IDIAP RESEARCH REPORT

Machine Learning for Information Retrieval

David Grangier [1,2]

IDIAP–RR 08-34

2008

PUBLISHED IN
Thèse de l'École Polytechnique Fédérale de Lausanne (EPFL), no. 4088,
Faculté des Sciences et Techniques de l'Ingénieur (STI).

Thesis Committee Members:

Dr. Samy Bengio [3], thesis advisor.
Prof. Hervé Bourlard [1,2], thesis advisor.

Dr. Jean Cédric Chappelier [2], internal expert.
Dr. Yves Grandvalet [4], external expert.
Prof. Thomas Hofmann [5,6], external expert.

1 IDIAP Research Institute, Martigny, Switzerland.
2 École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
3 Google Inc., Mountain View, CA, USA.
4 Université de Technologie de Compiègne, France.
5 Google Inc., Zürich, Switzerland.
6 Brown University, Providence, RI, USA.

IDIAP Research Institute **www.idiap.ch**
Av. des Prés Beudin 20 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

Résumé

Dans cette thèse, nous explorons l'utilisation de techniques d'apprentissage automatique pour la recherche d'information. Plus précisément, nous nous intéressons au problème de l'*ad-hoc retrieval* qui consiste à chercher les documents pertinents à une requête dans de grands corpus. Ce problème est généralement résolu à travers une tâche d'ordonnancement. Étant donnée une requête, le système de recherche d'information ordonne les documents du corpus avec l'objectif de produire une liste ordonnée dans laquelle les documents pertinents apparaissent avant les autres.

Dans un contexte d'apprentissage automatique, nous nous sommes intéressés à proposer des algorithmes d'apprentissage qui puissent bénéficier de données d'entraînement afin d'identifier un ordonnateur capable d'atteindre une performance élevée pour des documents et des requêtes non disponibles lors de l'apprentissage. Ce problème présente des défis nouveaux comparé aux problèmes traditionnels d'apprentissage automatique, tels que la catégorisation ou bien la régression. En effet, notre tâche est un problème d'ordonnancement, ce qui implique que la mesure de l'erreur pour une requête ne peut être exprimée comme la somme des coûts reliés à chaque document. Également, notre tâche correspond à un problème déséquilibré puisque seulement une très petite partie du corpus est pertinente pour chaque requête. De plus, l'*ad-hoc retrieval* présente un double problème de généralisation puisque le modèle doit à la fois être appliqué à de nouveaux documents mais également à de nouvelles requêtes. Finalement, notre tâche présente aussi des contraintes d'efficacité de calcul puisque l'*ad-hoc retrieval* est généralement appliqué à de grands corpus. L'objectif premier de cette thèse est l'apprentissage discriminant de modèles pour l'*ad-hoc retrieval*. Dans ce contexte, nous proposons plusieurs approches basées sur des machines à noyaux ou sur des réseaux de neurones spécifiquement

adaptés à différents problèmes de recherche d'information. Les modèles proposés sont basés sur différents algorithmes d'apprentissage en ligne permettant un apprentissage efficace sur de grands ensembles de données.

La première partie de ce document s'intéresse à la recherche d'information textuelle. Pour ce faire, nous adoptons une formulation classique de cette tâche, et ordonnons les documents selon une estimation de leur similarité avec la requête. La mesure de la similarité entre textes est donc un aspect crucial dans ce cadre. Nous proposons donc une approche pour apprendre une mesure de similarité entre textes. Notre stratégie d'apprentissage ne repose pas sur un large ensemble de requêtes et de documents pertinents correspondants car ce type de données étant particulièrement coûteux à annoter est rare. Au lieu de cela, nous proposons d'utiliser un corpus hypertexte, dont les hyperliens fournissent une information de proximité sémantique entre documents. Ce protocole correspond donc à un transfert d'apprentissage, où nous bénéficions de l'information fournie par les hyperliens pour améliorer la performance sur la tâche de recherche d'information.

Ensuite, nous nous intéressons à un autre problème de recherche d'information : la recherche d'images à partir de requêtes textuelles. Notre approche propose d'optimiser un critère lié à la performance d'ordonnancement. Ce critère est dérivé du critère utilisé dans notre travail précédent concernant l'apprentissage de similarité textuelle. Ce choix conduit à une approche d'apprentissage basée sur la performance finale, mesurée sur la tâche de recherche d'information. Notre approche est donc différente des méthodes usuelles qui s'intéressent généralement au problème final indirectement, à travers une tâche d'annotation d'images. De plus, notre algorithme d'apprentissage s'appuie sur des travaux récents concernant l'apprentissage en ligne de machine à noyaux. Ainsi, nous obtenons un algorithme permettant un entraînement sur de grands ensembles de données et pouvant bénéficier de noyaux récemment introduits pour la comparaison d'images.

Dans la dernière partie de la thèse, nous montrons que le critère utilisé dans les précédents problèmes considérés peut être appliqué à la tâche de repérage de mots-clés (le repérage des mots-clés correspond à la détection de mots-clés dans des séquences de parole). Pour obtenir un formalisme d'ordonnancement, nous proposons un modèle qui doit produire une liste ordonnée d'enregistrements de parole en réponse au mot-clé soumis. Cette liste doit idéalement placer les enregistrements contenant le mot-clé avant les enregistrements ne contenant pas le mot-clé. Il est intéressant de constater que notre formalisme conduit à un critère maximisant directement l'aire sous la courbe ROC (Receiver Opera-

ting Curve), qui est la mesure la plus communément utilisée pour l'évaluation des techniques de repérage de mots-clés. Ce critère est ensuite utilisé pour apprendre un modèle approprié à cette tâche séquentielle. Ce modèle est appris grâce à un algorithme dérivé de celui précédemment introduit pour la tâche de recherche d'image.

En conclusion, cette thèse introduit des techniques d'apprentissage statistique pour la recherche d'information. Nous proposons des modèles d'apprentissage pour plusieurs contextes multimodaux : la recherche de documents textuels à partir de requêtes textuelles, la recherche d'images à partir de requêtes textuelles, ainsi que la recherche des séquences de parole à partir de mots-clés écrits. Nos solutions reposent sur des approches discriminantes et des algorithmes d'apprentissage au coût de calcul réduit. Dans tous les cas, nous faisons un parallèle entre les méthodes proposées et l'état de l'art, puis nous mesurons l'amélioration apportée grâce à des comparaisons expérimentales.

Mots-Clés : apprentissage automatique, recherche d'information, apprendre à ordonner, apprentissage discriminant, apprentissage en ligne, recherche de textes, recherche d'images, repérage de mots-clés dans les enregistrements vocaux.

Abstract

In this thesis, we explore the use of machine learning techniques for information retrieval. More specifically, we focus on ad-hoc retrieval, which is concerned with searching large corpora to identify the documents relevant to user queries. This identification is performed through a ranking task. Given a user query, an ad-hoc retrieval system ranks the corpus documents, so that the documents relevant to the query ideally appear above the others.

In a machine learning framework, we are interested in proposing learning algorithms that can benefit from limited training data in order to identify a ranker likely to achieve high retrieval performance over unseen documents and queries. This problem presents novel challenges compared to traditional learning tasks, such as regression or classification. First, our task is a ranking problem, which means that the loss for a given query cannot be measured as a sum of an individual loss suffered for each corpus document. Second, most retrieval queries present a highly unbalanced setup, with a set of relevant documents accounting only for a very small fraction of the corpus. Third, ad-hoc retrieval corresponds to a kind of "double" generalization problem, since the learned model should not only generalize to new documents but also to new queries. Finally, our task also presents challenging efficiency constraints, since ad-hoc retrieval is typically applied to large corpora. The main objective of this thesis is to investigate the discriminative learning of ad-hoc retrieval models. For that purpose, we propose different models based on kernel machines or neural networks adapted to different retrieval contexts. The proposed approaches rely on different online learning algorithms that allow efficient learning over large corpora.

The first part of the thesis focus on text retrieval. In this case, we adopt a classical approach to the retrieval ranking problem, and order the text docu-

ments according to their estimated similarity to the text query. The assessment of semantic similarity between text items plays a key role in that setup and we propose a learning approach to identify an effective measure of text similarity. This identification is not performed relying on a set of queries with their corresponding relevant document sets, since such data are especially expensive to label and hence rare. Instead, we propose to rely on hyperlink data, since hyperlinks convey semantic proximity information that is relevant to similarity learning. This setup is hence a transfer learning setup, where we benefit from the proximity information encoded by hyperlinks to improve the performance over the ad-hoc retrieval task.

We then investigate another retrieval problem, i.e. the retrieval of images from text queries. Our approach introduces a learning procedure optimizing a criterion related to the ranking performance. This criterion adapts our previous learning objective for learning textual similarity to the image retrieval problem. This yields an image ranking model that addresses the retrieval problem directly. This approach contrasts with previous research that rely on an intermediate image annotation task. Moreover, our learning procedure builds upon recent work on the online learning of kernel-based classifiers. This yields an efficient, scalable algorithm, which can benefit from recent kernels developed for image comparison.

In the last part of the thesis, we show that the objective function used in the previous retrieval problems can be applied to the task of keyword spotting, i.e. the detection of given keywords in speech utterances. For that purpose, we formalize this problem as a ranking task: given a keyword, the keyword spotter should order the utterances so that the utterances containing the keyword appear above the others. Interestingly, this formulation yields an objective directly maximizing the area under the receiver operating curve, the most common keyword spotter evaluation measure. This objective is then used to train a model adapted to this intrinsically sequential problem. This model is then learned with a procedure derived from the algorithm previously introduced for the image retrieval task.

To conclude, this thesis introduces machine learning approaches for ad-hoc retrieval. We propose learning models for various multi-modal retrieval setups, i.e. the retrieval of text documents from text queries, the retrieval of images from text queries and the retrieval of speech recordings from written keywords. Our approaches rely on discriminative learning and enjoy efficient training procedures, which yields effective and scalable models. In all cases, links with prior approaches were investigated and experimental comparisons were conducted.

Keywords: machine learning, information retrieval, learning to rank, discriminative learning, online learning, text retrieval, image retrieval, spoken keyword spotting

Acknowledgments

"Everything should be made as simple as possible, but not simpler."
Albert Einstein

The completion of this thesis was a great adventure, that would not have been possible without many people. I am grateful to them.

First, I would like to thank Samy Bengio, who supervised my research during the last four years. Samy introduced me to the exciting area of Machine Learning research. He has always been present for technical advice, even when commuting between Lausanne and Martigny (some regular commuters should now be proficient in Machine Learning). More importantly, Samy allowed me to conduct my research in a very friendly and playful environment. I am also grateful to Hervé Bourlard, who made the IDIAP Research Institute a place where I could work with a lot of freedom. I further thank my colleagues at IDIAP, who brought me inspiration and essential social interactions.

I am very grateful to my family, that was supportive during this adventure (and before as well). In particular, I thank Elodie for encouraging me in periods of doubts and for calming me down in periods of over-enthusiasm...

Contents

1 Introduction **1**

1.1 Motivations 1

1.2 Organization & Contributions 2

2 State-of-the-Art **5**

2.1 Information Retrieval 5

2.2 Machine Learning for Information Retrieval 11

2.3 Conclusions 20

3 Learning Text Similarity from Hyperlinked Data **23**

3.1 LinkLearn 24

3.2 Related Work 28

3.3 Experiments and Results 31

3.4 Conclusions 37

3.5 Contributions 38

4 Retrieving Images from Text Queries **39**

4.1 Related Work 41

4.2 Passive-Aggressive Model for Image Retrieval 45

4.3 Text and Visual Features 52

4.4 Experiments and Results 55

4.5 Conclusions 68

4.6 Contributions 70

5 Discriminative Keyword Spotting **71**

5.1 Related Work 72

5.2 Discriminative Keyword Spotting 77

5.3 Experiments and Results 85
5.4 Conclusions 90
5.5 Contributions 91

6 Conclusions and Perspectives **93**
6.1 General Summary 93
6.2 Potential Future Work 95

A Appendix **97**
A.1 Regularization in Passive-Aggressive through Early Stopping . 97
A.2 Dynamic Programming Procedure for the Discriminative Spotter 97

1 Introduction

1.1 Motivations

The history of Information Retrieval (IR) parallels the development of libraries. The first civilizations had already come to the conclusions that efficient techniques should be designed to fully benefit from large document archives. As early as 5,000 years ago, the Sumerian librarians were already describing and categorizing official transaction records, legends and theological documents in indexes [Kramer, 1958]. Thematic indexes have then remained the main mean of accessing archived items for centuries. Only recently, IR has radically changed with the advent of computers. Digital technologies provide a unified infrastructure to store, exchange and automatically process large document collections. The search for information consequently evolved from the manual examination of brief document abstracts within predefined categories to algorithms searching through the whole content of each archived document. Nowadays, automatic retrieval systems are widely used in several application domains (e.g. web search, book search or video search) and there is a constant need for improving such systems. In this context, Information Retrieval is an active field of research within Computer Science. This thesis is concerned with one of the main tasks of IR, the so-called *ad-hoc retrieval* task which aims at finding the documents relevant to submitted queries. This problem is generally formalized as a ranking problem: given a query, the retrieval system should rank the documents, so that the items relevant to the query appear above the others. In this context, we focus on machine learning to automatically identify effective ranking function from limited training data.

Machine Learning proposes and studies algorithms that allow computer systems to improve automatically through experience, i.e. from training data. Learning systems are commonly used for several perception tasks, such as au-

tomatic face detection [Viola and Jones, 2001] or automatic speech recognition [Rabiner and Juang, 1993]. The application of Machine Learning to ad-hoc retrieval is attractive, as it is difficult to hand-design effective ranking functions. Moreover, large collections of documents, which provide useful information for learning retrieval models, are readily available. The use of learning techniques for ad-hoc retrieval is however not straightforward, as this task presents several difficulties compared to traditional learning problems, such as regression or classification [Bishop, 2006]. First, our task corresponds to a ranking problem, which implies that the performance for a given query cannot be formalized as a sum of a measure of performance evaluated for each corpus document. Second, most retrieval queries present an highly unbalanced setup, with a set of relevant documents accounting only for a very small fraction of the corpus. Third, ad-hoc retrieval corresponds to a kind of "double" generalization problem, since the learned model should not only encounter new documents but also new queries. Finally, our task also presents challenging efficiency constraints, since ad-hoc retrieval is typically applied to large corpora.

Certainly due to these obstacles, the Machine Learning community has only started devising specific solutions for retrieval ranking in the recent years (see next chapter) and learning techniques are not yet pointed to as potential tools in most reference books on Information Retrieval [Baeza-Yates and Ribeiro-Neto, 1999; Grossman and Frieder, 2004; Witten et al., 1999].

1.2 Organization & Contributions

The remainder of this document is divided into five chapters. Next chapter, Chapter 2, introduces the necessary background on Information Retrieval and the application of Machine Learning to this domain. The following chapters (Chapter 3, Chapter 4 and Chapter 5) are dedicated to our contributions. Each chapter presents an ad-hoc retrieval problem and proposes a learning algorithm to benefit from available training data.

Chapter 3 proposes an approach to learn a measure of similarity between texts from a hyperlinked corpus. Our approach assumes that hyperlinks convey information about document proximity, and it learns a measure assigning a higher similarity to linked documents than to unlinked documents. In a transfer learning setup, we apply the learned measure on ad-hoc retrieval problems, to rank documents according to their similarity with respect to the query. These experiments show that our approach allows the retrieval problem to benefit from the proximity information

encoded by the hyperlinks. Different aspects of this research have been published in [Grangier and Bengio, 2005a] and [Grangier and Bengio, 2005b].

Chapter 4 focuses on ranking images with respect to text queries and introduces a discriminative model for this task. The model parameters are learned so that the relevant images should appear above the others in the final retrieval rankings. This approach contrasts with previous research that mostly relies on an intermediate task, automatic annotation, in order to address this problem. Our experiments over stock photography data show the advantage of focusing directly on the final task. This research has yield several publications [Grangier et al., 2006a,b; Grangier and Bengio, 2006, 2008].

Chapter 5 formalizes the keyword spotting as a retrieval problem. This task, which aims at identifying whether a given keywords is uttered in a speech recording, is generally evaluated through the area under the receiver operating curve. We first show that maximizing this area is equivalent to ranking speech utterance so that the utterances containing the targeted keyword appear above the others. This allows us to apply the learning procedure introduced in the previous chapter to learn a model adapted to this sequential problem. This yields a discriminative keyword spotter that compares favorably with generative alternatives based on Hidden Markov

---

◇ D. Grangier and S. Bengio. Inferring document similarity from hyperlinks. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 359–360, Bremen, Germany, November 2005a.

◇ D. Grangier and S. Bengio. Exploiting hyperlinks to learn a retrieval model. In *NIPS Workshop on Learning to Rank*, pages 12–17, Whistler, Canada, December 2005b.

◇ D. Grangier, F. Monay, and S. Bengio. Learning to retrieve images from text queries with a discriminative model. In *International Workshop on Adaptive Multimedia Retrieval (AMR)*, pages 42–56, Geneva, Switzerland, July 2006a.

◇ D. Grangier, F. Monay, and S. Bengio. A discriminative approach for the retrieval of images from text queries. In *European Conference on Machine Learning (ECML)*, pages 162–173, Berlin, Germany, September 2006b.

◇ D. Grangier and S. Bengio. A neural network to retrieve images from text queries. In *International Conference on Artificial Neural Networks (ICANN)*, volume 2, pages 24–34, Athens, Greece, September 2006.

◇ D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*, 2008. (in press).

Models. This work has resulted in the following publications, [Keshet et al., 2007a; Grangier and Bengio, 2007].

These chapters (Chapter 3, Chapter 4 and Chapter 5) hence correspond to reviewed and published papers, which have been rewritten in a unified framework. Finally, Chapter 6 draws conclusions about this work and outlines potential future directions of research.

◇ J. Keshet, D. Grangier, and S. Bengio. Discriminative keyword spotting. In *International Workshop on Non-LInear Speech Processing (NOLISP)*, pages 47–50, Paris, France, May 2007a.

◇ D. Grangier and S. Bengio. Learning the inter-frame distance for discriminative template-based keyword detection. In *International Conference on Speech Processing (INTERSPEECH)*, pages 902–905, Antwerp, Belgium, August 2007.

2 State-of-the-Art

This chapter introduces some of the foundations of information retrieval and machine learning necessary to understand the rest of the dissertation work. The chapter begins with a discussion on information retrieval techniques and retrieval evaluation. It then moves to machine learning, and reviews how machine learning techniques have been applied to information retrieval problems.

2.1 Information Retrieval

Information Retrieval (IR) is concerned with finding the documents that answer a specific information need within a given corpus. This task is generally addressed through document ranking: the user inputs their information need as a query, and the IR system outputs a ranking, in which the documents relevant to the query should appear above the others.

IR research studies different aspects of this problem, which can mainly be grouped into three categories: ranking functions, data structures and user interfaces. Ranking functions are concerned with estimating the relevance of a document with respect to a query, so that the documents can be ordered according to these estimates. Data structures are concerned with storing the corpus in a form that allows a fast estimation of the ranking function. User interfaces are concerned with designing an intuitive interface to query the system and to presents the resulting rankings. In this thesis, we focus solely on ranking functions.

In this section, we present the most common framework for ranking text documents from text queries, *the vector space model*. We then present how a similar framework can be used to retrieve other types of documents, such as images or speech recordings. Finally, we describe the standard methodology to evaluate retrieval systems.

