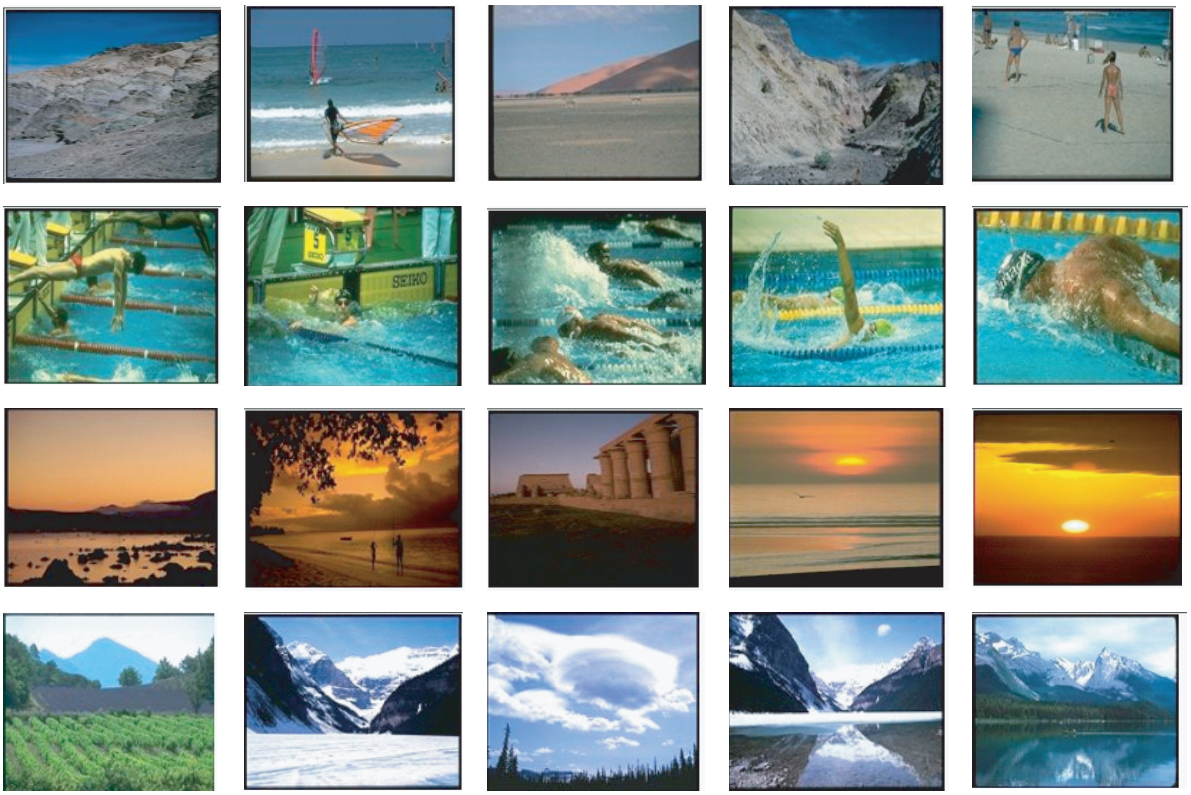


Figure 6.9: From top to bottom: Top 5 retrieved images for queries "sand", "swimmers", "sun" and "mountain"

table, we obtain improvements of 17.35% in F_1 measure compared to MISVM-color. In contrast to MISVM-texture, CMLMI significantly increases F_1 measure by 25.64% (from 12.9% to 38.54%). Comparing to previous works, CMLMI obtains better results than mi-SVM both in precision and recall, which leads to a raise of 7.54% in F_1 measure. Also, CMLMI outperforms ASVM-MIL in recall while obtaining comparable precision (P of 30.5% with CMLMI, and P of 31% with ASVM-MIL). This results in an improvement of our method over ASVM-MIL in F_1 measure (3.54%). In the context that 4% improvement of ASVM-SVM compared to its previous method mi-SVM is considered significant, our method obtained noticeable results.

Table 6.1(b) compares CMLMI to SVM with global features. We can see that CMLMI also obtain better results in F_1 (F_1 of 38.54%) compared with SVM-color (F_1 of 21.28%), and SVM-Gist (F_1 of 34.28%). Among the standalone classifiers (SVM-color, SVM-gist, MISVM-color, and MISVM-texture), SVM with global features outperform MISVM with region-based feature extractions. Interestingly, SVM-Gist is even comparable to ASVM-MIL although image segmentation, which is more expensive than global feature, has been used in ASVM-MIL. However, combining the classifiers in our cascading algorithm yields the best results.

We show examples of retrieval results with four sampled queries ("sand", "swimmers", "sun" and "mountain") in Figure 6.9. As we can see from the figure, the results are various in background, scene. For example, "sand" appears in mountainous regions, in beaches, or in deserts. The photos of swimmers also show different positions of the athletes. Although, we have one error with "sun" and one with "mountain", the results are satisfactory.