2.1.1 Text Retrieval in the Vector Space Model

In text retrieval applications, the user is interested in retrieving text documents from a text query. For that purpose, an IR system typically performs a three-step process: text normalization, text representation and document ranking. Normalization removes any variability in the data that is not helpful for retrieval. Indexing gives to each document and query a representation suitable to the ranking function. Ranking takes as input the query and the documents in their indexed form and ranks the documents according to the ranking function.

Normalization applies stopping and stemming to documents and queries. Stopping eliminates functional words (e.g. articles, conjunctions, pronouns) and other *topic neutral* words, which are considered useless for retrieval. Stemming substitutes each inflected form of a word by its *stem* (e.g. *connected*, *connection* and *connects* will be replaced by *connect*). After normalization, documents and queries are available as sequences of *index terms*. The set of all index terms allowed by an IR system is called the *vocabulary*.

In the vector space model [Salton et al., 1975], indexing represents each document with a *bag-of-words* vector. This representation neglects word ordering and assigns each document a vocabulary-sized vector, the i^{th} component of a document vector being the weight of the i^{th} term of the vocabulary in the document. For a given document d, this weight d_i is computed from statistics of the occurrence of term i in d and in the rest of the corpus. The weighting strategy is typically designed so that the terms that better describes the document content are assigned higher weights. Different strategies have been proposed for that purpose [Luhn, 1958; Salton and Buckley, 1988], including the popular $tf \cdot idf$ weighting. In this case, the weight d_i of term i in document d is computed as

$$d_i = tf_{i,d} \cdot idf_i,$$

where $tf_{i,d}$ refers to the number of occurrences of i in d, $idf_i = -\log(r_i)$ refers to the inverse document frequency of term and r_i refers to the fraction of documents in the corpus that contain i. On one hand, tf ensures that the terms occurring frequently in a document are considered as better content descriptors than the terms occurring rarely. On the other hand, idf ensures that the terms appearing in few documents receive high weights, as those terms are more discriminative to distinguish between the topics of the documents. Similarly to documents, queries are also assigned a vector representation, relying on the same weighting scheme. Further discussion on term weighting and the description of more refined weighting strategies is differed to Chapter 3.

After indexing, ranking takes as input the indexed corpus D, along with the query vector q, and outputs a ranking in which the documents relevant to q should appear above the others. In order to perform this ranking, each corpus document d is assigned a score $f(q, d)$ according to a ranking function f, and the documents of D are then ordered by decreasing scores. f hence measures the matching between a document and the query. Its output is commonly referred as the Retrieval Status Value (RSV) of a document with respect to the query [Baeza-Yates and Ribeiro-Neto, 1999]. In the context of the vector space model, the vector dot-product is commonly used as the ranking function, i.e.

$$\forall (q, d) \in \mathbb{R}^T \times \mathbb{R}^T, \quad f(q, d) = q \cdot d = \sum_{i=1}^{T} q_i \; d_i,$$

where T refers to the number of allowed terms, i.e. the vocabulary size. Also, the cosine similarity is often discussed in the literature [Baeza-Yates and Ribeiro-Neto, 1999; van Rijsbergen, 1979], which is actually equivalent to the vector dot-product if one considers normalizing the L_2 norm of the document vectors during indexing.

This dot-product RSV considers that the matching between a document and a query should be high when their vector representations have their highest weights assigned to the same components. This means that the RSV of a document is high when the terms that best describe its content are also the terms that best describe the query content. This dot-product approach is simple and has shown to be effective in various retrieval contexts [Salton and Buckley, 1988; van Rijsbergen, 1979]. Furthermore, it offers a great efficiency advantage, when the vector representation of documents and queries are sparse. In the case of $tf \cdot idf$ vectors, only the components corresponding to terms present in the document or query are non-zero. This means that the RSVs of all documents in the corpus can be computed efficiently by only examining the components of the document vectors corresponding to the few query terms. Inverted indexes allows to quickly access to such data by storing, for each term i, the list of documents containing i along with the corresponding weights [Harman et al., 1992].

Relying on dot products of sparse vectors has however some drawbacks, including term mismatch and sub-optimal weighting. Term mismatch means that this approach does not take into account any correlation between terms: i.e. a query and a document with no term in common will be considered as completely unrelated, even if it is a possible to express the same topic using different terms, such as synonyms. Sub-optimal weighting means that the function assigning the term weights is not the best possible function relying on term

occurrence statistics, when one is interested in maximizing the retrieval performance. In the context of machine learning for IR, different strategies have been proposed to circumvent those limitations. Discussion on these approaches is differed to Section 2.2.

2.1.2 Retrieving Multimedia Documents

The previous section has presented an approach to search for text documents within large corpora. However, several application domains, such as broadcast news, stock photography or ancient manuscript archives, collect other types of documents in the form of audio recordings, images or videos and the application of retrieval techniques to non-textual data is hence of a great interest for such cases.

In order to search within non-textual datasets, different query interfaces have been devised. For instance, it has been proposed to search an image dataset by providing a query image [Smeulders et al., 2000], or a sketch [Rajendran and Shih-Fu, 2000]. However, such interfaces require a significant effort from the users, either to identify an image describing their needs, or to draw a query. Hence, text querying seems a better alternative from a user perspective. Also, most people are already used to efficiently search large textual corpora from text queries and would like to benefit from a similar interface to access to collections of pictures, audio recordings or videos, as illustrated by the design choice of retrieval services like [YouTube] for videos, [Corbis] for images or [Google Book Search] for scanned documents.

In the case of text querying, popular retrieval approaches rely on the use of texts that can be extracted from the original data. Automatic Speech Recognition (ASR) [Jelinek, 1998], Optical Character Recognition (OCR) [LeCun et al., 1998a] or image auto-annotation techniques (see Chapter 4) extract text data that are then used by a text retrieval system as a proxy for the original media. This two-step process, i.e. text extraction followed by text retrieval, has several advantages. First, it allows to benefit from the abundant research performed for the context of text retrieval. It also benefit from well-established text extraction techniques [Bunke and Wang, 1997; Jelinek, 1998; LeCun et al., 1998a]. Indeed, approaches based on this scheme have shown to yield effective solutions [Garofolo et al., 2000; Smeaton et al., 2006], even when the extraction process introduces a significant amount of errors in the resulting text [Garofolo et al., 1999; Vinciarelli, 2004]. Such a strategy is also appealing from an efficiency perspective since the costly text extraction operations are performed offline, i.e. before query submission, while only efficient text retrieval opera-

tions are performed online, i.e. after query submission, when the user interacts with the system.

However, these types of approaches also have some drawbacks, mainly because the text extraction process has not been optimized specifically for retrieval. For instance, speech recognizers are designed to achieve a low word error rate (the edit distance between the system output and and a manually-produced transcription). This objective is different from maximizing the performance of a retrieval system relying on ASR outputs. Indeed, alternative approaches relying on lattices, which summarize the confidence of the ASR system for several alternative transcriptions, are currently investigated [Mamou et al., 2007]. In this dissertation, Chapter 4 addresses this problem in the context of image retrieval and proposes a model which learns the parameters of the annotation model in order to maximize the retrieval performance.

2.1.3 Retrieval Evaluation

The goal of a retrieval system is to enable its users to access the relevant material of a given corpus through a querying interface. For that purpose, the output of an IR system given a query q is a ranking of documents in which the documents relevant to q should appear above the others. Retrieval evaluations hence measure how close the obtained ranking is to this ideal condition. Different measures have been proposed to quantify this effectiveness. Most of them are based on precision and recall.

Precision and recall assume that the evaluated strategy has retrieved a set of documents $S(q)$ and compare this set to the set of relevant documents $R(q)$: precision is defined as the percentage of retrieved documents that are actually relevant:

$$\mathrm{Pr}(q) = \frac{|R(q) \cap S(q)|}{|S(q)|} \tag{2.1}$$

while recall is defined as the percentage of relevant documents that have been retrieved:

$$\mathrm{Rec}(q) = \frac{|R(q) \cap S(q)|}{|R(q)|}. \tag{2.2}$$

Precision and recall hence evaluate a set of retrieved documents $S(q)$. For a retrieval ranking, they are measured at each rank n, considering that the set of retrieved documents $S(q)$ corresponds to the documents ranked above position n. These quantities, $\mathrm{Pr}_n(q)$ and $\mathrm{Rec}_n(q)$ measured at each rank n, can then be summarized by a *Precision versus Recall curve*, which plots precision as a function of recall. This type of plot is generally reported in retrieval benchmarks, such as Text REtrieval Conference (TREC) [Voorhees, 2006], and

can be useful to choose a system for different recall requirements. However, comparing curves is not a reliable way to compare retrieval approaches. As an alternative, different quantities have been defined, based on $\mathrm{Pr}_n(q)$ and $\mathrm{Rec}_n(q)$. In the following, we introduce precision at top 10, break-even point and average precision, which correspond to the most commonly used measures to evaluate IR systems [Baeza-Yates and Ribeiro-Neto, 1999].

Precision at top 10 (P10) corresponds to Pr_{10}. It evaluates the percentage of relevant documents within the top 10 positions of the ranking. This is a widely-used measure, which evaluates the percentage of relevant material a user would encounter on the first 10–result page of a search engine. Although it is easy to interpret, this measure tends to overweight simple queries with many relevant documents when averaging over a query set. For such queries, it is easier to rank some relevant documents within the top 10, simply because the relevance set is larger and not because of any property of the ranking approach. One should also note that the optimum of this measure is lower than 100% for queries with less than 10 relevant documents.

Break-Even Point (BEP), also called R-Precision [Aslam and Yilmaz, 2005], measures the percentage $\mathrm{Pr}_{|R(q)|}$ of relevant documents within the top $|R(q)|$ ranking positions, where $|R(q)|$ is the number of relevant documents for the evaluated query q. Contrary to P10, this measure does not overweight queries with many relevant documents.

Average Precision (AvgP) is the primary measure used in retrieval benchmark [Baeza-Yates and Ribeiro-Neto, 1999; Voorhees, 2006]. It corresponds to the average of the precision at each position where a relevant document appears,

$$\mathrm{AvgP}(q) = \frac{1}{|R(q)|} \sum_{d \in R(q)} \mathrm{Pr}_{rk(q,d))}(q),$$

where $rk(q, d)$ is the rank of document d for query q. It can also be shown that AvgP corresponds to the area under the Precision versus Recall curve [Buckley and Voorhees, 2000]. This means that AvgP also corresponds to the averaged precision performance, assuming a flat prior over the recall requirements.

The above presented measures, P10, BEP and AvgP, evaluate the performance over one query and are commonly averaged over a set of test queries, to estimate the expected performance the user will encounter when submitting a new query. This set of test queries is considered unavailable during system development. This avoids biasing design choices toward a specific set of evaluation queries, which would makes the measured average a poor estimator of the expected performance over a newly submitted query.

In this dissertation, we only consider binary relevance judgments, i.e. a document is either relevant or non-relevant, since it corresponds to the most common setup in the retrieval literature [Baeza-Yates and Ribeiro-Neto, 1999]. However, in some cases, human assessors further provide a gradual relevance judgment along with their binary decision. Different measures, such as Discounted Cumulative Gain [Jarvelin and Kekalainen, 2000], have been proposed to evaluate IR systems relying on this information. We refer to [Voorhees, 2001] for further discussion on this issue.

2.2 Machine Learning for Information Retrieval

Machine Learning (ML) studies algorithms that learn to solve data processing tasks given a limited set of data samples. For instance, some learning algorithms aim at removing noise from recorded speech, learning from a set of noise-free speech recordings [Attias et al., 2001], other approaches aim at discriminating between male and female faces in photographs, learning from a set of male and female face pictures [Moghaddam and Ming-Hsuan, 2002], etc. Learning approaches are widely-used in pattern recognition problem, such as speech recognition, fault detection or fingerprint recognition [Bishop, 2006], since the human process yielding the desired decision is difficult to formalize mathematically.

In this dissertation, we focus on the learning of ranking functions for IR systems. This means that we are interested in identifying a ranking function f from a *set of training data*, such that its expected performance on a new ranking problem is high. Two main types of learning approaches have been applied in this context, *supervised* and *unsupervised* learning. In the case of supervised learning, the training data consist of both documents and queries along with the corresponding relevance assessments. This means that the learning procedure should generalize to a new ranking problem, while having access to the desired output on the training data. In the unsupervised case, the training data simply consists in a set of documents, without queries and relevance assessments. As the learning procedure has no access to examples of the desired strategy, it should discover some hidden structure of the data, from which it is possible to identify an effective ranking function.

In this section, we first review unsupervised approaches and we then present supervised approaches, as this order is more meaningful from a chronological perspective.

2.2.1 Unsupervised Learning

The unsupervised learning of ranking functions only requires a set of documents for training, without queries and relevance assessments. This is appealing since large document corpora are readily available at virtually not cost, while the collection of queries and relevance assessments requires an expensive labeling process [Baeza-Yates and Ribeiro-Neto, 1999]. Therefore, several unsupervised models have been proposed in the literature. The following reviews the most influential approaches.

Latent Semantic Analysis

Latent Semantic Analysis (LSA) aims at modeling term correlation [Deerwester et al., 1990], to overcome the term mismatch problem (see Section 2.1.1). For instance, one of LSA's goals is to assign a high RSV to a document which does not use any query term, but only related terms or synonyms. For that purpose, LSA assumes that the vocabulary-sized vectors actually originate from a lower dimensional space ($k < T$, the vocabulary-size), to which orthogonal noise has been added. Given a set of n training documents, represented as a matrix

$$D = [d_1, \ldots, d_n] \in \mathbb{R}^{T \times n},$$

LSA solves the least square problem,

$$D^k = \text{argmin}_{X:\text{rank}(X)=k} \; ||D - X||_2^2. \tag{2.3}$$

and replaces D with $D^k = [d_1^k, \ldots, d_n^k]$ as the "denoised" representation of documents. The solution of Equation (2.3) can be found through Singular Value Decomposition (SVD), for which efficient iterative algorithms can be applied [Trefethen and Bau, 1997]. The original LSA paper [Deerwester et al., 1990] proposes to select k through validation, i.e. picking the k value which maximizes the performance over a set of development queries. It also devises a way to denoise test queries, which are not available when solving Problem (2.3).

The substitution of D with D^k actually projects each document to a k-dimensional subspace, and LSA hence assumes that the term mismatch problem can be solved through linear projection. Rank lowering is expected to merge the dimensions corresponding to terms occurring often in similar contexts. Previous works [Deerwester et al., 1990] have shown that this strategy can indeed improve retrieval results in some cases. However, LSA is rarely reported to improve performance when used alone and the LSA-derived RSV is often linearly combined [Dumais, 1995] to the standard RSV presented in Section 2.1.1. There is however no guarantee that the solution of the least square

problem (2.3) yields a better RSV according to standard retrieval measures, see Section 2.1.3. Actually, some reported results [Hofmann, 2001] have shown worse IR performance with LSA than with the initial data.

Subsequently to LSA, other approaches based on linear algebra have been proposed [Kolda and O'Leary, 2004; Li and Shawe-Taylor, 2007; Tsuge et al., 2001; B. et al., 2003]. For instance, Kernel Canonical Correlation Analysis, KCCA [Li and Shawe-Taylor, 2007], relies on a bi-lingual corpus to learn a lower rank representation of the data. This approach takes a corpus where each document is available in two languages and recursively identifies pairs of directions in both language vector space. The pairs of directions are selected such that the projections of each document in both languages are maximally correlated. After this process, documents and queries can be projected onto the subspace spanned by the identified directions of the corresponding language, and this new representation is used for retrieval. Although KCCA has been introduced for bi-lingual retrieval, this approach has also shown to be useful for monolingual setup, since the measure of correlation among languages avoids modeling grammatical specificities of a language, but rather focuses on semantic terms [Vinokourov et al., 2003].

Besides linear algebra, several probabilistic models have also been proposed to model term correlation and solve the term mismatch problem, as explained in the following.

Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis, PLSA [Hofmann, 2001], proposes a probabilistic interpretation of the notion of topics in text documents to address the term mismatch problem. PLSA assumes that the documents can be decomposed as a mixture of aspects, where each aspect defines a multinomial over the vocabulary terms. In this model, documents and terms are considered as the observation of two discrete random variables $\mathcal{D}$ and $\mathcal{T}$. The occurrence of a term t in a document d corresponds to the observation of the pair (t, d), which is modeled by the joint probability

$$P(t, d) = \sum_i P(z_i)P(t|z_i)P(d|z_i), \tag{2.4}$$

where the discrete random variable $\mathcal{Z}$, of values $z_1, \ldots, z_k$, is called the *aspect variable*. PLSA hence assumes that the term variable $\mathcal{T}$ is conditionally independent from the document variable $\mathcal{D}$, given the aspect variable $\mathcal{Z}$. The parameters of the model, i.e. $P(z_i), P(t|z_i), P(d|z_i)$ for all aspects z_i, all vocabulary terms t and all corpus documents d are learned to maximize the likelihood

of the pairs (t, d) occurring in the training corpus, relying on the Expectation Maximization (EM) algorithm [Hofmann, 2001]. Similarly to LSA, the PLSA paper suggests to select the number of aspect k through validation and also devises an approach to estimate $p(q|z_i), \forall i$ for a new query q, unavailable at training time. In a retrieval context, PLSA then proposes to compute the RSV of a document d with respect to query q as

$$f(q, d) = \sum_i P(z_i|q)P(z_i|d),$$

which corresponds to the expected likelihood kernel [Jebara and Kondor, 2003] between the aspect distributions in the document and in the query.

Several similarities can be found when comparing PLSA and LSA. Both models have been proposed to solve the term mismatch problem, by modeling the correlation between the occurrence of terms. Compared to LSA, PLSA relies on a probabilistic framework, which introduces normalization constraints on the parameters and replaces the mean square problem with a maximum likelihood problem. In fact, LSA aims at minimizing the Euclidean distance between the original data matrix and its lower rank surrogate, while PLSA aims at minimizing the Kullback-Leibler divergence between the empirical training distribution and the model distribution, as shown in [Hofmann, 2001]. There is no theoretical justification to prefer one criterion or the other in an IR context, i.e. in order to derive a ranking function which achieves higher retrieval effectiveness. It has however been shown that PLSA can be more effective than LSA on different corpora [Hofmann, 2001].

Besides this empirical advantage, the statistical framework of PLSA also allows systematic model combination in a Bayesian scheme, and the combination of several PLSA models relying on different numbers of aspects has shown to be effective [Hofmann, 2005]. The main drawback of PLSA is also related to its statistical foundation: PLSA derives its parameters from simple word counts and does not allow the direct use of more effective weighting scheme [Salton and Buckley, 1988] during learning, which is not the case for LSA.

Subsequently to PLSA, other latent topic models have been introduced [Blei et al., 2003; Gruber et al., 2007; Blei and Lafferty, 2005]. Latent Dirichlet Allocation, LDA [Blei et al., 2003], proposes a more consistent probabilistic approach compared to PLSA. Instead of relying on a document-dependent mixture, the distribution of topics within a document is assumed to be sampled from a Dirichlet distribution shared across documents. This means that LDA is a true generative model from which *unseen* documents can be sampled from the model. The dependency of topic distribution in documents introduced by

the Dirichlet prior also allows better regularization of the model, compared to PLSA that requires modifications of the training procedure to avoid overfitting [Hofmann, 2001]. Building upon LDA, the Hidden Topic Markov Model, HTMM [Gruber et al., 2007], introduces a Markovian dependency between the latent topics generating successive words, which hypothesizes that subsequent words of a document are more likely to share the same topic. Correlated Topic Model, CTM [Blei and Lafferty, 2005], is a further example of a latent topic model. It introduces a dependency between topics, acknowledging that the presence of one latent topic might be correlated with the presence of another topic.

These models hence make different dependence assumptions to model term correlation, and then rely on maximum likelihood estimation for parameter learning. Parameter selection is hence not performed according to an objective related to the final retrieval problem. Like for the least square objective of LSA, there is no guarantee that the parameters maximizing the training data likelihood would achieve a high ranking performance. This problem is in fact generic to unsupervised learning: on one hand, unsupervised learning can benefit from plentiful unlabeled data, while, on the other hand, this framework does not strongly tie the learning objective and the final task performance. In the next section, we present the supervised learning framework, which has the opposite characteristics, since this framework requires training queries and relevance assessments to optimize a learning objective closely related to the retrieval performance.

2.2.2 Supervised Learning

In the recent years, information retrieval has enjoyed a rising interest from the web search companies. This resulted in higher budget for manual data annotation, i.e. the definition of queries and the assessment of document relevance. Consequently, different supervised approaches relying on this type of data have been proposed. In the following, we review the most influential models proposed in this context.

Pair Classification

Pair classification formalizes the learning of ranking functions as a binary classification problem. Given a query q and a document d, the ranking function f should determine whether (q, d) is a positive pair, i.e. d is relevant to q, or a negative pair, i.e. d is not relevant to q. This formalization of the problem

allows the retrieval community to benefit from the abundant research on the learning of classification models [Bishop, 2006].

This framework has been applied relying on different types of classifiers, such as neural networks [Mandl, 2000] or decision trees [Hatzivassiloglou et al., 1999]. However, this initial research did not foster a long term effort on pair classification for information retrieval. The lack of labeled data at that time might explain this failure. However, this framework also suffers two main weaknesses, *unbalanced* classification and *inter-query* discrimination. Unbalanced classification refers to a binary classification problem for which one class is predominant. In the case of the classification of query/document pairs, the negative class is predominant and the positive pairs account only for a very small fraction of the whole pair set. For instance, TREC retrieval datasets present less than 1% positive pairs [Voorhees, 2006]. This characteristic is a problem for most classification approaches which aim at maximizing the classification accuracy (the percentage of correctly classified pairs), since the useless model that always predicts the negative class would achieve more than 99% accuracy. In fact, problems with $\sim 10\%$ positive examples already constitute a challenging unbalanced classification setup, for which specific learning solutions are being devised [Grandvalet et al., 2005]. Inter-query discrimination refers to an intrinsic problem of the pair classification framework, which presents a more difficult problem to the classifier than the actual retrieval task. In this framework, the classifier should output a positive score $f(q, d) > 0$ for any positive pair (q, d) and a negative score $f(q', d') < 0$ for any negative pair (q', d'). However, in retrieval rankings, only the scores of the documents for the same query need to be compared (see Section 2.1.1), and there is hence no need to discriminate between (q, d) and (q', d'), when $q' \neq q$. In fact, this framework ignores that each query presents its own document classification problem, and attempts to solve all problems with the same classifier. In the following, we describe ordinal regression, which proposes to circumvent these shortcomings.

Ordinal Regression

Ordinal regression refers to the problem of predicting an ordering over a set of input items. *Recommender systems* is an instance of such a problem. In this case, the user orders their movies, books or music albums according to their preferences, and the system should predict a preference ordering over a large inventory of unseen items. Formally, the input to such a problem consists in a set of labeled training examples,

$$S_{\text{train}} = \{x_i, y_i\}_{i=1}^{N},$$

where for all i, x_i is vector from a space $\mathcal{X}$, and y_i is an integer preference value between 1 and m. The labels express an order preference, i.e. given (x_i, y_i) and (x_j, y_j), $y_i > y_j$ means that x_i should be ranked above x_j. The training algorithm then learns a function,

$$h : \mathcal{X} \to \mathbb{R},$$

whose output should ideally indicate preference, i.e. for all (x, y) and (x', y') in $\mathcal{X} \times \{1, \ldots, m\}$,

$$y > y' \Leftrightarrow h(x) > h(x').$$

For that purpose, solutions based on Support Vector Machines (SVMs) [Herbrich et al., 2000] and boosting [Freund et al., 2003] have been proposed. They both aim at minimizing an upper bound on the number of swapped training pairs, i.e. the number of training pairs (i, j) for which $h(x_i) \leq h(x_j)$ while $y_i > y_j$.

The retrieval setup differs from this standard ordinal regression framework, since each query presents a different ordinal regression problem and the model needs to generalize to both new queries and new documents. Formally, an IR system should order the corpus documents such that the relevant documents appear above the others, for any query. This means that the ranking function f should satisfy,

$$\forall q, \quad \forall (d^+, d^-) \in R(q) \times \overline{R}(q), \qquad f(q, d^+) > f(q, d^-), \tag{2.5}$$

where $R(q)$ refers to the relevant corpus documents and $\overline{R}(q)$ refers to the non-relevant corpus documents. The Ranking SVM model (RSVM) [Joachims, 2002] builds upon SVMs for ordinal regression [Herbrich et al., 2000] and introduces an approach to learn ranking functions for retrieval. In this model, the ranking function is parameterized as

$$\forall (q, d), \quad f_{\mathbf{w}}(q, d) = \mathbf{w} \cdot \phi(q, d),$$

where $\mathbf{w}$ is a parameter vector from a space $\mathcal{F}$ and $\phi(q, d)$ is a vector of $\mathcal{F}$ characterizing the matching between q and d. The parameter vector is selected as the solution of

$$\min_{\mathbf{w}} \|\mathbf{w}\|_2^2 + \ C \ \mathrm{L}^{\mathrm{RSVM}}(\mathbf{w}; Q_{\mathrm{train}}), \tag{2.6}$$

where $\|\mathbf{w}\|_2^2$ is a regularizer, $\mathrm{L}^{\mathrm{RSVM}}(\mathbf{w}; Q_{train})$ is a loss function defined from a set of training queries Q_{train} and C is a hyperparameter controlling the regularization strength. The loss corresponds to

$$\mathrm{L}^{\mathrm{RSVM}}(\mathbf{w}; Q_{\mathrm{train}}) \quad = \sum_{q \in Q_{\mathrm{train}}} l(\mathbf{w}; q)$$

where $l(\mathbf{w}; q)$ measures the loss on the ordinal regression problem corresponding to query q,

$$l(\mathbf{w}; q) \quad = \sum_{(d^+,d^-)\in R(q)\times\overline{R}(q)} \max\left(0, 1 - f_{\mathbf{w}}(q, d^+) + f_{\mathbf{w}}(q, d^-)\right).$$

This query specific loss is an upper-bound on the number of swapped pairs for the query ranking,

$$l(\mathbf{w}; q) \quad \geq \sum_{(d^+,d^-)\in R(q)\times\overline{R}(q)} \mathbb{1}_{f_{\mathbf{w}}(q,d^+)\leq f_{\mathbf{w}}(q,d^-)}$$

since for all (q, d^+, d^-),

$$\max\left(0, 1 - f_{\mathbf{w}}(q, d^+) + f_{\mathbf{w}}(q, d^-)\right) \geq \mathbb{1}_{f_{\mathbf{w}}(q,d^+)\leq f_{\mathbf{w}}(q,d^-)},$$

where $\mathbb{1}_{\cdot}$ denotes the indicator function.

Hence, the optimization problem (2.6) selects the ranking function as a trade-off between minimizing $\mathrm{L}^{\mathrm{RSVM}}(\mathbf{w}; Q_{\mathrm{train}})$, which implies minimizing the total number of swapped pairs over the rankings of all training queries, and having a regular model, with a small norm $\|\mathbf{w}\|$. The hyperparameter C controls this trade-off. This regularization scheme is introduced to prevent *overfitting*, i.e. the case where the model would achieve high ranking performance over the training data while performing poorly over unseen test data. In fact, the regularization strategy of the ranking SVM presents theoretical guarantees toward high generalization performance as discussed later in Chapter 4.

Compared to pair classification, this ordinal regression framework presents several advantages. First, the training strategy does not rely on the classification error, but on the number of swapped pairs in rankings, which is more adapted to unbalanced setups [Cortes and Mohri, 2003] like retrieval. Second, losses such as $\mathrm{L}^{\mathrm{RSVM}}$ avoid comparing the output of the learned ranking function among queries, and hence does not suffer the inter-query discrimination problem. In fact, RSVM and related approaches [Burges et al., 2005] have shown to yield high performance [Joachims, 2002; Schultz and Joachims, 2003; Burges et al., 2005] over various retrieval problems. On the negative side, the optimization of losses like $\mathrm{L}^{\mathrm{RSVM}}$ can become very costly for large document sets, as it relie on a sum over the pairs of documents with different labels.

Direct Optimization of Ranking Measures

In the ordinal regression framework, the model parameters are selected to minimize the number of swapped pairs in rankings, i.e. the number of times a

non-relevant document appear above a relevant one. This strategy is directly derived from the definition of the retrieval ranking problem (2.5). However, the number of swapped pairs in a ranking is rarely used as a measure of retrieval quality. Instead, measures such as P10, AvgP and BEP are generally used, see Section 2.1.3. Therefore, different approaches have been proposed to directly optimize such measures [Joachims, 2005; Yue et al., 2007; Le and Smola, 2007]. These approaches relies on the framework of SVM for structured prediction [Tsochantaridis et al., 2005] and maximize a convex lower bound on such measures.

Learning structured prediction aims at finding a mapping $g_{\mathbf{w}}$ from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$, given a labeled training set

$$S_{\text{train}} = \{x_i, y_i\}_{i=1}^{N} \in (\mathcal{X} \times \mathcal{Y})^N.$$

Learning is performed to minimize the loss

$$\mathrm{L}^{\text{Struct}}(g_{\mathbf{w}}, S) = \sum_{i=1}^{N} \Delta(y, g_{\mathbf{w}}(x_i)),$$

where $\Delta(y, y') \geq 0$ measures the cost of predicting y' instead of y. The structured prediction framework is not tied to a specific type of cost but provides a generic framework to minimize any loss that can be formalized as $\mathrm{L}^{\text{Struct}}$. For that purpose, the learning of $g_{\mathbf{w}}$ is reduced to learning a real valued function $h_{\mathbf{w}}$ over the joint input-output space $\mathcal{X} \times \mathcal{Z}$, and $g_{\mathbf{w}}$ is defined as,

$$\forall x \in \mathcal{X}, \quad g_{\mathbf{w}}(x) = \text{argmax}_{z \in \mathcal{Z}}\ h_{\mathbf{w}}(x, z). \tag{2.7}$$

The function $h_{\mathbf{w}}$ is parameterized as,

$$\forall (x, z) \in \mathcal{X} \times \mathcal{Z}, h_{\mathbf{w}}(x, z) = \mathbf{w} \cdot \Psi(x, z),$$

where $\mathbf{w}$ is a parameter vector and $\Psi(x, z)$ is a set of features extracted from (x, z). The parameter vector $\mathbf{w}$ is selected as the solution of

$$\begin{aligned} \min_{\mathbf{w}} \quad & \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N} \xi_i, \\ \text{s.t.} \quad & \forall i \in \{1, \ldots, N\}, \quad \begin{array}{l} \forall z \in \mathcal{Z}, \quad h_{\mathbf{w}}(x_i, z_i) - h_{\mathbf{w}}(x_i, z) \geq \Delta(z_i, z) - \xi_i, \\ \xi_i \geq 0, \end{array} \end{aligned} \tag{2.8}$$

where C is an hyperparameter controlling the regularizer strength. In Problem (2.8), it can be shown that the slack variables ξ bound the cost, i.e.

$$\forall i, \quad \xi_i \geq \Delta(z_i, \text{argmax}_{z \in \mathcal{Z}}\ h_{\mathbf{w}}(x_i, z)),$$

which means that the minimization of $\sum_{i=1}^{N} \xi_i$ yields the minimization of $\mathrm{L}^{\mathrm{Struct}}(g_{\mathbf{w}}, S)$. Problem (2.8) can be solved efficiently through the cutting plane method [Tsochantaridis et al., 2005]. This technique iterates through the examples and requires to solve

$$\max_{z \in \mathcal{Z}} h_{\mathbf{w}}(x_i, z) + \Delta(z_i, z). \tag{2.9}$$

at each iteration. Hence, structured prediction approaches should take care that both training and testing are computationally tractable. This means that the choice of Ψ and Δ should take into account that the maximizations required by (2.7) and (2.9) can be solved efficiently.

For retrieval rankings, a structured prediction model takes as input a pair $x = (q, D)$, composed of query q and a document set D, and outputs a permutation z over the elements of D. It aims at minimizing a cost Δ equal to $1 - M$, where M corresponds to a standard retrieval measure such as P10, BEP [Joachims, 2005] or AvgP [Yue et al., 2007]. The choice of $\Psi(q, D, z)$ makes the function $h_{\mathbf{w}}(q, D, z)$ decomposable, i.e.

$$\phi(q, D, z) = \sum_{d \in D} c_{rk(z;d)} \ \phi(q, d),$$

where $rk(z; d)$ refers to the rank of document d assigned by permutation z, c_i is a weight specific to a rank i and $\phi(q, d)$ is a vector characterizing the matching between q and d like for the RSVM. This definition implies that

$$\begin{aligned} h_{\mathbf{w}}(q, D, z) &= \mathbf{w} \cdot \Psi(q, D, z) \\ &= \mathbf{w} \cdot \sum_{d \in D} c_{rk(z;d)} \ \phi(q, d) \\ &= \sum_{d \in D} c_{rk(z;d)} \ \mathbf{w} \cdot \phi(q, d) \\ &= \sum_{d \in D} c_{rk(z;d)} \ f_{\mathbf{w}}(q, d), \end{aligned}$$

where $f_{\mathbf{w}}(q, d)$ corresponds to the same function as for the RSVM. Furthermore, the weights c are selected as a decreasing function of the rank, i.e. $\forall i < j, c_i > c_j$, which means that the maximization required to evaluate g,

$$g_{\mathbf{w}}(q, D) = \mathrm{argmax}_{z \in \mathcal{Z}} \sum_{d \in D} c_{rk(z;d)} f_{\mathbf{w}}(q, d),$$

can be performed by ranking $\{f_{\mathbf{w}}(q, d), \forall d \in D\}$ decreasingly, like for the RSVM. Hence, the structured prediction approach is identical to the RSVM at test time, only the training procedure differs. The main difficulty for applying structure prediction lies in the efficient resolution of (2.9) required for training.

Specific algorithms have been recently proposed to solve this problem for the optimization of various retrieval measures [Joachims, 2005; Yue et al., 2007; Le and Smola, 2007]. These solutions are however expensive. For instance, the SVM optimizing average precision requires a sort operation, i.e. $O(|D| \log |D|)$, to solve (2.9), which is costly for large datasets as this operation is performed at each training iteration.

SVMs for structure prediction is hence a way to optimize retrieval measures. Other approaches based on neural networks [Burges et al., 2006] or gradient boosting [Zheng et al., 2007] have also been proposed to maximize retrieval measures focussing on the first positions of ranking. Overall, these approaches are appealing, as their learning processes focus on the final goal [Vapnik, 1982]. However, training those models represents a costly optimization problem for large document sets. In practice, they are hence applied only to re-rank the best documents identified by another approach [Burges et al., 2006; Zheng et al., 2007].

2.3 Conclusions

This chapter has presented the background for Information Retrieval. It then introduced how Machine Learning has been applied to this domain, which is the core topic of this thesis. This chapter stresses that the first learning techniques specific to retrieval ranking have been introduced only recently, less than a decade ago. Since then, retrieval problems have increasingly received attention from the machine learning community, certainly due to the rising web search business. IR problems are however far from being solved, and several open issues remain. In the next chapters, we address some of these problems. First, Chapter 3 presents how term weighting functions can be learned from unlabeled text data with hyperlinks. Then, Chapter 4 addresses the problem of retrieving images from text queries as a supervised ranking problem, which significantly differs from previous approaches based on automatic annotation. Finally, Chapter 5 builds upon the discriminative strategy proposed in Chapter 4 to retrieve speech recordings from written keywords. This requires a model specifically tailored to the sequential nature of this problem. Finally, Chapter 6 summarizes the contributions of the thesis.

3 Learning Text Similarity from Hyperlinked Data

This chapter presents a method to infer an effective measure of similarity between text items from an unlabeled hypertext corpus. This method is of great interest in the context of text retrieval, where the assessment of similarity plays a crucial role. In order to produce a ranking in which the documents relevant to the submitted query appear above the others, IR systems generally rank documents according to the output of a ranking function that estimates the similarity between a query and a document, as explained in Chapter 2. The identification of an effective measure of similarity is hence of crucial importance to IR performance. This identification can be performed through supervised learning relying on a large set of queries with their corresponding relevant document sets. However, such data are scarce due to the high cost of relevance labeling. As an alternative, this chapter proposes to rely on abundant hyperlinked data for learning. Hyperlinks actually convey information about the semantic proximity of documents, and we hypothesize that this proximity information is close to the document/query relationships provided by relevance assessments.

In the following, we introduce a neural network, *LinkLearn*, which learns a similarity measure from the proximity information embedded in a hyperlinked corpus. The link information is used solely for supervision at training time. At test test time, the inferred function can be applied over any textual corpus, with or without hyperlinks. The parameterization of our network is based on the vector space model. *LinkLearn* computes the similarity between two text items d and d' as the dot-product between vocabulary-size vector, i.e.

$$f(d, d') = d \cdot d'.$$

In each vector d, the i^{th} component d_i is computed through a *weighting function* g, from the statistics of the occurrences of term i in d and in the whole corpus.

Traditionally, [Salton and Buckley, 1988; Robertson et al., 1994], the weighting function is hand-crafted to assign the highest weights to the terms that best describes the document content. Alternatively, *LinkLearn* introduces a parametric weighting function $g_{\mathbf{w}}$, whose parameters are learned from hyperlinked data.

This approach has several advantages. First, the semantic relationships needed for training, hyperlinks, can be easily obtained without any additional labeling effort. For instance, the experiments presented in Section 3.3 rely on the online encyclopedia *Wikipedia* to train our model. Second, the parameterization of *LinkLearn* allows for the use of fast techniques developed for sparse vector comparison (e.g. inverted index files [Harman et al., 1992]), and the model can hence be applied efficiently over large corpora ($> 100,000$ documents). Finally, *LinkLearn* shows an empirical advantage when compared to state-of-the art weighting techniques, such as *Okapi* [Robertson et al., 1994]. For instance, we report +11% improvement for P10 when comparing *LinkLearn* to *Okapi* over TREC-9 queries [Voorhees, 2000].

The remainder of this chapter is divided into four sections: Section 3.1 introduces *LinkLearn*, Section 3.2 relates our approach to previous work, Section 3.3 presents the experiments and results and Section 3.4 draws some conclusions.

3.1 LinkLearn

LinkLearn aims at learning a measure of text similarity from a hyperlinked corpus. The learned model can then be applied to compare any text items, with or without hyperlinks. It can notably be used to compare documents and queries in the context of IR.

In the following, we first motivate the use of hyperlinks as an indicator of semantic proximity and present a brief overview of methods already taking advantage of this characteristic. Then, we introduce the model parameterization and its training procedure.

3.1.1 Semantic Proximity of Hyperlinked Documents

The concept of hyperlink is inspired by encyclopedia cross-references: both provide authors with the possibility to refer to documents related to their writings. Even if hyperlinks may be used for other purposes (e.g. navigational links, advertising links, etc), it has been observed that they are mostly used for their primary goal [Davison, 2000]. This observation is also supported by previous research which relies on the semantic proximity of linked documents

to improve the way document similarity is assessed.