6.7.5 Experimental Results on Sample Foreground Labels

We conducted carefully analysis for "tiger", "horse" and "bear" in Corel5K since the concepts correspond to foreground objects which might benefit from finer levels. As previously mentioned, the negative examples for finer levels are drawn based on the ambiguity of coarser levels, which are able to detect the background better. By considering the negative examples of similar background, we are able to add "negative instances", which usually appear with the real positive instances of positive examples. As a result, there is more chance for us to separate the "positive instance" from "negative instance" in positive examples. Table 6.3 shows that the idea works for "tiger", "horse" and "bear" in Corel5K. We observed the selector variable in MI-SVM (Section 6.4) in 2 cases: standalone MISVM and MISVM of CMLMI. We then measure the portion in the collection of positive bags (of "tiger", "horse" and "bear") with correct instances selected by MISVM selector variable. The results of this measurement are shown in Table 6.3. It can be seen that MISVM methods of CMLMI are able to select relevant instances (subregions) from images of "tiger", "horse", "bear" better than standalone MISVM methods. For example, MISVM-texture of CMLMI can select correct positive instances in 72.5% bags in the collection of positve bags compared to 63.73% obtained from standalone MISVM-texture. On average, MISVM of CMLMI outperforms standalone MISVM by 32% (from 20.58% to 27.233%). Although, both standalone MISVM and MISVM of CMLMI do not perform very well on "horses" and "bear", MISVM of CMLMI always achieve better results. The low performance of standalone MISVM on "horses" and "bear" is due to several reasons:

- "Horses" and "bear" are not easy to capture by texture and color. Moreover, that most of "bear" in Corel5K are "polar bear" which hides themselves in the background makes the task of selecting "polar bear"-related regions from background become even harder.
- The influence of background: From the Corel5K dataset, we see that most of "horses" appear on "grass field". Imagine that one image is divided into 5 overlapping subregions at level 2, one image has one region related to horses while 4 other regions related to grass. This makes the number of negative instances dominates the number positive instances in the training dataset for "horses". In other words, the "negative instances" of the collection of positive bags are not "negative in their own way" like the assumption in most of previous MIL methods. Consequently, MISVM as well as previous MIL methods can not select correct positive instances in this case.

Figure 6.10 shows mean average precision (MAP), which is the mean of precisions calculated at every position on the returned ranking list where recall changes, of standalone classifiers and CMLMI for three labels. It can be seen that individual feature types have different influences on different labels. Except for Gist ($\mathcal{F}_1$) that shows its importance for all three labels, global color

	Standalone MISVM		MISVM of CMLMI	
	color	texture	color	texture
Tiger	9.9%	63.73%	21.97%	72.5%
Horses	0%	23.3%	0%	33.98%
Bear	21.71%	7.07%	23.73%	10%
Avg	20.58%		27.233%	

Table 6.3: The percentage of positive images (of tiger, horses, and bear) with their relevant instances selected by Multiple-instance SVM in 2 different strategies: 1. Standalone MI-SVM; 2. MI-SVM in the cascade.

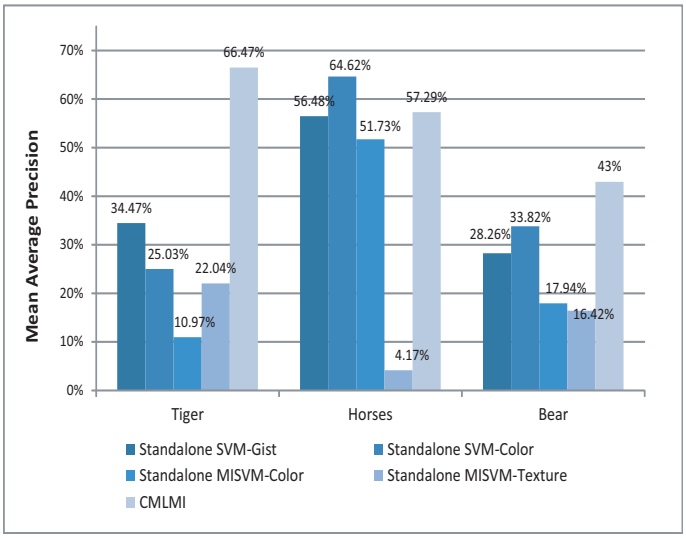
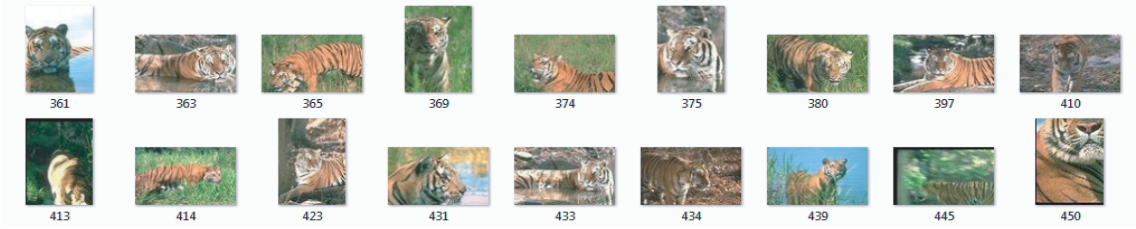


Figure 6.10: mAP of our cascade of multilevel multi-instance SVM (MLMI-SVM) in comparisions with different methods

histogram ($\mathcal{F}_2$) has more impact on annotating images with "horses" and "bear" than with "tiger". Texture feature at level 2 (of MISVM-texture) performs better than the other feature extraction methods only with "tiger". CMLMI significantly outperforms other standalone classifiers on "tiger" and "bear" while falls a little on "horses" compared with SVM-color. Interestingly, standalone MISVM-color is comparable to CMLMI for "horses". In order to uncover the question in the "horse" case, we conducted detailed analysis, and found that MISVM-color and SVM-color captured grass fields in the background instead of horses. Indeed, no subregion with the color of a horse was considered in MISVM-color. Thus, the good performance of standalone MISVM-color and SVM-color owes to special feature of the Corel5K dataset in which horses are on grass fields in most of pictures. This observation agrees with our claim above about the domination of background-related negative instances.