In [Brin and Page, 1998], document expansion is performed using hyperlink information: the weight of a term i in a document d is increased when i occurs in documents pointing to d. This approach has shown to be especially useful in the case of short documents whose retrieval is difficult without these added terms [Brin and Page, 1998].

In [Richardson and Domingos, 2002], an alternative approach is adopted. The document/query similarity is computed iteratively relying on the following idea: the similarity between a document d and a query q should depend on the similarity between q and the documents linked with d. Hence, in this framework, a document which is connected to documents highly similar to q, will itself be assigned a higher similarity with respect to q. This approach extends of the *PageRank* algorithm [Page et al., 1998] and hypothesizes that, among the documents having a content similar to the query, the most valuable are those which are referred to by the others.

Probabilistic Latent Semantic Analysis, PLSA [Hofmann, 2001], has also been extended to benefit from hyperlink data [Cohn and Hofmann, 2000]: a link to a document d is considered as a kind of additional term l_d and hence, when computing document similarity, documents sharing pointers to the same references are considered closer.

The above approaches modify the way document similarity is computed, so that it considers linked web-pages as more likely to be about the same topic than unlinked ones [Davison, 2000]. Our proposed approach *LinkLearn* shares the same objective during training. However, *LinkLearn* aims at generalizing to non-hyperlinked data, and only relies on hyperlinks as indicators of semantic proximity for model learning. In that sense, our approach differs from previous work.

3.1.2 Model Parameterization

LinkLearn aims at identifying a parameter vector $\mathbf{w}$ such that the measure of similarity

$$\mathrm{sim}_{\mathbf{w}} : (a, b) \rightarrow \mathrm{sim}_{\mathbf{w}}(a, b),$$

generally considers linked documents closer than unlinked ones. For that purpose, this section introduces the parametric form of $\mathrm{sim}_{\mathbf{w}}$, while next section introduces the training procedure to select $\mathbf{w}$.

The parameterization of $\mathrm{sim}_{\mathbf{w}}$ is inspired from ad-hoc approaches proposed for the vector space model, see Chapter 2. In *LinkLearn*, the similarity is

assessed through the dot product of vocabulary-sized document vectors,

$$\text{sim}_{\mathbf{w}} : a, b \rightarrow \sum_{i=1}^{T} a_i^{\mathbf{w}} b_i^{\mathbf{w}},$$

where T refers to the vocabulary size, and $a_i^{\mathbf{w}}$ (resp. $b_i^{\mathbf{w}}$) refers to the weight of term i in document a (resp. b). The weight of a term i in a document d is computed as,

$$d_i^{\mathbf{w}} = g_{\mathbf{w}}(tf_{i,d}, idf_i, ndl_d)$$

where $tf_{i,d}$ is the term frequency of i in d, idf_i is the inverse document frequency of term i and ndl_d is the normalized length of document d. tf and idf corresponds to the quantities defined in Section 2.1.1, while ndl_d corresponds to l_d/L, where l_d is the length of d, $l_d = \sum_{i=1}^{T} tf_{i,d}$, and L corresponds to the mean document length over the whole corpus from which d is extracted.

The specific form of $g_{\mathbf{w}}$ is

$$g_{\mathbf{w}}(tf_{i,d}, idf_i, ndl_d) = \mathbb{1}_{\{tf_{i,d} \neq 0\}} \; MLP_{\mathbf{w}}(tf_{i,d}, idf_i, ndl_d)$$

where $MLP_{\mathbf{w}}$ denotes a Multi-Layered Perceptron (MLP) [Bishop, 1995]. The parameterization of our model involves the indicator function, $\mathbb{1}_{\cdot}$. However, it should be stressed that this does not prevent $\mathbf{w} \rightarrow g_{\mathbf{w}}(tf_{i,d}, idf_i, ndl_d)$ to be a continuous function of $\mathbf{w}$, derivable everywhere.

Our similarity measure is based on dot-product matching, and therefore inherits the main efficiency advantage of this framework in a retrieval context (see Chapter 2, Section 2.1.1). The form of $\text{sim}_{\mathbf{w}}$ allows one to perform most of the computations offline, i.e. before the user interacts with the system. In an IR setup, all documents are generally available prior to query submission, and our parameterization allows the computation of all the weights $d_i^{\mathbf{w}}$ for each term i and each document d before the query is available. Such a strategy favors the IR system response time since only the query weights are computed after the user submits the query. Furthermore, the term $\mathbb{1}_{tf_{i,d} \neq 0}$ ensures that $d_i^{\mathbf{w}} = 0$, when $tf_{i,d} = 0$, i.e. when term i is not present in d. This property is of great interest, regarding the efficient evaluation of $\text{sim}_{\mathbf{w}}$. In fact, the evaluation of $\text{sim}_{\mathbf{w}}(a, b)$ only requires to evaluate the MLP outputs for the terms present in both documents, which is typically much smaller than 2 T, the total number of components in a and b vectors. Moreover, this strategy also allows the use of efficient data structure for sparse vectors [Harman et al., 1992].

In addition to these efficiency aspects, the proposed parameterization has also good properties regarding generalization. The measure $\text{sim}_{\mathbf{w}}$ only relies on simple features of term occurrences which makes it *vocabulary-independent*, i.e.

the learned parameters are not linked to a specific term set and the function $\text{sim}_{\mathbf{w}}$ inferred from one corpus can therefore be applied to another corpus, possibly indexed with a different vocabulary, as shown by our TREC experiments in Section 3.3. This aspect is an advantage when comparing *LinkLearn* to other data-driven approaches like PLSA [Hofmann, 2001] or *Ranking SVM* [Schultz and Joachims, 2003] describerd in Chapter 2.

3.1.3 Model Criterion and Training

The objective of *LinkLearn* is to build a reliable text similarity measure from a hyperlinked corpus. In such a corpus, links convey information about the semantic proximity of documents. In particular, it has been observed [Davison, 2000] that, in most cases, a document d is semantically closer to a document l^+, hyperlinked with d, than to a document l^-, not hyperlinked with d:

$$\forall d, \forall l^+ \in H(d), \forall l^- \in \overline{H(d)}, \text{sim}(d, l^+) - \text{sim}(d, l^-) > 0, \tag{3.1}$$

where $H(d)$ refers to the set of documents hyperlinked with d (i.e. the documents referring to d and the documents referred to by d), and $\overline{H(d)}$ refers to the other corpus documents. This proximity relationships are of a great interest for retrieval applications, since these kinds of relationships are analogous to relevance assessments that state that a query q is semantically closer to a document d^+, relevant to q, than to a document d^-, not relevant to q.

To benefit from such proximity information, we propose to learn $\mathbf{w}$ such that $\text{sim}_{\mathbf{w}}$ satisfies most constraints (3.1) for a hyperlinked training dataset D_{train}. For that purpose, we introduce a loss function that measures how close $\text{sim}_{\mathbf{w}}$ is to the objective (3.1) and we select $\mathbf{w}$ that minimizes this loss.

A simple loss to minimize in this context would be the fraction of constraints which are not satisfied,

$$\text{L}(\mathbf{w}; D_{\text{train}}) = \frac{1}{|D_{\text{train}}|} \sum_{d \in D_{\text{train}}} \text{L}(\mathbf{w}; d), \tag{3.2}$$

where

$$\text{L}(\mathbf{w}; d) = \frac{1}{|H(d)| \cdot |\overline{H(d)}|} \sum_{\substack{l^+ \in H(d) \\ l^- \in \overline{H(d)}}} \mathbb{1}_{\{\text{sim}_{\mathbf{w}}(d,l^+) - \text{sim}_{\mathbf{w}}(d,l^-) \leq 0\}}.$$

However, this loss is difficult to minimize directly since its gradient is null everywhere. We hence propose to minimize an upper bound of this quantity:

$$\text{L}^{\text{LinkLearn}}(\mathbf{w}; D_{\text{train}}) = \frac{1}{|D_{\text{train}}|} \sum_{d \in D_{\text{train}}} \text{L}^{\text{LinkLearn}}(\mathbf{w}; d), \tag{3.3}$$

where

$$\mathrm{L}^{\mathrm{LinkLearn}}(\mathbf{w}; d) = \frac{1}{|H(d)| \cdot |\overline{H(d)}|} \sum_{\substack{l^+ \in H(d) \\ l^- \in \overline{H(d)}}} \max(0, 1 - \mathrm{sim}_{\mathbf{w}}(d, l^+) + \mathrm{sim}_{\mathbf{w}}(d, l^-)),$$

This loss is actually an upper bound of L since, for all x,

$$\max(0, 1 - x) \geq I\{x < 0\}.$$

Furthermore, if there exists a minimum such that $\mathrm{L}^{\mathrm{LinkLearn}}(\mathbf{w}; D_{\mathrm{train}}) = 0$, it also implies that $\mathrm{L}(\mathbf{w}; D_{\mathrm{train}}) = 0$, which means that all the constraints (3.1) are satisfied.

The minimization of $\mathrm{L}^{\mathrm{LinkLearn}}$ can be performed through stochastic gradient descent, i.e. we iteratively pick a document in D_{train} and update $\mathbf{w}$ according to $\partial \mathrm{L}^{\mathrm{LinkLearn}}(\mathbf{w}; d)/\partial \mathbf{w}$, see Algorithm 3.1. This training strategy both offers great scalability advantage, and robustness toward poor local minima [LeCun et al., 1998b]. The hyperparameters of the model (i.e. the number of hidden units in the MLP, the number of training iterations and the learning rate) are selected through cross-validation (see Section 3.3).

Algorithm 3.1: Stochastic Training Procedure

Input: Training set D_{train}, learning rate λ, number of iterations N_{iter}
Initialize $\mathbf{w}$ randomly.
foreach i *in* $1, \ldots, N_{\mathrm{iter}}$ **do**
  Sample d in D_{train},
  compute $\partial \mathrm{L}^{\mathrm{LinkLearn}}(\mathbf{w}; d)/\partial \mathbf{w}$,
  update $\mathbf{w} \leftarrow \mathbf{w} + \lambda \, \partial \mathrm{L}^{\mathrm{LinkLearn}}(\mathbf{w}; d)/\partial \mathbf{w}$.
end
Output: Parameter vector $\mathbf{w}$

3.2 Related Work

LinkLearn combines ideas from previous research in information retrieval and machine learning. This section first describes the links between the parameterization of *LinkLearn* and prior work on term weighting. It then describes the links between our learning objective and approaches to learning to rank.

3.2.1 Prior Work on Term Weighting

The parameterization of our model relies on the vector space model [Salton et al., 1975] and proposes to learn the term weighting function from data.

Term weighting has been extensively studied in the retrieval literature [Salton and Buckley, 1988; van Rijsbergen, 1979; Baeza-Yates and Ribeiro-Neto, 1999]. The goal of term weighting is to assign a real value weight d_i reflecting the importance of term i as a content descriptor of document d.

In the early days of IR [Lancaster, 1979], such weights were simply binary values, with $d_i = 1$ for each term i present in document d and $d_i = 0$ for each term i absent from document d. In this case, the dot-product matching between two documents d and d',

$$\text{sim}(d, d') = d \cdot d'$$

is equivalent to counting the number of terms shared by d and d'.

Term weighting has then been extended to real-valued vectors [Cooper, 1988] in order to considers some terms as more important than others. Terms occurring frequently in a document have been assigned higher weights, while non-discriminative terms occurring in most documents have been assigned lower weights. This strategy is notably embodied in the popular tf idf weighting introduced in Chapter 2. Then, different refinements of this approach have been investigated [Salton and Buckley, 1988]. These ad-hoc weighting strategies propose different functions of the term occurrence statistics, with the objective to improve the retrieval performance. Among them, *Okapi BM25* is considered as one of the most effective [Robertson et al., 1994]. In this case, the weight of term i in document d is computed as

$$d_i^{\text{Okapi}} = \frac{(K+1) \cdot tf_{d,i} \cdot idf_i}{K \cdot ((1-B) + B \cdot ndl_d) + tf_{d,i}},$$

where (K, B) are two hyperparameters. The choice of (K, B) is generally performed through cross-validation, i.e. the parameters are selected to maximize the performance over a set of *validation* queries.

Like any hand-crafted weighting function, *Okapi* suffers from its simplicity: the selection of two parameters only allows us to optimize the average performance over a query set but prevents us from obtaining a function which consistently reaches high performance for each query. This underfitting problem is hard to circumvent with manually designed weighting scheme.

LinkLearn addresses this problem and learns the weighting function from data. Our approach is based on an MLP relying on the same inputs as *Okapi*. Hence, considering that MLPs are universal approximators [Bishop, 1995], the *Okapi* weights can even be computed in our framework. However, our strategy allows to infer more complex functions through learning. Indeed, the capacity of our model can be selected depending on the amount of data available for training and the desired regularity of the learned function.

3.2.2 Prior Work on Learning to Rank

Our approach aims at learning from proximity information such as

"document a should be closer to document b than it is to document c,"

which is also the objective of ranking SVM [Joachims, 2002]. For that purpose, ranking SVM builds upon ordinal regression approaches such as [Freund et al., 2003] or [Herbrich et al., 2000], as explained in Chapter 2. The parameterization of the learned function is

$$f_{\mathbf{w}}(d, d') = \mathbf{w} \cdot \phi(d, d')$$

where $\mathbf{w}$ is the parameter vector and $\phi(d, d')$ is a vector characterizing the match between d and d'. Hence, this linear parameterization has so far been used either for combining existing matching measures [Joachims, 2002] or for learning a weight per term [Schultz and Joachims, 2003]. This simple setup could be extended to more complex parameterization with non-linear models relying on kernels [Burges, 1998]. However, a richer parameterization would certainly require a larger set of proximity constraints for training an effective model. This could be problematic as the observed training complexity of *Ranking SVM* is $O(|P_{\text{train}}|^p), 2 < p \leq 3$, where $|P_{\text{train}}|$ refers to the number of training constraints [Joachims, 1998].

RankNet [Burges et al., 2005] is a gradient based approach to similarity measure learning, which addresses this issue. It proposes a non-linear models that can be trained efficiently, at the expense of relying on non-convex optimization. Like *Ranking SVM*, *RankNet* is also trained from a set of proximity constraints P_{train},

$$\forall (a, b, c) \in P_{\text{train}}, \text{sim}(a, b) > \text{sim}(a, c).$$

In this case, each $(a, b, c) \in P_{\text{train}}$ can additionally be labeled with $p_{a,b,c}$, the probability that constraint (a, b, c) is actually true. This framework can be helpful when, for instance, the constraints originate from several annotators that may disagree. When such probabilities are not available, it can simply be assumed that $\forall (a, b, c) \in P_{\text{train}}, p_{a,b,c} = 1$.

RankNet learns a parameterized similarity measure relying on MLP,

$$\text{sim}_{\mathbf{w}}^{\text{RankNet}}(d, d') = MLP_{\mathbf{w}}(\phi(d, d')).$$

Learning use gradient descent to minimize a Cross Entropy (CE) loss,

$$\mathrm{L}^{\mathrm{CE}}(\mathbf{w}; P_{\mathrm{train}}) = \sum_{(a,b,c)\in P_{\mathrm{train}}} \mathrm{L}^{\mathrm{CE}}(\mathbf{w}; a, b, c), \tag{3.4}$$

$$\text{where} \quad \mathrm{L}^{\mathrm{CE}}(\mathbf{w}; a, b, c) = -p_{a,b,c} \log o_{a,b,c} - (1 - p_{a,b,c}) \log(1 - o_{a,b,c}),$$

$$\text{and} \quad o_{a,b,c} = \frac{\exp(\mathrm{sim}_{\mathbf{w}}^{\mathrm{RankNet}}(a, b) - \mathrm{sim}_{\mathbf{w}}^{\mathrm{RankNet}}(a, c))}{1 + \exp(\mathrm{sim}_{\mathbf{w}}^{\mathrm{RankNet}}(a, b) - \mathrm{sim}_{\mathbf{w}}^{\mathrm{RankNet}}(a, c))}.$$

Contrary to *Ranking SVM*, *RankNet* relies on gradient descent optimization [LeCun et al., 1998b] and its training cost grows linearly with respect to $|P_{train}|$ which allows for its training over larger constraint sets. In practice, *RankNet* has shown great scalability, reporting successful results over web data from a commercial search engine [Burges et al., 2005].

Although the losses of *Ranking SVM* and *RankNet* are different, both cost functions are related to margin maximization and ensure good generalization properties [Rosset et al., 2004]. *LinkLearn* builds upon these two approaches and proposes to minimize the pairwise hinge loss, like *Ranking SVM*, while relying on gradient optimization, like *RankNet*. This choice yields a further efficiency advantage compared to *RankNet* for training. In particular, the gradient of the hinge loss $Ł^{\mathrm{LinkLearn}}$ is especially inexpensive to compute since the constraints verifying

$$1 - \mathrm{sim}_{\mathbf{w}}(d, l^+) + \mathrm{sim}_{\mathbf{w}}(d, l^-) < 0$$

yields null gradients, i.e.

$$\frac{\partial}{\partial \mathbf{w}} \max(0, 1 - \mathrm{sim}_{\mathbf{w}}(d, l^+) + \mathrm{sim}_{\mathbf{w}}(d, l^-)) = 0.$$

Hence, after a few iterations, the hinge loss yields a null gradient for most constraints. This contrasts with *RankNet* for which each constraint yields a non-zero gradient $\partial \mathrm{L}_{a,b,c}^{\mathrm{CE}} / \partial \mathbf{w}$, which needs to be computed through back-propagation [LeCun et al., 1998b].

3.3 Experiments and Results

This section presents two sets of experiments assessing *LinkLearn*. First, *LinkLearn* is trained over a subset of the hyperlinked *Wikipedia* corpus [Wikipedia], and is then tested over a disjoint subset of the same corpus. Second, the model learned over *Wikipedia* is applied to a retrieval dataset from the TREC benchmark [Voorhees, 2000]. In both cases, we compared the learned term weighting to *Okapi* weighting [Robertson et al., 1994].

	train	valid	test
Number of documents	150,625	150,625	150,625
Vocabulary size	229,003	227,018	230,198
Avg. number of terms per doc.	83.3	83.5	83.4
Avg. number of links per doc.	13.4	12.5	12.6

Table 3.1. The 3 subsets of the *Wikipedia* corpus (vocabulary size and number of terms per document are measured after stopping and stemming).

3.3.1 Wikipedia Experiments

In the following, we first present the experimental setup for the *Wikipedia* experiments and then describe the results.

Experimental Setup

Wikipedia consists of $\sim 450,000$ encyclopedia articles. In each article, the authors refer to other related articles using hyperlinks. In order to evaluate the generalization properties of our model, the corpus is randomly split into three different parts, D_{train}, D_{valid} and D_{test}. This split results in three sets composed of $150,625$ documents. For all documents, the links pointing to articles in a different set are removed, so that each set can be used individually. Table 3.1 summarizes set statistics. As a preprocessing step, all three sets are stopped (i.e. common terms occurring in more than $10,000$ documents are removed) and stemmed (i.e. each word is replaced with its stem, e.g. connection, connected are replaced with connect, using Porter's algorithm [Porter, 1980]). Moreover, terms occurring only once in the corpus are also removed since this greatly reduces the vocabulary size without any impact on document comparisons.

The D_{train} and D_{valid} sets are used during model training: D_{train} is used for gradient descent (i.e. $\mathrm{L}^{\mathrm{LinkLearn}}$ is minimized over this set) while D_{valid} is used to select the model hyperparameters. In contrast, D_{test} is considered unavailable during training and is used solely for the final evaluation. This evaluation relies on the task of *Related Documents Search*, considering each document as a query for which the IR system should retrieve documents presenting similar topics. This task corresponds to the "related pages" function that exists in web search engines, such as Google. In our case, we assume that the documents relevant to a given query documents d are the documents

	Okapi	*LinkLearn*
Precision at top 10	21.5%	25.2% (+18%)
Break Even Point	36.6%	42.1% (+15%)
Average Precision	37.3%	43.8% (+17%)

Table 3.2. Results for the Related Document Search Task (*Wikipedia* D_{test} set), relative improvements are reported in brackets.

sharing a link with d. These relevance assessments are certainly correct since the author of an encyclopedia article uses the links to point at related articles. They are also certainly incomplete, since the author of an article may not refer to all related articles. This weakness is however not specific to our labeling hypothesis, since *system pooling*, the most common way to label datasets for relevance used at TREC [Voorhees, 2006], also underestimate the relevance set [Baeza-Yates and Ribeiro-Neto, 1999].

Hence, for each document $d \in D_{test}$, we perform a retrieval ranking that should ideally order the documents linked to d on the first positions. The results of this task are evaluated to the three standard retrieval measures, P10, BEP and AvgP, introduced in Chapter 2. The performance of our learned weighting strategy is then compared with the performance of *Okapi* over the same task, see Section 3.2. Like for *LinkLearn*, *Okapi* hyperparameters have been selected over the D_{valid} part of the corpus.

Results

Table 3.2 presents the results over the D_{test} part of *Wikipedia*. According to all measures, *LinkLearn* outperforms *Okapi*. In all cases, the relative improvement is more than 15%. We further compare the results of *LinkLearn* and *Okapi* for each of the 150,625 query documents in order to verify whether the advantage of *LinkLearn* could be due to a few queries. For that purpose, we apply the Wilcoxon signed rank test [Rice, 1995] to check whether the sets of 150,625 query results from *LinkLearn* and *Okapi* could have been drawn from two distributions having the same median. The test rejects this hypothesis with 95% confidence for each retrieval measure, meaning that the observed improvement is consistent over the query set.

Hence, these experiments show that a measure of document similarity inferred from hyperlinked training data can be effective on a Related Document Search task. This also underscores the fact that hyperlinks convey valuable

N. of documents	24,823	
N. of terms per doc.	63.8	
Vocabulary size	45,188	
Query Set	Dev.	Eval.
Query Set Id.	TR-8	TR-9
N. of queries	50	50
Avg. n. of terms per q.	7.0	6.0
Avg. n. of rel. doc. per q.	35.1	41.0

Table 3.3. The TREC-9/TDT-2 Dataset (vocabulary size and number of terms per document are measured after stopping and stemming).

information about the semantic proximity of documents from which the *LinkLearn* model benefits. However, the Related Document Search experiments are only a first evaluation of our model, and its assessment over ad-hoc retrieval experiments is of greater interest as both *LinkLearn* and *Okapi* primarily target such tasks.

3.3.2 TREC Experiments

In the following, we first present the experimental setup for the TREC experiments and then describe the results.

3.3.3 Experimental Setup

The Text REtrieval Conference (TREC) defines different benchmark corpora to assess IR system performance [Harman, 1993]. The data provided consists of documents, queries and their corresponding relevance assessments. For each document set, there are generally two sets of ~ 50 queries: the development set is intended for parameter tuning and the evaluation set is used to measure the generalization performance.

In the following, we presents the results over the dataset referred as TREC-9/TDT-2 [Voorhees, 2000]. This set consists of $\sim 25,000$ news documents, with a vocabulary of $\sim 45,000$ words. Table 3.3 summarizes the set statistics. Compared to the *Wikipedia* Related Document Search, the TREC data represents a more challenging task due to the brevity of its queries (~ 6.5 words per query). Our experiment over TREC data applies the *LinkLearn* model learned over *Wikipedia* to this task, without re-training or adaptation. This experiment

	Okapi	*LinkLearn*
Precision at top 10	38.8%	43.2% (+11%)
Break Even Point	30.3%	32.4% (+7%)
Average Precision	29.3%	30.6% (+4%)

Table 3.4. Results for the Retrieval task with TREC9 queries for TDT-2 corpus, relative improvements are reported in brackets.

aims at showing whether it is possible to benefit from the proximity information conveyed by a hyperlinked corpus, even when targeting an application relying on different type of data. Like for the previous experiments, we rely on *Okapi* as the baseline term weighting approach. The development queries of TREC-9 have been used to select the hyperparameters of *Okapi*.

3.3.4 Results

Table 3.4 presents the results over the TREC data. This table agrees with the results obtained over Wikipedia: *LinkLearn* yields to an improvement with respect to *Okapi* according to the different performance measures (e.g. 11% improvement for P10, 43.3% vs 38.8%). This shows that the similarity measure derived from one training corpus (i.e. Wikipedia, a set of encyclopedia articles) can be applied over a different corpus (i.e. TDT-2, a set of broadcast news transcripts and newswire articles). This means that a retrieval task over a given corpus, which may not even be hyperlinked, can benefit from large hypertext datasets that can easily be found on the web nowadays.

As a final analysis of our approach, we compare the inferred *LinkLearn* weighting with *Okapi*. Figure 3.1 shows the weight d_i of a term i in a document d with respect to $tf_{d,i}$, df_i and l_d. For each plot, we vary only one of these factors, keeping the other factors to their mean value. Moreover, the *Okapi* value has been multiplied by a constant, so that both weighting scheme have the same average. This scaling has no effect on *Okapi* results but helps reading the plots.

These plots show that the inferred weighting (*LinkLearn*) and the one given *a-priori* (*Okapi*) are close to each other which may highlight the appropriateness of *Okapi*'s parameterization. However, the plots also highlight some differences between the weighting schemes that could explain why *LinkLearn* yields higher performance. The main differences are summarized below:

Term Frequency tf For low tf values ($tf < 25$), the term weight grows much

slower when tf increases for *LinkLearn* than for *Okapi* which means that term repetitions (high tf) are considered more important in *Okapi* than in *LinkLearn*.

Document Frequency df The term weights of the two approaches are similar for low df but different for large df. In this case, *Okapi* gives more weight to terms occuring in many documents, which means that general terms have more influence on *Okapi* matching than on *LinkLearn* similarity.

Document Length l The two weighting schemes are similar for short documents, while *LinkLearn* gives less weight to longer documents, which may mean that long documents may contain a large amount of repetitions about the same topic rather than being of richer content.

This analysis however only shows a partial picture since the most interesting properties of *LinkLearn* certainly lie in the dependencies between the different variables. Also, this analysis should be taken carefully, since even a slight variation not appearing on these plots may have a large impact on document ranking.

3.4 Conclusions

This chapter has introduced *LinkLearn*, an approach to derive a document similarity measure from hyperlink information. As stated in previous work [Brin and Page, 1998; Davison, 2000], links between documents can be considered as an indicator of topical relatedness and a hyperlinked training corpus provides valuable information to identify a reliable measure of similarity between documents. The proposed approach could benefit text mining applications, like IR or document clustering [Baeza-Yates and Ribeiro-Neto, 1999] in which the assessment of semantic similarity is a crucial point.

The proposed model infers a text similarity measure by learning the term weighting function in the context of the vector space model. This function is learned such that the resulting similarity $\text{sim}_{\mathbf{w}}$ considers linked documents closer than unlinked ones. To achieve this goal, a learning objective inspired by ranking SVM [Joachims, 2002] is optimized through gradient descent.

The proposed approach has been compared to the state-of-the-art similarity measure used in *Okapi* systems [Robertson et al., 1994] over two different tasks. In both cases, *LinkLearn* is trained over a subset of *Wikipedia*. Then, the first task aims at finding the articles related to a given article in another subset of

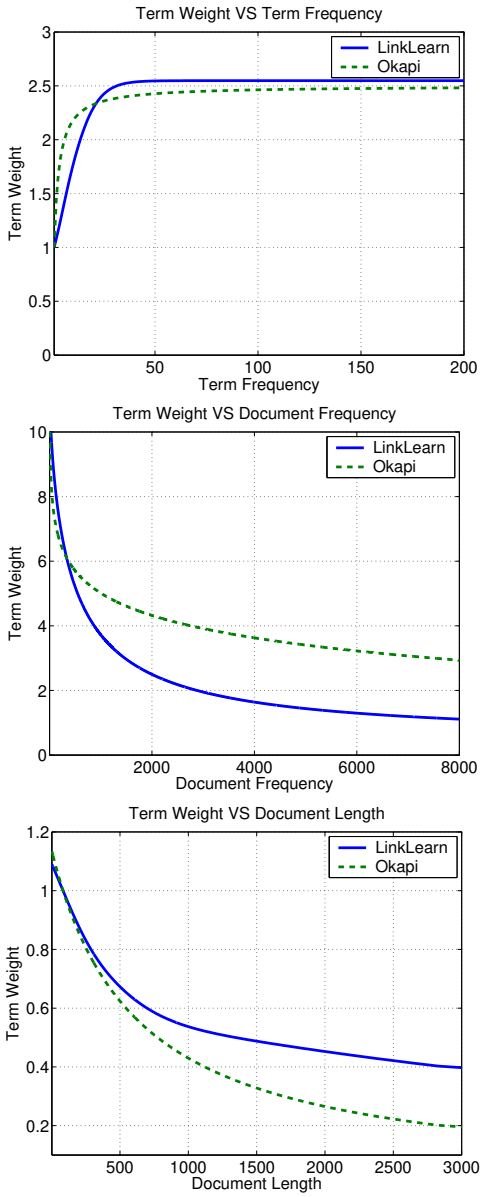

Figure 3.1. Term Weights for *Okapi* and *LinkLearn*

Wikipedia, while the second task is an ad-hoc retrieval task from the TREC benchmark [Voorhees, 2000]. Compared to *Okapi*, *LinkLearn* brings a significant improvement in both cases. The results over TREC data [TREC] are of special interest since they highlight that our learning model can outperform state-of-the-art term weighting, even when learning and testing are performed

over very different datasets (Wikipedia articles vs. newswire data).

Besides this transfer learning aspect, *LinkLearn* also opens several possibilities regarding the learning of term weighting functions. With a data-driven strategy, it is possible to consider functions that depend on more features than handcrafted approaches. For instance, one can exploit the document structure, using as separate features the number of occurrences in the title, in the section headings, etc. Moreover, one could consider modeling term correlation by computing features from term co-occurrence statistics or from bi-gram statistics.

The next chapter is dedicated to a different IR problem, the retrieval of images from text queries. The proposed approach builds upon the online optimization strategy of *LinkLearn*, and proposes an efficient learning procedure to this ranking problem.

3.5 Contributions

The *LinkLearn* model has been presented at an IR conference [Grangier and Bengio, 2005a] and at a machine learning workshop on learning to rank [Grangier and Bengio, 2005b].

◇ D. Grangier and S. Bengio. Inferring document similarity from hyperlinks. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 359–360, Bremen, Germany, November 2005a.

◇ D. Grangier and S. Bengio. Exploiting hyperlinks to learn a retrieval model. In *NIPS Workshop on Learning to Rank*, pages 12–17, Whistler, Canada, December 2005b.

4 Retrieving Images from Text Queries

In this chapter, we address the problem of retrieving pictures from text queries. In this task, the retrieval system is given a set of pictures and a few word query, it then outputs a picture ranking in which the pictures relevant to the query should appear above the others. This type of setup is common in several application domains, including web search engines, news wire services or stock photography providers. So far, the most widely-used approach to this problem consists in applying text retrieval techniques over a set of manually-produced captions that describe each picture. Although effective, this solution is expensive, as it requires a significant manual labeling effort.

Consequently, several automatic annotation approaches have been proposed in the literature. These approaches rely on a set of captioned pictures to learn a model, which can then predict textual annotations for any unlabeled picture. Two main types of auto-annotation models have been introduced: *concept classification models* and *bi-modal generative models*. In the case of concept classification, a classifier is learned for each vocabulary term, or concept, t. This classifier takes as input a picture and outputs a confidence value indicating whether the term t should occur in the predicted picture caption. This classification problem is typically addressed using Support Vector Machine (SVM) [Naphade, 2004; Vogel and Schiele, 2004] or boosting classifiers [Tieu and Viola, 2004], as these large margin approaches enjoy good generalization properties [Vapnik, 1995]. In the case of bi-modal generative models, the training procedure learns a distribution estimating the joint probability $P(p, c)$ of a picture p (i.e. a set of visual features) and a caption c (i.e. a set of terms describing the picture). Given a test picture p, the learned distribution can then be used to infer the most likely caption, or a distribution over the whole vocabulary. Compared to concept classification, this generative approach hence

learns a single model for all vocabulary terms, which notably yields a better modeling of term dependencies. Several bi-modal generative models have been proposed in the recent years [Jeon et al., 2003; Barnard et al., 2003; Monay and Gatica-Perez, 2004], each model relying on different conditional independence assumptions between the observation of the text and the visual features.

Besides their differences, both concept classification and bi-modal generative models address the image retrieval problem through an intermediate task, auto-annotation. Image retrieval is performed by applying text retrieval techniques over the textual outputs of the auto-annotation model. Therefore, their learning procedure does not maximize a criterion related to the final *retrieval* performance, instead it maximizes a criterion related to the *annotation* performance. In this chapter, we adopt an alternative approach and introduce a model to learn an image retrieval model directly, without relying on auto-annotation. Our approach builds upon recent work on discriminative learning for text retrieval, such as Ranking SVM [Joachims, 2002], RankNet [Burges et al., 2005] and *LinkLean*, described in Chapter 3. The proposed model, Passive-Aggressive Model for Image Retrieval (PAMIR), adopts a learning criterion related to the final retrieval performance. Its learning procedure takes as input a set of training queries, as well as a set of pictures, and outputs a trained model likely to achieve high ranking performance on new data. Moreover, PAMIR also enjoys an efficient learning algorithm derived from recent advances in online learning of kernel-based classifiers [Crammer et al., 2006]. The advantages of the proposed approach are several: our model parameterization can benefit from effective kernels for pictures comparison, while its optimization procedure permits an efficient learning over large training sets. Furthermore, our ranking criterion yields a discriminative retrieval model that does not rely on an intermediate annotation task, which is theoretically appealing [Vapnik, 1995]. These advantages are actually supported by our experiments, in which PAMIR is shown to outperform various state-of-the-art alternatives. For instance, the precision at top 10 of PAMIR reaches 10% over the Corel dataset [Duygulu et al., 2002], which should be compared to 9.3% for SVM for concept classification, the best alternative (see Section 4.4).

The remainder of this chapter is organized as follows. Section 4.1 briefly describes previous related research. Section 4.2 introduces the proposed approach. Section 4.3 presents the features used for image and query representation. This section also describes different picture kernels from which PAMIR could benefit. Section 4.4 reports the experiments comparing PAMIR to the alternative approaches. Finally, Section 4.5 draws some conclusions.

4.1 Related Work

With the advent of the digital photography era, image retrieval has increasingly received attention. This study focuses on an important part of this research domain, the *query-by-text* task. This task aims at identifying the pictures relevant to a few word query, within a large picture collection. Solving such a problem is of particular interest from a user perspective since most people are used to efficiently access large textual corpora through text querying and would like to benefit from a similar interface to search collections of pictures. In this section, we briefly describe prior work focusing on this task.

So far, the *query-by-text* problem has mainly been addressed through automatic annotation approaches. In this case, the objective is to learn a model that can predict textual annotations from a picture. Such a model permits the retrieval of unlabeled pictures through the application of text retrieval techniques over the auto-annotator outputs. In the following, we briefly describe the two main types of approaches adopted in this context, *concept classification* and *bi-modal generative models*.

4.1.1 Concept Classification

Concept classification formulates auto-annotation within a classification framework. Each vocabulary term t, also referred as a *concept*, defines a binary classification problem, whose positive examples are the pictures for which the term t should appear in the predicted annotation. In this case, the learning procedure hence consists in training a binary classifier for each vocabulary term, and each classifier is learned to minimize the error rate of its concept classification problem.

Efforts in concept classification started with the detection of simple concepts such as indoor/outdoor [Szummer and Picard, 1998], or landscape/cityscape [Vailaya et al., 1998]. Then, significant research has been directed towards detecting more challenging concepts, notably in the context of the TREC video benchmark [Smeaton et al., 2006]. Large sets of various concepts have then been addressed in recent work, such as [Carneiro and Vasconcelos, 2005; Chang et al., 2006]. Nowadays, popular approaches in concept classification mainly relies on large margin classifiers, such as Support Vector Machines (SVM) [Amir et al., 2005; Naphade, 2004; Vogel and Schiele, 2004] or boosting approaches [Tieu and Viola, 2004]. SVM for concept classification constitutes the state-of-the-art for single word queries. In this application scenario, the images of the test corpus are ranked according to the confidence scores outputted by the classifier corre-

sponding to the query term [Naphade, 2004; Vogel and Schiele, 2004]. However, in the case of multiple word queries, concept classifiers are more difficult to apply since the independent training of each concept classifier requires to further define fusion rules to combine the scores of the different concept classifiers [Amir et al., 2005; Chang et al., 2006]. [Amir et al., 2005] compares different fusion strategies and concludes that, for query-by-text tasks, it is generally effective to compute the average of the score of the concept classifiers corresponding to the query terms, after having normalized their mean and variance. Therefore, we will adopt this fusion procedure latter in our experiments. As an alternative to such ad-hoc fusion strategies, bi-modal generative approaches have been introduced to learn a single model over the whole vocabulary, yielding a solution which can natively handle multiple-word queries.

4.1.2 Bi-Modal Generative Models

Contrary to concept classification, bi-modal generative approaches do not consider the different vocabulary words in isolation. Instead, these approaches model the joint distribution $P(c, p)$ of the textual caption (c) and the picture visual features (p), $P(c, p)$. The parameters of such a distribution are typically learned through maximum likelihood training, relying on a set of picture/caption pairs. After this learning phase, the retrieval of unlabeled pictures can be performed by ranking the pictures according to their likelihood $P(p|q)$ given query q, which is derived from the joint $P(q, p)$ through Bayes rule. Alternatively, it is also possible to estimate a conditional multinomial over the vocabulary $\{P(t|p), \forall t \in V\}$, for each unlabeled picture. This enables to retrieve pictures through the application of text retrieval techniques over the inferred multinomials. In this case, each multinomial $P(\cdot|p)$ is considered to represent a textual item, in which the number of occurrences of term t is proportional to $P(t|p)$. This alternative retrieval technique is generally preferred since it is more efficient (the multinomials need to be inferred only once for all queries) and it has shown to be more effective [Monay and Gatica-Perez, 2004].

Several approaches based on the bi-modal generative framework have been proposed in the recent years. These models mainly differ in the types of distributions chosen to model textual and visual features, as well as in the way they model the dependencies between both modalities. In the following, we have chosen to briefly describe three such models, Cross-Media Relevance Model (CMRM) [Jeon et al., 2003], Cross-Media Translation Table (CMTT) [Pan et al., 2004] and Probabilistic Latent Semantic Analysis (PLSA) [Monay and Gatica-Perez, 2004]. A longer survey could also have described alternative mod-

els such as Multimodal Hierarchical Aspect Model [Barnard and Forsyth, 2001; Barnard et al., 2003], Multiple Bernoulli Relevance Model [Feng and R. Manmatha, 2004] or Latent Dirichlet Allocation [Blei et al., 2003]. However, we decided to focus on models that have shown to be the most effective over the Corel dataset [Duygulu et al., 2002].