Figure 6.11 and Figure 6.12 show the examples of selecting positive instances from corresponding positive bags with standalone MISVM and MISVM of CMLMI. We can see from the figures that

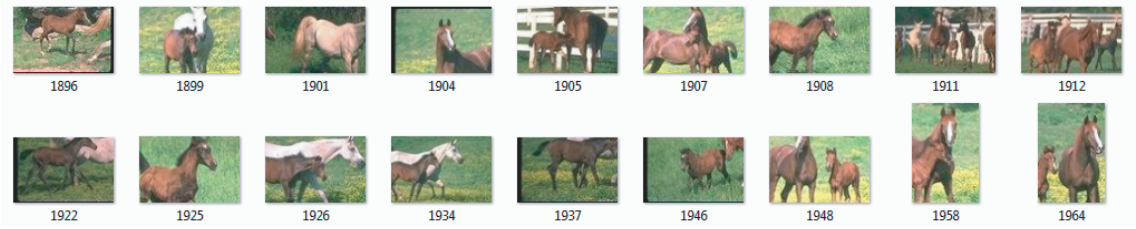
(a) Subregions selected by standalone MISVM-color



(b) Subregions selected by MISVM-color of CMLMI

Figure 6.11: The subregions selected by standalone MISVM-color for label "tiger", and the subregions selected by MISVM-color of CMLMI from the corresponding images. Here, the numbers under each subregion indicate the identifier numbers of corresponding images.

(a) Subregions selected by standalone MISVM-texture



(b) Subregions selected by MISVM-texture of CMLMI

Figure 6.12: The subregions selected by standalone MISVM-texture for label "horses" and the subregions (of corresponding images) selected by MISVM-texture at the 2-nd level of CMLMI

Methods	P	R	F1
CMLMI	30.50%	52.35%	38.54%
CMLMI-HT10	33.20%	47.09%	38.94%
CMLMI-HT20	36.73%	46.32%	40.97%
CMLMI-HT30	34.04%	49.05%	40.19%
CMLMI-HT40	**36.32**%	**47.20** %	**40.99**%
CMLMI-HT50	32.48%	49.20%	39.13%
CMLMI-HT60	34.12%	47.97%	39.87%
CMLMI-HT70	33.30%	48.50%	39.48%
CMLMI-HT80	33.89%	48.92%	40.04%
CMLMI-HT90	33.54%	48.93%	39.80%
CMLMI-HT100	32.05%	48.50%	38.59%

Table 6.4: Annotation Refinement Results with Latent Dirichlet Allocation

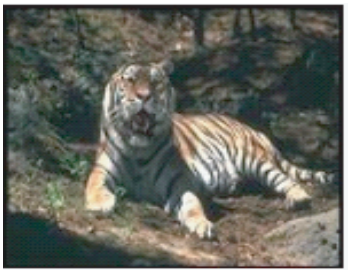
MISVM of CMLMI is able to select more relevant subregions.

6.7.6 Annotation Refinement Results

For annotation refinement, we performed topic estimation on captions of 4500 images of Corel5K with Latent Dirichlet Allocation with different number of topics ($K = 10, ...100$). We then performed annotation refinement with the estimated topic models as described in previous section. Table 6.4 shows precision, recall of using CMLMI with topics vs. CMLMI without refinement. In general, refinement process obtain an increase in precision while making recall reduce. The rate of increasing precision, however, is larger than that of reducing recall, which results in the rise of F_1. The most significant increase in F_1 is about 6% in CMLMI-HT40 (40.99%) compared to CMLMI (38.54%) and about 17.4% compared to ASVM-MIL. Image annotation refinement process not only produces topic-oriented annotation, but also it brings a reasonable threshold to balance precision and recall, which is one problematic issue in multiple label/class learning. Some demonstrative examples are given in Figure 6.13. As can be seen from the figure, the annotation after refinement is more reasonable. For example, in the picture of street view, the mistaken labels ("water", "bridge") caused by the ambiguity of visual content have been removed successfully to obtain reasonable annotations to describe street view.

Unlike mixture hierarchy method that only makes use of positive bags for estimating feature-word distribution in Chapter 3, CMLMI exploits negative bags, thus be more discriminative. It is mentioned in Section 5.7.4 of Chapter 5 that mixture hierarchy leads to a near-uniform feature-word distribution on top 20 candidates of an image. In order to obtain high retrieval performance with mixture hierarchy, we need to fix 5 top labels for indexing. When one candidate from mixture hierarchy is removed from top 5, it makes space for other candidates of the same scene to fill in. CMLMI, on the other hand, gives more weights on a small number of labels. As a result, we do not need to cutoff labels for indexing or annotation refinement. The side effect of this benefit is that we do not have the same "fill-in" effect when refining CMLMI with topics. The examples in

Human: mountain, sky, sun, water
CMIML: water, sky, mountain, sun, bear, city
CMIML-HT: water, sky, mountain, sun