Cross Media Relevance Model (CMRM) [Jeon et al., 2003], is inspired by Cross-Lingual Relevance Model [Lavrenko et al., 2002], considering caption of an image as the translation of its visual properties into words. In this model, it is assumed that the visual properties of an image are summarized as a set of discrete visual features. Formally, the visual features of a picture p are represented as a vector,

$$p = (vtf_{1,p}, \ldots, vtf_{V,p}),$$

where $vtf_{i,p}$ refers to the number of features of type i in picture p and V is the total number of visual feature types.

Given such a representation, CMRM infers a multinomial $P(t|p^{test})$ over the vocabulary for any test picture p^{test}. For that purpose, the joint probability of term t and all the visual elements of p^{test} is estimated by its expectation over the training pictures in P_{train},

$$P(t, p^{test}) = \sum_{j=1}^{|P_{train}|} P(j) \cdot P(t, p^{test}|j).$$

It is then assumed that terms and visual elements are independent given a training picture, leading to

$$P(t, p^{test}) = \sum_{j=1}^{|P_{train}|} P(j) \cdot P(t|j) \prod_{v=1}^{V} P(v|j)^{vtf_{v,p^{test}}}. \tag{4.1}$$

In this equation, the probability of a training picture $P(j)$ is assumed to be uniform over the training set, i.e. $P(j) = 1/|P_{train}|$, while the probability of a term given a training picture $P(t|j)$ and the probability of a visual element given a training pictures $P(v|j)$ are estimated through maximum likelihood, smoothed with the *Jelinek-Mercer* method [Jeon et al., 2003]. From (4.1), $P(t|p^{test})$ can then be estimated through Bayes rule, $P(t|p^{test}) = P(t, p^{test})/P(p^{test})$. Although simple, this approach has shown to be more effective compared to other approaches inspired by translation models, e.g. [Duygulu et al., 2002].

Cross Media Translation Table (CMTT) also builds upon cross-lingual retrieval techniques [Pan et al., 2004]. This model considers textual terms and discrete visual features, or *visterms*, as words originating from two different

languages and constructs a translation table containing $P(t|v)$ for any pair of term/visterm (t, v). This table allows for the estimation of $P(t|p^{test})$ for any term t and any picture p^{test}:

$$P(t|p^{test}) = \sum_{i=1}^{m} P(t|v_i)P(v_i|p^{test}),$$

where $P(v_i|p^{test}) = \frac{vtf_{i,p^{test}}}{\sum_{i=1}^{m} vtf_{i,p^{test}}}$, and $v_1, \ldots, v_m$ are the visterms of p^{test}.

The translation table $\{P(t|v), \forall t, v\}$ is built from the training data D_{train} according to the following process. First, each term i (and each visterm j) is represented by a $|D_{train}|$ dimensional vector, t_i (v_j), in which each component k is the weight of term i (visterm j) in the k^{th} training example. As a noise removal step, the matrix $M = [t_1, \ldots, t_T, v_1, \ldots, v_V]$ containing all term and visterm vectors is approximated with a lower rank matrix, $M' = [t'_1, \ldots, t'_T, v'_1, \ldots, v'_V]$, through Singular Value Decomposition, and $P(j|i)$ is finally defined as

$$P(j|i) = \frac{\cos(t'_i, v'_j)}{\sum_{k=1}^{|V|} \cos(t'_i, v'_k)}.$$

Like CMRM, this method has also been evaluated over the *Corel* corpus [Pan et al., 2004], where it has shown to be effective. The use of Singular Value Decomposition has notably shown to improve noise robustness. However, CMTT has also some limitations. In particular, cosine similarity can only model simple relationships between terms and visual features. Approaches modeling more complex relationships, such as Probabilistic Latent Semantic Analysis [Monay and Gatica-Perez, 2004], have subsequently been introduced.

Probabilistic Latent Semantic Analysis (PLSA) has first been introduced for text retrieval [Hofmann, 2001], as explained in Chapter 2. It has then been modified for image retrieval [Monay and Gatica-Perez, 2004]. This model introduces the following conditional independence assumption: "terms and discrete visual features are independent from pictures conditionally to an unobserved discrete variable, the *aspect* variable $z_k \in \{z_1, \ldots, z_K\}$". In this framework, the probability of observing a term t or a visual feature v in a picture p follows

$$P(p, t) = P(p) \cdot \sum_k P(z_k|p)P(t|z_k), \tag{4.2}$$

$$P(p, v) = P(p) \cdot \sum_k P(z_k|p)P(v|z_k). \tag{4.3}$$

The different parameters of the model can be estimated relying on a two-step process. First, the probabilities $P(p), P(z_k|p)$ and $P(t|z_k)$ for all $p \in P_{train}$

are estimated to maximize the likelihood of the training captions through the Expectation Maximization algorithm. Then the probabilities $P(v|z_k), \forall v, k$ are fitted to maximize the likelihood of the training pictures, keeping $P(p)$ and $P(z_k|p)$ fixed. For a test picture without caption, the probabilities $P(p), P(z_k|p)$ are estimated to maximize the likelihood of the test picture, keeping $P(v|z_k), \forall (v, k)$ to the values estimated during training. After this procedure, (4.2) is applied to infer $P(p, t)$ for any test picture p and any term t. Similarly to CMRM, Bayes rule can then derive $P(t|p)$ from $P(p, t)$.

PLSA has several strengths: the latent aspect assumption allows one to model more complex dependencies between term and visual features, compared to CMRM or CMTT. Moreover, the two step training procedure biases the latent space toward the text modality, yielding better performance than less constrained latent models [Monay and Gatica-Perez, 2004].

In absence of manual annotations, bi-modal generative models constitute the state-of-the-art for the retrieval of images from multiple-word queries, while, as mentioned above, concept classification is generally preferred for single word queries. However, one could wonder whether it is possible to provide a single solution for both settings. More fundamentally, one can also question the auto-annotation framework on which both types of approaches are based. In both cases, model training aims at solving an auto-annotation problem: for concept classification, the learning objective is to minimize the number of false positives (predicting a word which does not occur in the reference annotation) and false negatives (not predicting a word occurring in the reference annotation), while, for bi-modal generative models, the learning objective is to maximize the likelihood of the training picture/caption pairs. None of these criteria is tightly related to the final retrieval performance and there is hence no guarantee that a model optimizing such annotation objectives also yields good retrieval rankings.

In order to address those issues, we propose a *discriminative ranking model* for the query-by-text problem. The proposed approach is based on recent work on discriminative learning for the retrieval of text documents, such as [Burges et al., 2005; Joachims, 2002] or our own work presented in previous chapter. It learns a retrieval model with a criterion related to the ranking performance over a set of training queries. To the best of our knowledge, this is the first attempt to address the query-by-text problem directly, without solving an intermediate annotation problem.

4.2 Passive-Aggressive Model for Image Retrieval

This section introduces our discriminative model for the retrieval of images from text queries, *Passive-Aggressive Model for Image Retrieval* (PAMIR). It first formalizes the query-by-text problem before introducing PAMIR parameterization and learning objective. Finally, it explains how the proposed linear model can be applied to infer non-linear decision functions relying on kernels.

4.2.1 Formalizing the Query-by-Text Problem

In the query-by-text problem, the retrieval system receives a text query q, from the text space $\mathcal{T}$, and a set of pictures P, from the picture space $\mathcal{P}$. It should then output a picture ranking in which the pictures relevant to q would ideally appear above the others, i.e.

$$\forall p^+ \in R(q,P), \forall p^- \in \overline{R}(q,P),\ rk(q,p^+) < rk(q,p^-) \tag{4.4}$$

where $R(q,P)$ refers to the set of pictures of P that are relevant to q, $\overline{R}(q,P)$ refers to the set of pictures of P that are not relevant to q and $rk(q,p)$ refers to the position of picture p in the ranking outputted for query q. Our goal is hence to learn a ranking model from training pictures P_{train} and queries Q_{train} such that the constraints of type (4.4) are likely to be verified over new pictures P_{test} and queries Q_{test}.

Based on the framework introduced in Chapter 2, we address this ranking problem with a ranking function f. This function $f : \mathcal{T} \times \mathcal{P} \to \mathbb{R}$ assigns a real value $f(q,p)$ expressing the match between any query q and any picture p. Given a query q, f computes the picture scores $\{f(q,p), \forall p \in P\}$ and the pictures are ordered by decreasing scores. Hence, (4.4) translates to

$$\forall p^+ \in R(q,P), \forall p^- \in \overline{R}(q,P), f(q,p^+) > f(q,p^-), \tag{4.5}$$

and our learning task aims at identifying a function f likely to verify (4.5) for unseen pictures P_{test} and queries Q_{test}. For that purpose, we introduce a parametric function $f_{\mathbf{w}}$ along with an algorithm to infer the parameter $\mathbf{w}$ from (P_{train}, Q_{train}), so that $f_{\mathbf{w}}$ is likely to achieve this objective.

4.2.2 Model Parameterization

The parameterization of $f_{\mathbf{w}}$ is inspired from text retrieval (see Chapter 2),

$$f_{\mathbf{w}} : \mathcal{T} \times \mathcal{P} \to \mathbb{R}, \quad \text{where} \quad f_{\mathbf{w}}(q,p) = q \cdot g_{\mathbf{w}}(p),$$

$g_{\mathbf{w}}$ refers to a parametric mapping from the picture space $\mathcal{P}$ to the text space $\mathcal{T}$, and $\cdot$ refers to the dot product in the text space, which is commonly applied to measure the matching between textual vectors, as explained in the previous chapters. In other words, our scoring function $f_{\mathbf{w}}$ measures the match between a picture p and a query q by first projecting the picture into the text space according to $g_{\mathbf{w}}$, before measuring the dot-product between the obtained textual vector $g_{\mathbf{w}}(p)$ and the query q.

In the following, the form of $g_{\mathbf{w}}$ is first limited to linear mappings,

$$g_{\mathbf{w}} : \mathcal{P} \to \mathcal{T}, \quad \text{where} \quad g_{\mathbf{w}}(p) = (w_1 \cdot p, \ldots, w_T \cdot p) \tag{4.6}$$

and $\mathbf{w} = (w_1, \ldots, w_T)$ is a vector of $\mathcal{P}^T$, T being the dimension of the text space $\mathcal{T}$. Section 4.2.5 then shows that the training procedure proposed thereafter can be extended to non-linear mappings through the kernel trick [Shawe-Taylor and Cristianini, 2000].

4.2.3 Large Margin Learning for our Ranking Problem

Our goal is to learn the parameter $\mathbf{w}$ such that $f_{\mathbf{w}}$ yields high ranking performance over unseen test queries. For that purpose, we first introduce a geometric interpretation of $f_{\mathbf{w}}$, from which we can derive a margin maximization objective suitable to our ranking task.

For any query $q = (q_1, \ldots, q_T) \in \mathcal{T}$ and picture $p \in \mathcal{P}$, we define $\gamma(q, p)$ as the vector $(q_1 p, \ldots, q_T p)$ of $\mathcal{P}^T$ and rewrite $f_{\mathbf{w}}(q, p)$ as $\mathbf{w} \cdot \gamma(q, p)$, since

$$\begin{aligned} f_{\mathbf{w}}(q, p) &= q \cdot g_{\mathbf{w}}(p) = q \cdot (w_1 \cdot p, \ldots, w_T \cdot p) \\ &= \sum_{t=1}^{T} w_t \cdot (q_t p) = \mathbf{w} \cdot \gamma(q, p). \end{aligned}$$

Hence, we can interpret $f_{\mathbf{w}}(q, p)$ as the projection of $\gamma(q, p)$ onto the vector $\mathbf{w}$. This means that PAMIR ranks the pictures of P according to the order of the projections of $\{\gamma(q, p), \forall p \in P\}$ along the direction of $\mathbf{w}$, see Figure 4.1. With such an interpretation, one can easily notice that only the direction of $\mathbf{w}$ determines whether the constraints of type (4.5), $\forall q \in \mathcal{T},\ \forall p^+ \in R(q, P)$,

$$\forall p^- \in \overline{R}(q, P),\ \mathbf{w} \cdot \gamma(q, p^+) - \mathbf{w} \cdot \gamma(q, p^-) > 0,$$

are verified since the norm of $\mathbf{w}$ has no influence on the sign of $\mathbf{w} \cdot \gamma(q, p^+) - \mathbf{w} \cdot \gamma(q, p^-)$.

Hence, we can arbitrarily constrain the weight vector to lie on the unit circle $\mathcal{U}$, and solve our learning problem by finding a vector $u \in \mathcal{U}$ that verifies all

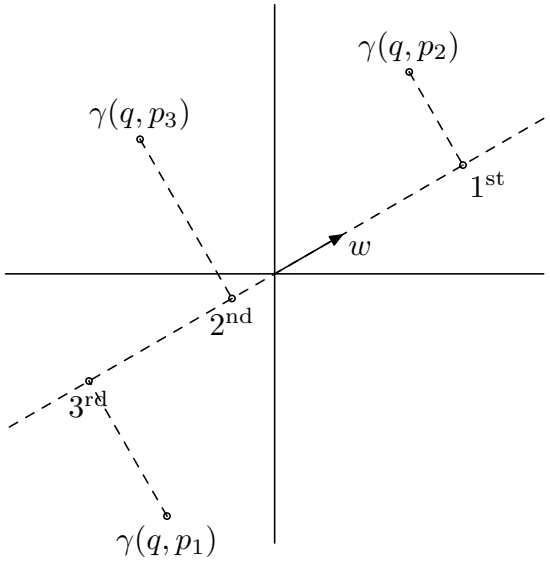

Figure 4.1. PAMIR ranking strategy: in this example, the pictures of $\{p_1, p_2, p_3\}$ are ranked p_2, p_3, p_1 in answer to the query q. This figure illustrates that the pictures are ranked according to the order of the projections of $\{\gamma(q,p_1), \gamma(q,p_2), \gamma(q,p_3)\}$ along the direction of $\mathbf{w}$.

training constraints. In other words, we want to select the weight vector in the set

$$\mathcal{S} = \{u \in \mathcal{U} \quad \text{s.t.} \quad \forall (q, p^+, p^-) \in D_{train}, \quad u \cdot \gamma(q,p^+) - u \cdot \gamma(q,p^-) > 0\}$$

where D_{train} refers to all triplets (q, p^+, p^-) such that $q \in Q_{train}$, $p^+ \in R(q, P_{train})$, $p^- \in \overline{R}(q, P_{train})$.

When the training constraints are feasible ($\mathcal{S} \neq \emptyset$), any weight vector of $\mathcal{S}$ yields perfect retrieval performance over the training set. However, not all these solutions will yield the same results over some new test data. In order to select a vector of $\mathcal{S}$ likely to yield high generalization performance, we introduce the notion of *margin* for our ranking problem. For any vector $u \in \mathcal{S}$, we define its margin as

$$m(u) = \min_{(q,p^+,p^-) \in D_{train}} u \cdot \gamma(q,p^+) - u \cdot \gamma(q,p^-),$$

which is, by definition of $\mathcal{S}$, a positive quantity. This notion of margin is inspired from the definition introduced in [Herbrich et al., 2000] in the context of ranked categorization.

Equipped with this definition, we now explain why large margin solutions are preferable to ensure good generalization performance. Given a test triplet $(q_{test}, p^+_{test}, p^-_{test})$ composed of a query q_{test}, a picture p^+_{test} relevant to q_{test} and a picture p^-_{test} non-relevant to q_{test}, we define $R(q_{test}, p^+_{test}, p^-_{test})$ as the smallest quantity that satisfies

$$\exists (q_{train}, p^+_{train}, p^-_{train}) \in D_{train}$$
$$\text{s.t.} \begin{cases} \|\gamma(q_{train}, p^+_{train}) - \gamma(q_{test}, p^+_{test})\| < R(q_{test}, p^+_{test}, p^-_{test}) \\ \|\gamma(q_{train}, p^-_{train}) - \gamma(q_{test}, p^-_{test})\| < R(q_{test}, p^+_{test}, p^-_{test}). \end{cases}$$

This definition implies that,

$$\forall u \in \mathcal{S}, \begin{cases} |u \cdot \gamma(q_{train}, p^+_{train}) - u \cdot \gamma(q_{test}, p^+_{test})| < R(q_{test}, p^+_{test}, p^-_{test}) \\ |u \cdot \gamma(q_{train}, p^-_{train}) - u \cdot \gamma(q_{test}, p^-_{test})| < R(q_{test}, p^+_{test}, p^-_{test}) \end{cases}$$

since $\|u\| = 1$. Therefore,

$$\begin{aligned} & u \cdot \gamma(q_{test}, p^+_{test}) - u \cdot \gamma(q_{test}, p^-_{test}) \\ = \quad & (u \cdot \gamma(q_{test}, p^+_{test}) - u \cdot \gamma(q_{train}, p^+_{train})) \\ - \quad & (u \cdot \gamma(q_{test}, p^-_{test}) - u \cdot \gamma(q_{train}, p^-_{train})) \\ + \quad & (u \cdot \gamma(q_{train}, p^+_{train}) - u \cdot \gamma(q_{train}, p^-_{train})) \end{aligned}$$

can be bounded as,

$$u \cdot \gamma(q, p^+_{test}) - u \cdot \gamma(q, p^-_{test}) > -2R(q_{test}, p^+_{test}, p^-_{test}) + m(u)$$

since $u \cdot \gamma(q, p^+_{train}) - u \cdot \gamma(q, p^-_{train}) > m(u)$ by definition of $m(u)$. Consequently, any solution $u \in \mathcal{S}$ for which the margin $m(u)$ is greater than $2R(q_{test}, p^+_{test}, p^-_{test})$ satisfies the test constraint $u \cdot \gamma(q, p^+_{test}) - u \cdot \gamma(q, p^-_{test}) > 0$.

Therefore, we decide to focus on the selection of the weight vector of $\mathcal{S}$ with the largest margin, as this weight is the most likely to satisfy all the constraints of a given test set,

$$u^* = \text{argmax}_{u \in \mathcal{S}}\ m(u).$$

This maximization problem is actually equivalent to the following minimization problem,

$$\min_{u \in \mathcal{P}^T} \frac{1}{m(u)^2} \quad \text{s.t.} \quad \begin{cases} \|u\| = 1 \\ \forall (q, p^+, p^-) \in D_{train}, u \cdot \gamma(q, p^+) - u \cdot \gamma(q, p^-) > m(u), \end{cases}$$

and the introduction of the vector $\mathbf{w} = \frac{1}{m(u)} u$ yields the following formulation of the same problem,

$$\min_{w \in \mathcal{P}^T} \|\mathbf{w}\|^2, \quad \text{s.t.} \quad \forall (q, p^+, p^-) \in D_{train}, \mathbf{w} \cdot \gamma(q, p^+) - \mathbf{w} \cdot \gamma(q, p^-) > 1.$$

This formulation of our retrieval problem is similar to the Ranking Support Vector Machine (RSVM) problem [Joachims, 2002] introduced in the context of text retrieval, even if the notion of margin was not formalized as such in the case of RSVM.