Human: Bengal, cat, forest, tiger
CMIML: tiger, cat, tree, forest, rocks, fox
CMIML-HT: tiger, cat, tree, forest, rocks

Human: buildings, hotel, skylines, street
CMIML: street, buildings, sky, water, people, window, bridge
CMIML-HT: street, buildings, sky, people

Human: coral, ocean, reefs
CMIML: reefs, ocean, coral, flowers, sky, valley, sand, coast
CMIML-HT: reefs, ocean, coral

Human: flowers, garden, monks, people
CMIML: flowers, plants, garden, cat, tiger
CMIML-HT: flowers, garden

Human: athlete, pool, swimmers, water
CMIML: water, swimmers, pool, people, boat
CMIML-HT: water, swimmers, pool, people

Figure 6.13: Examples of annotations before and after topic-based refinement

Figure 6.13 demonstrate this discussion. Fortunately, a small procedure can be added to annotation refinement to transfer "likely words" of the same scene even they do not appear in candidate words obtained from base image annotation.

6.8 Concluding Remarks

In this chapter, we proposed a method based on cascading multi-level multi-instance classifiers, which has main advantages as follows:

- Our cascade of MLMI classifiers is able to reduce training time since we can remove some negative examples, those are "easily" detected as negative based on the scene.
- Multi-level feature extraction allows us to tune suitable features for different types of labels (background/foreground). Also, it allows annotating images with multiple resolutions. One

example is that a photo of tiger might be a close-up photo or the photo of a tiger in its context. Multi-level feature extractions bring more chance to capture all of this variety.

- We also show experimentally that it is able to reduce the ambiguity of "weakly labeling" in image annotation, and separate the foreground objects from the scene in finer levels of the cascade. Indeed, the idea that we have exploited is that *"positive instances co-occur with its context-related negative instances"*.

Also in this chapter, we applied Latent Dirichlet Allocation for annotation refinement. Moreover, we discussed the influence of the sparseness of captions and topics of images on modeling the scene in topics. Several heuristic solutions have been showed to deal with these problems. We proposed an algorithm to refine annotation with topics based on measuring how topic distribution, which is inferred for label set obtained by CMLMI, is far from a uniform distribution. The idea is the it is more certain to remove insignificant topics if the difference of the dominant topics and the removed topics is larger.

The experiments show significant results of CMLMI and CMLMI with topics in comparison with several baselines on Corel5K - the common benchmark in image annotation. The most significant improvement is obtained with CMLMI-HT40 with F1-measure of 40.99%, that gains 17.14% compared to previous work ASVM-MIL[111] (F1-measure of 35%).

Chapter 7

Conclusions

The overall goal of this thesis was to bridge semantic gaps in information retrieval. Two types of semantic gaps have been considered that are the gap between textual captions and human concepts, and the gap between image features and textual captions. The first gap is related to multiple linguistic phenomenon such as synonymy, polysemy. It causes data mismatching, reduces retrieval performance in text-based searching. The later is between visual features of images and descriptive labels, which prevents object recognition extending to a larger number of objects. Together, the gaps make the problem of organizing, searching images much more challenging than the textual counterpart. Toward advanced context-based image search, we focused on closing the gaps in the following aspects:

- A general framework for clustering and ranking short documents, which is able to close the gap between textual content and human concepts. The main part is that we collect a large universal dataset for topic estimation. The topic model obtained from the universal dataset can be used as an easy-to-obtain knowledge base to enrich short texts with richer semantics. Our solution is simple, easy to implement, adaptable to multiple languages, and shown the effectiveness in multiple applications, which are search clustering and contextual advertising.

- A feature-word-topic model was proposed in order to obtain topic-oriented image annotation for image retrieval. The model consists of two parts: 1) feature-word distribution, and 2) word-topic distribution. When annotating new images, the feature-word distribution is used to obtain weighted candidate words for topic inference, which makes use of word-topic distribution. By using topics in annotation, we are able to obtain more reasonable annotation which leads to a better retrieval performance. The separation between feature-word part and word-topic part makes it easier to adapt to different topic models, different methods for estimating feature-word distribution. Moreover, since we do not consider word-to-word relationships for annotation refinement like previous works, it is easier for feature-word-topic model to extend the vocabulary of image annotation.

- A cascade of multi-level and multi-instance classifiers (CMLMI) has been proposed to address three open questions in image annotation: 1) how to compromise between enriching feature

extraction and the large number of labels; 2) how the context can be used to reduce ambiguity in image annotation; and 3) what is the reasonable way to reduce the domination of negative examples and positive examples caused by the large number of labels. For the first question, we exploited multi-level feature extraction on the attempt to have more suitable features for different types of labels (foreground and background labels). For the second and third question, we made use of cascade learning method in which we sample negative examples in similar context. Indeed, the idea exploited in CMLMI is that *"positive instances co-occur with their context-related negative instances"*. The context help recognize negative instances in positive bag, thus reduce ambiguity of image annotation.

This research has posed several questions in need of further investigation. In the future work, we would like to focus on several directions as follows:

- Further research regarding topic modeling of scenes would be great help in overcoming semantic gaps between visual features and labels, as well as between labels and human concepts. A number of future studies are needed to target the open questions: 1) how to deal with the sparseness of image captions, which contains from ten to dozen labels; 2) how to efficiently incorporate image tags, filename, surrounding texts provided by users into topic models. As suggested in Chapter 6, we are able to tune hyper-parameter α of LDA to control the sparseness of topic distribution on image captions. Also, it will be interesting to access other resources such as Wordnet to reduce the sparseness for modeling scene. In order to address the latter question, topic models based on multiple modalities such as texts, tags, meta-data [86, 68] are likely to be followed. Moreover, it is also interesting to see how topics can co-occur to form a picture. For example, a kitchen scene and a street scene can co-occur in a picture of an open restaurant on the street. However, a scene of desert and beach can not co-occur even they may share a significant part of "sand" region. Toward this direction, CTM-like model can be developed to model scenes, which takes into account characteristics of image captions.

- Our claimed issue in Chapter 6, i.e. "positive instances co-occur with negative instances of the scene", is an intriguing one which could be usefully explored in further research. More research can be performed to encode context as topics. As a result, negative images of the same topic can be considered as hints to exclude negative instances, which often appear with positive instances in positive bags.

- Considerably more work will need to be done to obtain a general solution for image search. The point is the human language is very dynamic with abbreviation, synonymy, polysemy, named entities, and so on. For image retrieval, we can not consider all the words of human language for image annotation. Our general solution is to maintain a set of "hint" foreground objects for annotation. The most confident words obtained from image annotation, the surrounding text provided by users, and global visual features can provide evidences to infer "topics" of the picture. Once the topics of the image has been discovered, we can refine topic annotation, refine surrounding texts as well as transfer more appropriate labels of the same topic. For example, if we know the scene is "OFFICE", even "table lamps" does not occur in

the surrounding texts, or not included in the annotation vocabulary, we still can assign these labels to images with high chance that they appear in the image.

List of publications

Refereed Journal Papers

1. Cam-Tu Nguyen, Xuan-Hieu Phan, Thu-Trang Nguyen, Quang-Thuy Ha and Susumu Horiguchi. Web Search Clustering and Labeling with Hidden Topics. In *ACM Transactions on Asian Language Information Processing (TALIP), ACM, 2009, 8, 1-40*

2. Xuan-Hieu Phan, Cam-Tu Nguyen, Dieu-Thu Le, Le-Minh Nguyen, Quang-Thuy Ha, and Susumu Horiguchi, A Hidden Topic-Based Framework toward Building Applications with Short Web Documents. In *IEEE Transactions on Knowledge and Data Engineering (TKDE), 2011, 23, 961-976*

Refereed Conference Papers

1. Xuan-Hieu Phan, Le-Minh Nguyen, Cam-Tu Nguyen and Susumu Horiguchi, Semantic Analysis of Entity Contexts Toward Open Named Entity Classification on The Web. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING), 2007, 137-144*

2. Cam-Tu Nguyen, Trung-Kien Nguyen, Xuan-Hieu Phan, Le-Minh Nguyen, and Quang-Thuy Ha. Vietnamese Word Segmentation with CRFs and SVMs: an Investigation. In *Proceedings of the 20th Pacific Asia Conference on Language, Information, and Computation (PACLIC), 2007, 215-222*

3. Dinh-Quang Thang, Hong-Phong Le, Thi Minh Huyen Nguyen, Cam Tu Nguyen, Mathias Rossignol, and Xuan Luong Vu. Word Segmentation of Vietnamese Texts: a Comparison of Approaches. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'08), 2008, 1933-1936.*

4. Dieu-Thu Le, Cam-Tu Nguyen, Quang-Thuy Ha, Xuan-Hieu Phan, and Susumu Horiguchi. Matching and Ranking with Hidden Topics toward Online Contextual Advertising. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, 2008, 888-891*

5. Cam-Tu Nguyen, Natsuda Kaothanthong, Xuan-Hieu Phan, and Takeshi Tokuyama. A Feature-Word-Topic Model for Image Annotation. In *Proceedings of 19th ACM International Conference on Information and Knowledge Management (CIKM'10), 2010, 1481-1486*

6. Cam-Tu Nguyen Bridging Semantic Gaps in Information Retrieval: Context-based Approaches. In *Proceedings of VLDB 2010 Phd Workshop, 2010*

Submitted Papers

1. Cam-Tu Nguyen, Vu Ha Le, and Takeshi Tokuyama. Cascade of Multi-level Multi-instance Classifiers for Image Annotation. (submitted to KDIR, May, 2011)

2. Cam Tu Nguyen, Natsuda Kaothanthong, Xuan-Hieu Phan, and Takeshi Tokuyama. A Feature-Word-Topic Model for Image Annotation and Retrieval (submitted to ACM Transaction on Web (ACM TWEB), 2011).

Bibliography

[1] E. Akbas and F.T. Yarman Vural. Automatic image annotation by ensemble of visual descriptors. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, jun. 2007.

[2] Stuart Andrews, Thomas Hofmann, and Ioannis Tsochantaridis. Multiple instance learning with generalized support vector machines. In *Eighteenth national conference on Artificial intelligence*, pages 943–944, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[3] C. Andrieu, N. Freitas, A. Doucet, and M. Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50:5–43, 2003.

[4] Baamboo. Vietnamese search engine. http://mp3.baamboo.coms, 2008.

[5] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics*, pages 79–85, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[6] Satanjeev Banerjee and Ted Pedersen. The design, implementation and use of the ngram statistics. Technical report, 2003.