Like for RSVM, we need to relax the training constraints for the non-feasible case ($\mathcal{S} = \emptyset$), which yields the following optimization problem,

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{P}^T} \quad & \|\mathbf{w}\|^2 \quad + \quad C \sum_{(q,p^+,p^-)\in D_{train}} \xi_{q,p^+,p^-} \\ \text{s.t.} \quad & \forall (q,p^+,p^-) \in D_{train}, \begin{cases} w\cdot\gamma(q,p^+) - w\cdot\gamma(q,p^-) > 1 - \xi_{q,p^+,p^-} \\ \xi_{q,p^+,p^-} \geq 0, \end{cases} \end{aligned} \tag{4.7}$$

where the hyperparameter C controls the trade-off between maximizing the margin and satisfying all the training constraints. This problem (4.7) can equivalently be written as,

$$\min_{\mathbf{w}\in\mathcal{P}^T} \ \|w\|^2 \ + \ C \ L(\mathbf{w}; D_{train}),$$

where

$$L(\mathbf{w}; D_{train}) = \sum_{(q,p^+,p^-)\in D_{train}} l(\mathbf{w}; q, p^+, p^-) \tag{4.8}$$

and, $\forall (q,p^+,p^-) \in D_{train}$,

$$l(\mathbf{w}; q, p^+, p^-) = \max(0, 1 - \mathbf{w}\cdot\gamma(q,p^+) + \mathbf{w}\cdot\gamma(q,p^-)),$$

see [Collobert and Bengio, 2004].

4.2.4 An Efficient Learning Algorithm

The resolution of problem (4.7) involves a costly optimization procedure, if the RSVM approach is adopted. In fact, state-of-the-art techniques to solve this problem have a time-complexity greater than $O(|D_{train}|^2)$ [Joachims, 1998], where $|D_{train}|$ denotes the number of training constraints. As we would like to handle large constraint sets, we derive an efficient training procedure by adapting the *Passive-Aggressive* (PA) algorithm, originally introduced for classification and regression problems [Crammer et al., 2006]. For our ranking problem, PA should minimize $L(\mathbf{w}; D_{train})$ while keeping $||\mathbf{w}||^2$ small.

For that purpose, the algorithm constructs a sequence of weight vectors $(\mathbf{w}^0, \ldots, \mathbf{w}^n)$ according to the following iterative procedure: the first vector is set to be zero, $\mathbf{w}^0 = 0$ and, at the i^{th} iteration, the weight $\mathbf{w}^i$ is selected according to the i^{th} training example (q^i, p^{i+}, p^{i-}) and the previous weight $\mathbf{w}^{i-1}$,

$$\mathbf{w}^i = \text{argmin}_w \ \frac{1}{2}\|\mathbf{w} - \mathbf{w}^{i-1}\|^2 + c \ l(\mathbf{w}; (q^i, p^{i+}, p^{i-})). \tag{4.9}$$

Hence, at each iteration, we select the weight $\mathbf{w}^i$ as a trade-off between minimizing the loss on the current example $l(\mathbf{w}; (q^i, p^{i+}, p^{i-}))$ and remaining close to the previous weight vector $\mathbf{w}^{i-1}$. The *aggressiveness* parameter c controls this trade-off. Based on [Crammer et al., 2006], it can be shown that the solution of (4.9) is

$$\begin{aligned} & \mathbf{w}^i = \mathbf{w}^{i-1} + \tau_i v^i, \\ \text{where} \quad & \tau_i = \min\left\{c, \frac{l(\mathbf{w}^{i-1}; (q^i, p^{i+}, p^{i-}))}{\|v_i\|^2}\right\} \\ \text{and} \quad & v^i = \gamma(q^i, p^{i+}) - \gamma(q^i, p^{i-}). \end{aligned} \tag{4.10}$$

The hyperparameter c is selected to maximize the performance over some validation data D_{valid}. The number of iterations n is also validated: training is stopped as soon as the validation performance stops improving. This *early stopping* procedure actually allows one to select a good trade-off between satisfying all training constraints (i.e. minimizing the training loss $L(w; D_{train})$) and maximizing the margin (i.e. minimizing $\|w\|^2$). During the training process, it can be shown that, while the training error is decreasing [Crammer et al., 2006], $\|w\|^2$ tends to increase, see Appendix A.1. Hence, the number of iterations n plays a role similar to C in RSVM (4.7), setting the trade-off between margin maximization and training error minimization. The introduced PA algorithm therefore solves our learning problem with a time-complexity growing linearly with the number of iterations n. The observed complexity, reported later in Section 4.4, actually shows that n grows much slower than $|D_{train}|^2$, a lower bound on RSVM time-complexity, enabling PAMIR to address much larger constraint sets.

4.2.5 Non-Linear Extension

Our model parameterization is based on a linear mapping $g_{\mathbf{w}}$ from the picture space $\mathcal{P}$ to the text space $\mathcal{T}$, see Eq. (4.6). This parameterization can be extended to non-linear mappings through the kernel trick, which allows PAMIR to benefit from effective picture kernels recently introduced in the computer vision literature, e.g. [Wallraven and Caputo, 2003; Kondor and Jebara, 2003; Lyu, 2005]. To kernelize PAMIR, we show that its parameterization solely requires the evaluation of dot products between picture vectors. For that purpose, we prove that, in the weight vector $\mathbf{w} = (w_1, \ldots, w_T)$, each subvector $\mathbf{w}_t$, $\forall t$, is a linear combination of training pictures. This then implies that the evaluation of

$$g_{\mathbf{w}}(p) = (w_1 \cdot p, \ldots, w_T \cdot p), \ \forall p \in \mathcal{P},$$

only requires to compute the dot product between p and any training picture. The proof that, for all t, the vector $\mathbf{w}_t$ is a linear combination of training pictures is performed by induction over the iterations of our training procedure: at the first iteration, the property is obviously verified since $\mathbf{w}_t^0 = 0$, then the update preserves the property since, $\mathbf{w}_t^i = \mathbf{w}_t^{i-1} + \tau_i v_t^i$, where v_t^i is itself a linear combination of training pictures, $v_t^i = q_t^i \ (p^{i+} - p^{i-})$, see Eq. (4.10). Hence, at the last iteration n, $\mathbf{w}_t = w_t^n$ verifies the property. This means that we can rewrite $\mathbf{w}_t$ as $w_t = \sum_{j=1}^{|P_{train}|} \alpha_{t,j} p_j$, where $\forall j,\ \alpha_{t,j} \in \mathbb{R}$. Consequently, we can introduce any kernel function $k : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$, and rewrite $g_{\mathbf{w}}$ as,

$$\forall p \in \mathcal{P},\ [g_{\mathbf{w}}(p)]_t = \sum_{j=1}^{|P_{train}|} \alpha_{t,j} k(p_j, p),$$

where $[g_{\mathbf{w}}(p)]_t$ denotes the t^{th} component of the $g_{\mathbf{w}}(p)$ vector. Practically, in this kernelized case, each $\mathbf{w}_t$ is stored as a support set, consisting of pairs $(\alpha_{t,j}, p_j)$. The following section notably discusses different types of kernels suitable for our task.

This section has introduced PAMIR, a model suitable for image retrieval from text queries. This model has several advantages compared to the previous approaches presented in Section 4.1: unlike SVM for concept classification, PAMIR can natively handle multiple-word queries, without requiring any fusion strategy; unlike bi-modal generative models, it relies on margin maximization training and hence enjoys good generalization properties [Vapnik, 1995]. More importantly, unlike both SVM for concept classification and bi-modal generative models, PAMIR training relies on a ranking criterion related to the final retrieval performance of the model. This criterion yields a discriminative retrieval model, which does not learn from textual annotations, but directly from training queries with pictures assessed for relevance.

4.3 Text and Visual Features

This section introduces both the representation of text queries, and the representation of pictures, along with kernel functions suitable for picture comparison.

4.3.1 Query Representation

The *bag-of-words* framework is borrowed from text retrieval for query representation. As in previous chapters, the vocabulary is given prior to training to define the set of allowed words. Then, each query is assigned a vector $q \in \mathbb{R}^T$,

where T denotes the vocabulary size. The i^{th} component of this vector, corresponding to the weight of term i in query q, is defined according to the *normalized idf* weighting scheme [Baeza-Yates and Ribeiro-Neto, 1999],

$$q_i = \frac{b_{i,q}\ idf_i}{\sqrt{\sum_{j=1}^{T}(b_{j,q}\ idf_j)^2}}$$

where $b_{i,q}$ is a binary weight, denoting the presence ($b_{i,q} = 1$) or absence ($b_{i,q} = 0$) of i in q, and idf_i is the inverse document frequency of i. This latter quantity is defined based on a reference corpus, such as an encyclopedia, and corresponds to $idf_i = -\log(r_i)$, where r_i refers to fraction of corpus documents containing term i. This weighting hypothesizes that, among the terms present in q, the terms appearing rarely in the reference corpus are more discriminant and should be assigned higher weights (see previous chapters for a more complete discussion on term weighting).

4.3.2 Picture Representation

The representation of pictures for image retrieval is a research topic in itself, and different approaches have been proposed in the recent years, e.g. [Feng and R. Manmatha, 2004; Takala et al., 2005; Tieu and Viola, 2004]. Contrary to the well-established bag-of-words representation for text data, there is not yet a single image representation that would be adequate for a wide variety of retrieval problems. However, among the proposed representations, a consensus is emerging on using *local descriptors* for various tasks, e.g. [Lowe, 2004; Quelhas et al., 2005]. This type of representation segments the picture into *regions of interest*, and extracts visual features from each region. The segmentation algorithm as well as the region features vary among approaches, but, in all cases, the image is then represented as a set of feature vectors describing the regions of interest. Such a set is often called a *bag-of-local-descriptors*.

This study also adopts the local descriptor framework. Our features are extracted by dividing each picture into overlapping square blocks, and each block is then described with edge and color histograms. For edge histograms, we rely on *uniform Local Binary Patterns* [Ojala et al., 2002]. These texture descriptors have shown to be effective on various tasks in the computer vision literature [Ojala et al., 2002; Takala et al., 2005], certainly due to their robustness with respect to changes in illumination and other photometric transformations [Ojala et al., 2002]. Local Binary Patterns assign the texture histogram of a block by considering differences in intensity at circular neighborhoods centered on each pixel. Precisely, we use $LBP_{8,2}$ patterns, which means that a

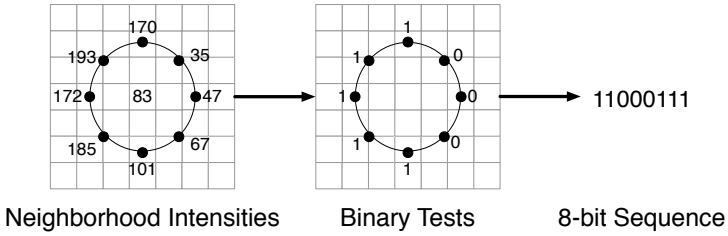

Figure 4.2. An example of Local Binary Pattern ($LBP_{8,2}$). For a given pixel, the Local Binary Pattern is a 8-bit code obtained by verifying whether the intensity of the pixel is greater or lower than its 8 neighbors.

circle of radius 2 is considered centered on each block. For each circle, the intensity of the center pixel is compared to the interpolated intensities located at 8 equally-spaced locations on the circle, as shown on Figure 4.2, left. These eight binary tests (lower or greater intensity) result in an 8-bit sequence, see Figure 4.2, right. Hence, each block pixel is mapped to a sequence among $2^8 = 256$ possible sequences and each block can therefore be represented as a 256-bin histogram. In fact, it has been observed that the bins corresponding to non-uniform sequences (sequences with more than 2 transitions $1 \rightarrow 0$ or $0 \rightarrow 1$) can be merged, yielding more compact 59-bin histograms without performance loss [Ojala et al., 2002].

Color histograms are obtained by k-means clustering. The color codebook is learned from the Red-Green-Blue pixels of the training pictures, and the histogram of a block is obtained by mapping each block pixel to the closest codebook color.

Finally, the histograms describing color and edge statistics of each block are concatenated, which yields a single vector descriptor per block. Our local descriptor representation is therefore simple, relying on both a basic segmentation approach and simple features. Of course, alternative representations could have been used, e.g. [Feng and R. Manmatha, 2004; Grangier et al., 2006a; Tieu and Viola, 2004]. However, this work focuses on the learning model, and a benchmark of picture representations is beyond the topic of this research.

4.3.3 Picture Kernels

Our model relies on a kernel function $k : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ over the picture space $\mathcal{P}$, as explained in Section 4.2. Given our picture representation, we hence need a kernel to compare bags of local descriptors. Fortunately, several kernels comparing sets of feature vectors have been proposed along with the development

of local descriptors [Wallraven and Caputo, 2003; Kondor and Jebara, 2003; Lyu, 2005].

Distribution Kernel approaches fit a distribution $p(v|p)$ over the space of local descriptors for each picture p, and then apply a kernel between distributions to compare pictures. Such kernels includes the Bhattacharya kernel or the expected likelihood kernel [Jebara and Kondor, 2003].

In this study, we fit a Gaussian Mixture Model for each picture p through Expectation-Maximization, as proposed in [Lyu, 2005]. Motivated by scalability issues, we fit standard Gaussians on the input space, not kernelized Gaussian mixtures like [Lyu, 2005]. The learned distributions are then compared with the Expected Likelihood Kernel (ELK),

$$k^{\mathrm{ELK}}(p, p') = \int_v p(v|p)\ p(v|p')dv,$$

which can be computed in closed form for Gaussian mixtures [Jebara and Kondor, 2003; Lyu, 2005].

Matching Kernel approaches [Wallraven and Caputo, 2003] rely on a *minor* kernel, k^l, that compares local descriptors. The kernel between two sets of local descriptors, $p = \{d_{p,i}\}_{i=1}^{|p|}$ and $p' = \{d_{p',i}\}_{i=1}^{|p'|}$, is defined as the average of the best-match-score between the descriptors of p and p',

$$k^{\mathrm{match}}(p, p') = \frac{1}{2}\left[\hat{k}(p, p') + \hat{k}(p', p)\right],$$

$$\text{where} \quad \hat{k}(p, p') = \frac{1}{|p|}\sum_{i=1}^{|p|} \max_j k^l(d_{p,i}, d_{p',j}).$$

Formally, this function k^{match} is not a true Mercer kernel, since its Gram matrix is not always positive definite [Boughorbel et al., 2004]. However, in practice, it can be used with SVM or PAMIR, without enjoying the same theoretical guarantee as a true kernel [Boughorbel et al., 2004]. Empirically, SVMs relying on this kernel have shown to be effective over several object categorization tasks [Boughorbel et al., 2004; Eichhorn and Chapelle, 2004; Wallraven and Caputo, 2003].

Visterm Kernel approaches explicitly represent the pictures in a high dimensional vector space, where the linear kernel is applied. For that purpose, each local descriptor of a picture p is represented as a discrete index, called *visual term* or *visterm*, and, like for text data, the picture is represented as a *bag-of-visterms* vector, in which each component p_i is related to the presence or absence of visterm i in p.

The mapping of the descriptors to discrete indexes is performed according to a codebook C, which is typically learned from the local descriptors of the

training pictures through the k-means algorithm [Duygulu et al., 2002; Jeon and Manmatha, 2004; Quelhas et al., 2005]. This study also applies this standard strategy. The assignment of the weight p_i of visterm i in picture p is classical as well,

$$p_i = \frac{vtf_{i,p}\ vidf_i}{\sqrt{\sum_{j=1}^{|C|}(vtf_i\ vidf_i)^2}},$$

where vtf_i, the term frequency of i in p, refers to the number of occurrences of i in p, while $vidf_i$, the inverse document frequency of i, is defined as $-log(r_i^v)$, r_i^v being the fraction of training pictures containing at least one occurrence of i.

Each of the presented kernels proposes a different technique to compare bags of local descriptors, whose effectiveness highly depends on the application context. For our task, we selected the most appropriate kernel through validation, as explained in the next section.

4.4 Experiments and Results

In this section, we present the experiments performed to evaluate PAMIR. We first describe our experimental setup, and then discuss the various issues related to hyperparameter selection, including the choice of a suitable kernel. Finally, we report the experimental results comparing PAMIR to the alternative models presented in Section 4.1.

4.4.1 Experimental Setup

The datasets used for evaluation originate from stock photography, one of the application context of query-by-text image retrieval. Data from other domains, such as web search engine or newspaper archive, could also have been used. However, we decided to focus on stock photography, since the annotations associated with such pictures are generally produced by professional assessors with well defined procedures, which guarantees a reliable evaluation.

Two datasets are used in our experiments, $\texttt{Corel}^{\texttt{Small}}$ and $\texttt{Corel}^{\texttt{Large}}$. Both sets originate from the *Corel* stock photography collection [Corel], which offers a large variety of pictures, ranging from wilderness scenes to architectural building pictures or sport photographs. Each picture is associated with a textual caption that depicts the main objects present in the picture, see Figure 4.3.

$\texttt{Corel}^{\texttt{Small}}$ corresponds to the 5,000-picture set presented in [Duygulu et al., 2002]. This set, along with the provided split between development and test data, has been used extensively in the query-by-text literature, e.g. [Barnard

coast water
lighthouse cliff

people market
food parasol

Figure 4.3. Examples of Corel pictures along with the associated captions.

et al., 2003; Jeon and Manmatha, 2004; Monay and Gatica-Perez, 2004]. It is composed of a $4,500$-picture development set P^s_{dev} and a 500-picture test set P^s_{test}. For model training and hyperparameter selection, we further divided the development set into a $4,000$-picture train set P^s_{train} and a 500-picture validation set P^s_{valid} (see Table 4.1).

The queries needed to train and evaluate our model originate from the caption data. For that purpose, we first defined the relevance assessments considering that a picture p is relevant to a query q if and only if the caption of p contains all query words. Then, we defined the query set, Q^s_{train}, Q^s_{valid}, or Q^s_{test}, as the set containing all the queries for which there is at least one relevant picture in the picture set, P^s_{train}, P^s_{valid}, or P^s_{test}. This strategy defining queries and relevance assessments is hence not identical to a labeling in which a human assessor issues queries and labels pictures. However, it is based on manually produced captions and the resulting relevance information can be considered as reliable. In fact, there is no doubt that the pictures marked as relevant according to the definition above are indeed relevant, e.g. if the words *beach, sky* are present in a caption, it can confidently be claimed that the corresponding picture is relevant to the queries *"beach", "sky"* and *"beach sky"*. The only problem that could affect our relevance data is due to the possible incompleteness of some captions. If a word is missing from a caption, the corresponding picture will wrongly be marked as non-relevant for all queries containing this word. This weakness is however not specific to our labeling process. For instance, *system pooling*, the semi-automatic technique used for labeling data in retrieval benchmarks, also underestimates the number of relevant documents [Baeza-Yates and Ribeiro-Neto, 1999].

$\texttt{Corel}^{\texttt{Small}}$ statistics are summarized in Table 4.1. The datasets are used

as follows: the parameter vector $\mathbf{w}$ is learned over $(P^s_{\text{train}}, Q^s_{\text{train}})$ through the training procedure defined in Section 4.2. Hyperparameters, such as the number of training iterations, or the type of kernel used, are selected over $(P^s_{\text{valid}}, Q^s_{\text{valid}})$. Final evaluation is conducted over $(P^s_{\text{test}}, Q^s_{\text{test}})$. The training and evaluation of the alternative models is also performed over to the exact same data split, as it is the only way to conduct a fair comparison between the models [Mueller et al., 2002].

Table 4.1. $\texttt{Corel}^{\texttt{Small}}$ Statistics

	train	valid	test
Number of pictures	4,000	500	500
Picture size	384x256 or 256x384		
Number of queries	7,221	1,962	2,241
Avg. # of rel. pic. per q.	5.33	2.44	2.37
Vocabulary size		179	
Avg. # of words per query	2.78	2.51	2.51

The second dataset, $\texttt{Corel}^{\texttt{Large}}$, contains 35,379 images and hence corresponds to a more challenging retrieval problem than $\texttt{Corel}^{\texttt{Small}}$. Like for the smaller set, $\texttt{Corel}^{\texttt{Large}}$ pictures originate from the *Corel* collection and $\texttt{Corel}^{\texttt{Large}}$ queries have been defined relying on the picture captions as explained above. The statistics of the training, validation and test sets of $\texttt{Corel}^{\texttt{Large}}$ are reported in Table 4.2.