[7] Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. Clustering short texts using wikipedia. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788, New York, NY, USA, 2007. ACM.

[8] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando De Freitas, David M. Blei, Jaz K, Thomas Hofmann, Tomaso Poggio, and John Shawe-taylor. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[9] D. Blei and J. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[10] D. Blei and J. Lafferty. A correlated topic model of *science*. *The Annals of Applied Statistics*, 1:17–35, 2007.

[11] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *J. Machine Learning Research*, 3:993–1022, 2003.

[12] David M. Blei and Michael I. Jordan. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134, 2003.

[13] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proceedings of WWW*, pages 757–766, 2007.

[14] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *Proceedings of ACM SIGIR*, 2007.

[15] Razvan C. Bunescu and Raymond J. Mooney. Multiple instance learning for sparse positive bags. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 105–112, New York, NY, USA, 2007. ACM.

[16] L. Cai and T. Hofmann. Text categorization by boosting automatically extracted concepts. In *Proceedings of ACM SIGIR*, 2003.

[17] Gustavo Carneiro, Antoni B. Chan, Pedro J. Moreno, and Nuno Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):394–410, 2007.

[18] Patrali Chatterjee, Donna L. Hoffman, and Thomas P. Novak. Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science*, 22:520–541, October 2003.

[19] Hao Chen and Susan Dumais. Bringing order to the web: Automatically categorizing search results. In *Proceedings of CHI'01, Human Factors in Computing Systems*, pages 145–152, 2001.

[20] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tokey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329, 1992.

[21] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.

[22] S. Deerwester, G. Furnas, and T. Landauer. Indexing by latent semantic analysis. *Journal of the American Society for Info. Science*, 41:391–407, 1990.

[23] Thomas Deselaers and Vittorio Ferrari. A conditional random field for multiple-instance learning. In *ICML' 10: The 27th International Conference on Machine Learning*, pages 287–294, 2010.

[24] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11:77–107, April 2008.

[25] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA, 2009. ACM.

[26] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA, 2009. ACM.

[27] P. Duygulu, Kobus Barnard, J. F. G. de Freitas, and David A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 97–112, London, UK, 2002. Springer-Verlag.

[28] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of the 2004 CVPR*, 2004.

[29] Yansong Feng and Mirella Lapata. Automatic image annotation using auxiliary text information. In *Proceedings of ACL-08: HLT*, pages 272–280, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[30] Paolo Ferragina and Antonio Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 801–810, New York, NY, USA, 2005. ACM.

[31] Evgeniy Garilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI*, 2007.

[32] Filippo Geraci, Marco Pellegrini, Marco Maggini, and Fabrizio Sebastiani. Cluster generation and cluster labelling for web snippets: A fast and accurate hierarchical solution. pages 25–36. 2006.

[33] David Grangier and Samy Bengio. A discriminative kernel-based approach to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1371–1384, 2008.

[34] Thomas L. Griffiths, Joshua B. Tenenbaum, and Mark Steyvers. Topics in semantic representation. *Psychological Review*, 114:2007, 2007.

[35] Tom Griffiths and Mark Steyvers. Finding scientific topics. *The National Academy of Sciences*, 101:5228–5235, 2004.

[36] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[37] Jonathon S. Hare, Sina Samangooei, Paul H. Lewis, and Mark S. Nixon. Semantic spaces revisited: investigating the performance of auto-annotation and semantic retrieval using semantic spaces. In *Proc. of the Int. Conf. on Content-based Image and Video Retrieval (CIVR)*, pages 359–368, 2008.

[38] G. Heinrich. Parameter estimation for text analysis. Technical report, University of Leipzig & vsonix, 2005.

[39] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA, 1999. ACM.

[40] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[41] Eva Hörster, Rainer Lienhart, and Malcolm Slaney. Image retrieval on large-scale image databases. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 17–24, New York, NY, USA, 2007. ACM.

[42] Eva Hörster, Rainer Lienhart, and Malcolm Slaney. Continuous visual vocabulary modelsfor plsa-based scene recognition. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 319–328, New York, NY, USA, 2008. ACM.

[43] Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, and Qiang Yangand Zheng Cheng. Enhancing text clustering by leveraging wikipedia semantics. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 179–186, New York, NY, USA, 2008. ACM.

[44] Xiaohong Hu, Xu Qian, Xinming Ma, and Ziquiang Wang. A novel region-based image annotation using multi-instance learning. In *Second International Workshop on Knowledge Discovery and Data Mining*, 2009.

[45] Interactive Advertising Bureau (IAB). Iab internet advertising revenue report. Technical report, 2010.

[46] Bernard J. Jansen, Amanda Spink, Judy Bateman, and Tefko Saracevic. Real life information retrieval: a study of user queries on the web. *SIGIR Forum*, 32(1):5–17, 1998.

[47] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Improving bag-of-features for large scale image search. *Int. J. Comput. Vision*, 87(3):316–336, 2010.

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

[49] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation of news images with large vocabularies and low quality training data. In *Proceedings of ACM Multimedia*, 2004.

[50] Rong Jin, Joyce Y. Chai, and Luo Si. Effective automatic image annotation via a coherent language model and active learning. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 892–899, New York, NY, USA, 2004. ACM.

[51] Yohan Jin, Latifur Khan, Lei Wang, and Mamoun Awad. Image annotations by combining multiple evidence & wordnet. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 706–715, New York, NY, USA, 2005. ACM.

[52] Yushi Jing and Shumeet Baluja. Pagerank for product image search. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 307–316, New York, NY, USA, 2008. ACM.

[53] Yushi Jing and Shumeet Baluja. Visualrank: Applying pagerank to large-scale image search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1877 –1890, nov. 2008.

[54] Lyndon S. Kennedy and Shih-Fu Chang. A reranking approach for context-based concept fusion in video indexing and retrieval. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 333–340, New York, NY, USA, 2007. ACM.

[55] S.B. Kotsiantis and P. E. Pintelas. Recent advances in clustering: A brief survey. *WSEAS Transactions on Information Science and Applications*, 1(1):73–81, 2004.

[56] Anísio Lacerda, Marco Cristo, Marcos André Gonçalves, Weiguo Fan, Nivio Ziviani, and Berthier Ribeiro-Neto. Learning to advertise. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 549–556, New York, NY, USA, 2006. ACM.

[57] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Advances in Neural Information Processing Systems (NIPS?f03)*. MIT Press, 2003.

[58] Svetlana Lazebnix, Cordelia Schmid, and Jean Ponce. *Object Categorization: Computer & Human Vision Perspectives*, chapter Spatial Pyramid Matching. Cambridge University Press, 2009.

[59] Todd A. Letsche and Michael W. Berry. Large-scale information retrieval with latent semantic indexing. *Inf. Sci.*, 100(1-4):105–137, 1997.

[60] Rainer Lienhart, Stefan Romberg, and Eva Hörster. Multilayer plsa for multimodal image retrieval. In *CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA, 2009. ACM.

[61] Jing Liu, Bin Wang, Hanqing Lu, and Songde Ma. A graph-based image annotation framework. *Pattern Recogn. Lett.*, 29(4):407–415, 2008.

[62] Ameesh Makadia, Vladimir Pavlovic, and Sanjiv Kumar. A new baseline for image annotation. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 316–329. Springer-Verlag, 2008.

[63] Ameesh Makadia, Vladimir Pavlovic, and Sanjiv Kumar. Baselines for image annotation. *Int. J. Comput. Vision*, 90(1):88–105, 2010.

[64] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *An Introduction to Information Retrieval*. Cambridge University Press, 2009.

[65] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistic Natural Language Processing*. MIT Press, 1999.

[66] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, pages 570–576, Cambridge, MA, USA, 1998. MIT Press.

[67] Qiaozhu Mei, Xuehua Shen, and Chengxiang Zhai. Automatic labeling of multinomial topic models. In *Proceeding of KDD'07*, San Jose, California, USA, august 2007.

[68] David M. Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, pages 411–418, 2008.

[69] Florent Monay and Daniel Gatica-Perez. Plsa-based image auto-annotation: constraining the latent space. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 348–351, New York, NY, USA, 2004. ACM.

[70] Florent Monay and Daniel Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(10):1802–1817, 2007.

[71] Florent Monay and Daniel GaticaPerez. On image autoannotation with latent space models. In *Proceedings of 2003 MM conference*, 2003.

[72] Milind R. Naphade, Igor Kozintsev, Thomas S. Huang, and Kannan Ramchandran. A factor graph framework for semantic indexing and retrieval in video. In *CBAIVL '00: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, page 35, Washington, DC, USA, 2000. IEEE Computer Society.

[73] Chi-Lang Ngo. A tolerance rough set approach to clustering web search results. Master's thesis, Warsaw University, December 2003.

[74] Cam-Tu Nguyen, Trung-Kien Nguyen, Xuan Hieu Phan, Le Minh Nguyen, and Quang Thuy Ha. Vietnamese word segmentation with crfs and svms: An investigation. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Compuation (PACLIC20)*, pages 215–222, Wuhan, China, november 2006.

[75] Cam-Tu Nguyen, Xuan-Hieu Phan, Susumu Horiguchi, Thu-Trang Nguyen, and Quang-Thuy Ha. Web search clustering and labeling with hidden topics. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(3):1–40, 2009.

[76] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[77] Stanislaw Osinski. An algorithm for clustering web search result. Master's thesis, Poznan University of Technology, Poland, June 2003.

[78] Xuan-Hieu Phan, Cam-Tu Nguyen, Dieu-Thu Le, Le-Minh Nguyen, S Horiguchi, and Q Ha. A hidden topic-based framework towards building applications with short web documents. *Knowledge and Data Engineering, IEEE Transactions on*, PP(99):1 –1, 2010.

[79] Xuan Hieu Phan, Le Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of WWW*, 2008.

[80] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 91–100, New York, NY, USA, 2008. ACM.

[81] Xuan-Hieu Phan, Le-Minh Nguyen, **Cam-Tu Nguyen**, and Susumu Horiguchi. Semantic analysis of entity contexts toward open named entity classification on the web. In *Proceedings of The Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 137–144, Melbourne, Australia, 9 2007.

[82] Adrian Popescu, Christophe Millet, and Pierre-Alain Moëllic. Ontology driven content based image retrieval. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 387–394, New York, NY, USA, 2007. ACM.

[83] A. Popescul and L. Ungar. Automatic labeling of document clusters, 2000.

[84] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. Correlative multi-label video annotation. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 17–26, New York, NY, USA, 2007. ACM.

[85] Till Quack, Bastian Leibe, and Luc Van Gool. World-scale mining of objects and events from community photo collections. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 47–56, New York, NY, USA, 2008. ACM.

[86] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[87] Vikas C. Raykar, Balaji Krishnapuram, and Shipeng Yu. Designing efficient cascaded classifiers: tradeoff between accuracy and cost. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 853–860, 2010.

[88] Berthier Ribeiro-Neto, Marco Cristo, Paulo B. Golgher, and Edleno Silva de Moura. Impedance coupling in content-targeted advertising. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 496–503, New York, NY, USA, 2005. ACM.

[89] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of WWW*, 2006.

[90] Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors. *Advances in kernel methods: support vector learning*. MIT Press, Cambridge, MA, USA, 1999.

[91] Peter Schonhofen. Identifying document topics using the wikipedia category network. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 456–462, Washington, DC, USA, 2006. IEEE Computer Society.

[92] Socbay. Vietnamese search engine. www.socbay.com, 2008.

[93] Mark Steyvers and Tom Griffiths. *Latent Semantic Analysis: A Road to Meaning*, chapter Probabilistic topic models. Laurence Erlbaum, 2007.

[94] Martin Szummer and Rosalind W. Picard. Indoor-outdoor image classification. In *CAIVD '98: Proceedings of the 1998 International Workshop on Content-Based Access of Image and Video Databases (CAIVD '98)*, page 42, Washington, DC, USA, 1998. IEEE Computer Society.

[95] Japan The BioCaster Project Members at NII. Global health awareness. http://born.nii.ac.jp/, 2011.

[96] A. Torralba, K. P. Murphy, and W. T. Freeman. Using the forest to see the trees: exploiting context for visual object detection and localization. *Commun. ACM*, 53(3):107–114, 2010.

[97] Pucktada Treeratpituk and Jamie Callan. Automatically labeling hierarchical clusters. In *Proceedings of the 2006 International Conference on Digital government research*, San Diego, California, USA, 2006.

[98] Theodora Tsikrika, Christos Diou, Arjen P. de Vries, and Anastasios Delopoulos. Image annotation using clickthrough data. In *CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA, 2009. ACM.

[99] A. Vailaya, A. Jain, and H. J. Zhang. On image classification: City vs. landscape. In *CBAIVL '98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries*, page 3, Washington, DC, USA, 1998. IEEE Computer Society.

[100] Nuno Vasconselos. Image indexing with mixture hierarchies. In *IEEE Conference in Computer Vision and Pattern Recognition*, 2001.

[101] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511 – I–518 vol.1, 2001.

[102] Vivisimo. Clustering engine. http://vivisimo.com/, 2008.

[103] Vnnic. Vietnam internet center. http://www.thongkeinternet.vn, 2008.

[104] Changhu Wang, Feng Jing, Lei Zhang, and Hong-Jiang Zhang. Image annotation refinement using random walk with restarts. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 647–650, New York, NY, USA, 2006. ACM.

[105] Chingning Wang, Ping Zhang, Risook Choi, and Michael D'Eredita. Understanding consumers attitude toward advertising. In *Eighth Americas Conference on Information Systems. (2002) 1143–1148*, pages 1143–1148, 2002.

[106] Chong Wang, D. Blei, and Fei-Fei Li. Simultaneous image classification and annotation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1903–1910, 2009.

[107] Mei Wang, Xiangdong Zhou, and Tat-Seng Chua. Automatic image annotation via local multi-label classification. In *ACM International Conference on Image and Video Retrieval*, 2008.

[108] Yong Wang and Shaogang Gong. Refining image annotation using contextual relations between words. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 425–432, New York, NY, USA, 2007. ACM.

[109] Wikipedia. Latent semantic analysis. http://en.wikipedia.org/wiki, 2008.

[110] Xalo. Vietnamse search engine. http://xalo.vn, 2008.

[111] Changbo Yang and Ming Dong. Region-based image annotation using asymmetrical support vector machine-based multi-instance learning. In *In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (June 17 - 22*, 2006.

[112] W. Yih and C. Meek. Improving similarity measures for short segments of text. In *Proceedings of AAAI*, 2007.

[113] Wen-tau Yih, Joshua Goodman, and Vitor R. Carvalho. Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 213–222, New York, NY, USA, 2006. ACM.

[114] Oren Zamir and Oren Etzioni. Grouper: a dynamic clustering interface to Web search results. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1361–1374, 1999.

[115] Hua Jun Zeng, Qi Cai He, Zheng Chen, Wei Ying Ma, and JinWen Ma. Learning to cluster web search results. In *Proceedings of 27th Annual International ACM SIGIR*, Sheffield, South Yorkshire, UK, july 2004.

[116] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, and Zengfu Wang. Joint multi-label multi-instance learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[117] Zhongfei Zhang and Roufei Zhang. *Multimedia Data Mining*. Chapman & Hall/CRC Press, 2009.

[118] Qizhen He Zhiwu Lu, Horace Ip. Context-based multi-label image annotation. In *ACM International Conference on Image and Video Retrieval*, 2009.

[119] Xiangdong Zhou, Mei Wang, Qi Zhang, Junqi Zhang, and Baile Shi. Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 25–32, New York, NY, USA, 2007. ACM.

[120] Zing. Vietnamse website directory. http://directory.zing.vn, 2008.