For both datasets, performance evaluation has been conducted relying on the standard information retrieval measures defined in Chapter 2: average precision (AvgP), precision at top 10 (P10), and break-even point (BEP). In the following, we report the performance of PAMIR and the alternative models as the average of these measures over the sets of test queries Q^s_{test} and Q^l_{test}.

4.4.2 Hyperparameter Selection

This section studies the influence of the hyperparameters on PAMIR performance. The feature extractor parameters, the type of kernel used, and the learning algorithm parameters are selected through validation: the model is trained with different parameter values over the training set and the parameters achieving the highest average precision over the validation set are selected. For $\texttt{Corel}^{\texttt{Small}}$, all types of parameters are validated. For $\texttt{Corel}^{\texttt{Large}}$, only the learning parameters are validated for efficiency reasons, keeping the feature extractor and kernel parameters to the value selected over $\texttt{Corel}^{\texttt{Small}}$.

Table 4.2. $\texttt{Corel}^{\texttt{Large}}$ Statistics

	train	valid	test
Number of pictures	14,861	10,259	10,259
Picture size	384x256 or 256x384		
Number of queries	55,442	39,690	39,613
Avg. # of rel. pic. per q.	3.79	3.51	3.52
Vocabulary size		1,892	
Avg. # of words per query	2.75	2.72	2.72

Table 4.3. Selecting the block size over $(Q^s_{\text{valid}}, P^s_{\text{valid}})$. The other hyperparameters (kernel and learning parameters) are set to their optimal validation value.

block size	32	48	64	96	128	192	256
blocks per pic.	345	135	77	28	15	3	2
AvgP (valid.)	26.1	25.3	27.3	25.3	22.3	17.8	18.3

Feature extraction requires to select the block segmentation parameters (block size and block overlap) and the number of clusters used for color quantization. The block size determines the trade-off between obtaining local information (with small blocks) and extracting reliable statistics for each block (with large blocks), this parameter is selected through validation. Block overlap is set to half the block size such that all pixels belong to the same number of blocks, to avoid the predominance of pixels located at the block borders. The number of color bins is set to 50, as a trade-off between extracting a compact block representation and obtaining a perceptually good image reconstruction. Table 4.3 reports the validation performance for different block sizes. These results show that large blocks (> 128 pixels) are not suitable for our retrieval problem. In fact, it seems that considering less than 15 local descriptors per image does not provide PAMIR with enough statistics to address the retrieval task. The performance is stable for small blocks, between 32 and 96 pixels, with a slight advantage for 64 pixel blocks. We therefore pick this latter value for evaluation.

The selection of the kernel is also performed through validation. In fact, the different kernels comparing bag-of-local descriptors have been proposed recently and few studies focused on the empirical comparison of these approaches [Eich-

horn and Chapelle, 2004]. Table 4.4 reports the best validation performance for each kernel, along with its parameters. Among the three kernels evaluated, the visterm kernel is clearly yielding the best performance, followed by the match kernel and then the Expected Likelihood Kernel. These results yields several remarks.

The Expected Likelihood Kernel (ELK) over Gaussian mixtures surprisingly yields its best results with only a single Gaussian per picture. This observation is not in line with the handwritten digit recognition experiments reported in [Lyu, 2005]. Even if the differences in the datasets and the tasks performed might explain this difference, we further investigated on this point. In fact, the non-convex Expectation-Maximization procedure seems to explain the failure of ELK over Gaussian mixtures. The fitting of a mixture over the same picture with different initializations yield similar distributions in terms of data likelihood. However, these distributions are not equivalent for ELK evaluations and large relative variations are observed for a given pair of pictures, depending on the initialization of the Expectation-Maximization procedure for these pictures. This effect could possibly be reduced through averaging, if one fits multiple mixtures per picture. However, such a solution would be too costly for large datasets.

The performance of the match kernel is reported to be higher than the ELK. The match kernel relies on a *minor* kernel to compare pairs of local descriptors. In our experiments, the linear kernel, the Radial Basis Function (RBF) kernel, and the polynomial kernel have been tried as minor kernels. Table 4.4 reports results only for the RBF kernel, which yielded the highest validation performance. Regarding efficiency, the match kernel is computationally demanding as it needs to compare all pairs of local descriptors between two pictures.

The visterm kernel is reported to yield the highest validation performance and optimal performance is reached with a codebook of $10,000$ prototypes. Moreover, the visterm approach also yields a more efficient model, compared to the other kernels. In fact, the visterm framework represents the pictures as bag-of-visterms vectors, where the linear kernel is applied. This means that the picture vectors can be pre-computed, as soon as the pictures are available. Then, model training and testing only require the evaluations of the linear kernel between sparse vectors. Such an operation can be performed efficiently as its complexity only depends on the number of non-zero components of the vectors (bounded by 77, the number of blocks per image), not on the data dimension ($10,000$, the codebook size) [Baeza-Yates and Ribeiro-Neto, 1999]. Furthermore, the linear kernel allows for handling $\mathbf{w}$ explicitly, which involves

Table 4.4. Selecting the kernel over $(Q^s_{\text{valid}}, P^s_{\text{valid}})$. The other hyperparameters (feature extractor and learning parameters) are set to their optimal validation value.

Kernel	AvgP	Parameters
Exp. Likelihood	23.1	num. of Gaussians per picture (1)
Match	25.6	stdv of the local RBF kernel (5)
Visterm-Linear	27.3	codebook size (10,000)

Table 4.5. Selecting the parameters of the learning procedure. The other hyperparameters (feature extractor and kernel parameters) are set to their optimal $\texttt{Corel}^{\texttt{Small}}$ validation value.

Dataset	Aggressiveness c	Num. of iter. n
$\texttt{Corel}^{\texttt{Small}}$	0.1	2.53×10^6
$\texttt{Corel}^{\texttt{Large}}$	0.1	1.55×10^7

much less computation than handling support sets.

The training parameters of PAMIR are the number of iterations n and the aggressiveness c. Both of them sets the trade-off between the two learning objectives, i.e. minimizing the training loss and identifying a large margin model. Table 4.5 reports the selected values. For both $\texttt{Corel}^{\texttt{Large}}$ and $\texttt{Corel}^{\texttt{Small}}$, the number of iterations is significantly lower than the number of training constraints (e.g. for $\texttt{Corel}^{\texttt{Small}}$, 2.53×10^6 iterations should be compared to 1.45×10^8 training constraints). The algorithm hence converges before examining all the training set, which is certainly due to some redundancy in the training data. This highlights the efficiency of the PA approach, compared to other optimization techniques for SVM-like problems, as discussed in Section 4.2.

To conduct a fair comparison, the alternative models have been trained over the same local descriptors and their hyperparameters have been selected with the same validation procedure. Namely, we selected the block size (for all models), the visual codebook size (for CMRM, CMTT and PLSA), and the kernel along with the corresponding parameters (for concept classification SVM) based solely on the validation set of $\texttt{Corel}^{\texttt{Small}}$, while all other parameters have been validated for both $\texttt{Corel}^{\texttt{Small}}$ and $\texttt{Corel}^{\texttt{Large}}$, see Table 4.6.

Table 4.6. Hyperparameters for CMRM, CMTT, PLSA and SVM

Model	Dataset	Hyperparameters
CMRM	$\texttt{Corel}^{\texttt{Small}}$	block size (192), visual codebook size (3,000), smoothing parameters ($\alpha = 0.5$, $\beta = 0.1$)
	$\texttt{Corel}^{\texttt{Large}}$	block size (192), visual codebook size (3,000), smoothing parameters ($\alpha = 0.2$, $\beta = 0.1$)
CMTT	$\texttt{Corel}^{\texttt{Small}}$	block size (256), visual codebook size (2,000), number of singular values kept (50)
	$\texttt{Corel}^{\texttt{Large}}$	block size (256), visual codebook size (2,000), number of singular values kept (1,000)
PLSA	$\texttt{Corel}^{\texttt{Small}}$	block size (32), visual codebook size (50,000), number of aspects (400)
	$\texttt{Corel}^{\texttt{Large}}$	block size (32), visual codebook size (50,000), number of aspects (600)
SVM	$\texttt{Corel}^{\texttt{Small}}$	block size (48), kernel (visterm kernel with a 20,000-visterm codebook)
	$\texttt{Corel}^{\texttt{Large}}$	block size (48), kernel (visterm kernel with a 20,000-visterm codebook)

Note that Table 4.6 does not report the regularization parameter (C) for the SVM as it has been individually tuned for each term.

Before presenting the generalization performance, we briefly compare the computational time required by the different models, for both indexing and retrieval. Table 4.7 reports the indexing times needed by PAMIR and the alternative models. *Indexing* corresponds to all the computations performed prior to the submission of the test queries, once the test pictures are available, excluding the operations related to feature extraction, such as visterm quantization. Indexing can hence be performed *off-line*, before the user can interact with the system. In the case of PAMIR, it includes the training step, plus the mapping of each test picture to the text space. For bi-modal generative models (CMRM, CMTT and PLSA), it corresponds to model training, plus the inference of $p(t|p)$ for each vocabulary term t and each test picture p. In the case of concept classification SVM, it corresponds to the training of an SVM for each vocabulary term, and the classification of each test image according to each of the trained SVMs. Table 4.7 shows that PAMIR training procedure yields the most efficient training time over $\texttt{Corel}^{\texttt{Large}}$. This table also shows that SVM for concept classification is especially costly: this approach involves training a model for each vocabulary term, and each model training has a complexity

Table 4.7. Indexing times for PAMIR and the alternatives models. Execution times have all been measured in seconds on the same machine (AMD Athlon64, 2.4Ghz, 2GB RAM).

	CMRM	CMTT	PLSA	SVM	PAMIR
$\texttt{Corel}^{\texttt{Small}}$	3	9	240	687	17
$\texttt{Corel}^{\texttt{Large}}$	849	4,099	1,025	24,650	450

Table 4.8. Retrieval times for PAMIR and the alternatives models. All models have the same retrieval complexity. Execution times have all been measured on the same machine (AMD Athlon64, 2.4Ghz, 2GB RAM).

	CMRM	CMTT	PLSA	SVM	PAMIR
$\texttt{Corel}^{\texttt{Small}}$	0.34 ms per query				
$\texttt{Corel}^{\texttt{Large}}$	7.94 ms per query				

that grows at least quadratically with the training set size [Joachims, 1998]. This makes the application of this technique challenging for large datasets such as $\texttt{Corel}^{\texttt{Large}}$. Of course, the reported times highly depend on implementation details and optimization tricks (our implementation of PAMIR is available at `www.idiap.ch/pamir/`), and should be considered carefully. It should also be noted that the reported times correspond to a single run of training, while, in a real-world usage scenario, a variable number of runs might be required depending on the number of hyperparameter values selected for validation. However, the results clearly indicate that indexing a corpus with PAMIR is not more costly than indexing a corpus with the other models. After indexing, all models then need to compute the dot-product matching between the submitted query and the textual representations inferred from the text pictures, before ranking the obtained scores. All models hence yield the *same* retrieval time, 0.34 msec per query for $\texttt{Corel}^{\texttt{Small}}$ and 7.94 msec per query for $\texttt{Corel}^{\texttt{Large}}$, on our reference machine, see Table 4.8. This hence means that all models can be used interactively by the user, without any perceived delay.

4.4.3 Experimental Results

This section evaluates PAMIR and the alternative models over the test parts of $\texttt{Corel}^{\texttt{Small}}$ and $\texttt{Corel}^{\texttt{Large}}$.

Table 4.9. Averaged Performance on $\texttt{Corel}^{\texttt{Small}}$ Test Queries. Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
AvgP (%)	19.2	19.8	20.7	22.0	**26.3**
BEP (%)	13.1	13.7	12.8	13.8	**17.4**
P10 (%)	7.8	7.6	8.7	9.3	**10.0**

Table 4.9, which reports the results over $\texttt{Corel}^{\texttt{Small}}$, shows that PAMIR outperforms all the alternative evaluated models. Compared to the best alternative, SVM, a relative improvement of 21% is reported for AvgP (26.3% for PAMIR versus 22.0% for SVM). Improvements are also observed for the other measures, P10 and BEP, which means that the use of PAMIR is advantageous for both users focussing on the first positions of the ranking (as shown by P10 results) or users focussing on the whole ranking (as shown by AvgP results). One should note that the relatively low values reported for the P10 results does not indicate a failure of the models but reflects the difficulty of the task: in fact, the optimal value for P10 is 20.2% due to the low number of relevant pictures per query. This therefore means that the PAMIR user focussing only on the first ten results will retrieve about half the pictures he would have retrieved using the ideal ranker. In order to verify whether the observed advantage on the average results could be due to a few queries, we further ran the Wilcoxon signed rank test to compare PAMIR and each alternative model [Rice, 1995]. This test examines the distribution of the differences in the score obtained for each query and verifies whether it is symmetric around zero, which would mean that PAMIR has actually no advantage over the alternative approach. The test rejected this hypothesis at the 95% confidence level for all alternative models and all measures, as indicated by the bold numbers in the tables.

In order to compare the models over difficult and easy queries, we split the set of test queries into an 'easy' set, containing the queries with 3 or more relevant pictures in P^s_{test}, and a 'difficult' set, containing the queries with only one or two relevant pictures in P^s_{test}. Table 4.10 reports the average precision obtained over the two sets. PAMIR is shown to be the best model over both sets and its advantage is reported to be greater over the 'difficult' set (on this set, the relative AvgP improvement compared to SVM, the second best model, is +29%, as compared to +3.2% over the 'easy' set). This outcome is certainly due

Table 4.10. AvgP (%) for Easy and Difficult Queries of $\texttt{Corel}^{\texttt{Small}}$. The 'easy' query set contains the 421 test queries with 3 or more relevant pictures, while the 'difficult' query set contains the 1,820 test queries with only 1 or 2 relevant pictures. Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
Easy Queries	34.0	31.3	38.0	41.9	**43.3**
Difficult Queries	15.8	17.2	16.7	17.3	**22.4**

to PAMIR ranking criterion, since previous work showed that similar criteria for classification are especially adapted to unbalanced problems, i.e. classification tasks with a low percentage of positive examples [Rakotomamonjy, 2004].

As a further comparison, Table 4.11 reports the average precision obtained over single and multiple-word queries separately. Several previous papers focused on single-word queries only, e.g. [Jeon and Manmatha, 2004; Monay and Gatica-Perez, 2004; Pan et al., 2004], and reporting those results allows for direct comparison with this literature. The single-word queries correspond to an easier task since the average number of relevant pictures per query is 9.4 for the single-word queries, compared to 1.8 for the multiple-word queries. The results reported in Table 4.11 agree with this observation and all models are reported to reach higher performance on the single-word queries compared to multiple-word queries. On both query subsets, the advantage of PAMIR is confirmed. The PAMIR improvement is shown to be greater for multiple-word queries (+22.3% relative improvement in AvgP compared to the second best model, SVM) than for single-word queries (+4.0% relative improvement in AvgP compared to SVM). Two characteristics of PAMIR might explain this outcome: PAMIR training criterion has shown to be adapted to retrieval problems with few relevant pictures, which is the case of multiple-word queries. Moreover, PAMIR is the only model trained over multiple-word queries, which certainly helps achieving better performance over such queries at test time. In fact, we observed that, for multiple-word queries, the other models often favor one of the query terms at the expense of the others. Figure 4.4 shows, for instance, that SVM favors the term 'car' at the expense of 'building' for the query 'building car'. On this example, the SVM ranking provides only one picture containing both cars and buildings, while PAMIR succeed in retrieving all the 3 relevant pictures in the top 5 positions. The PAMIR results even provide a

Table 4.11. AvgP (%) on Single & Multi-Word Queries of $\texttt{Corel}^{\texttt{Small}}$. $\texttt{Corel}^{\texttt{Small}}$ contains 179 test queries with a single word and 2,062 queries with more than one word. Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
Single-Word Que.	25.8	26.4	31.7	32.7	**34.0**
Multi-Word Que.	18.6	19.3	19.7	21.0	**25.7**

non-relevant picture that could have been labeled relevant with looser labeling instructions (see the fifth picture of the ranking). The other example on Figure 4.4 is a single word query, 'petals'. It yields good results for both models, which retrieve 3 relevant pictures out of 4 in the top 5 positions. One can note a slight advantage for PAMIR that returns only flower-related pictures. Of course, these examples have limited statistical values but they give an idea on the type of ranking the user is facing.

With our setup, some queries appear in both the test and train sets (for instance, single-word queries are common to both sets). In order to verify the ability of PAMIR to generalize to new queries, we evaluated our model on the 601 test queries that are not present in the training set. These queries can be considered as difficult, not only because the model has not seen pictures relevant to them during training, but also because they have very few relevant documents (1.03 on average). This second aspect can easily by explained if one remark that test queries with many relevant test pictures are also likely to have at least one relevant picture within the training data, which means that such queries are likely to belong to the training set as well. The results over this set of queries confirm the results observed on the whole set (see Table 4.12) and PAMIR is reported to outperform the alternative according to all measures. Moreover, for all models, the performance is much lower than for the 'difficult' query set (see Table 4.10), which indicates that generalization to new queries deserves to be investigated further in the future.

Overall, the results over $\texttt{Corel}^{\texttt{Small}}$ outline the advantage of PAMIR over the alternative solutions. This outcome is certainly due to our discriminative learning strategy. The training of the other models either maximizes the joint picture/caption likelihood (CMTT, CMRM and PLSA) or minimizes the error rate of the per-term classification problems (SVM for concept classification), while our model relies on a ranking criterion, related to the final retrieval per-

Table 4.12. Results over Test-Only Queries of $\texttt{Corel}^{\texttt{Small}}$ Queries. Among the 2,241 test queries of $\texttt{Corel}^{\texttt{Small}}$, 601 queries are not appearing in the training or in the validation set. Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
AvgP (%)	12.7	12.9	11.1	10.1	**16.1**
BEP (%)	7.1	6.3	4.1	3.1	**7.7**
P10 (%)	2.5	2.7	2.9	2.5	**3.5**

Table 4.13. Averaged Performance on $\texttt{Corel}^{\texttt{Large}}$ Queries. Bold numbers report when a model outperforms all others according to the Wilcoxon test at the 95% confidence level.

	CMRM	CMTT	PLSA	SVM	PAMIR
AvgP (%)	2.11	2.23	2.61	3.60	3.65
BEP (%)	1.26	1.46	1.69	1.81	**1.90**
P10 (%)	1.44	1.49	1.79	2.26	**2.53**

formance. This difference has shown to be especially helpful for both difficult queries (queries with few relevant pictures) and multiple-word queries.

Table 4.13 reports the results of the experiments performed over $\texttt{Corel}^{\texttt{Large}}$. The reported performance over this set are much lower than for $\texttt{Corel}^{\texttt{Small}}$, which is not surprising considering the difficulty of the task. In $\texttt{Corel}^{\texttt{Large}}$, the relevant pictures account for 0.27 per thousand on average, which should be compared to 4.7 per thousand on average for $\texttt{Corel}^{\texttt{Small}}$. Moreover, the limited amount of relevant material present in the training set of $\texttt{Corel}^{\texttt{Large}}$ also makes this task more difficult: in $\texttt{Corel}^{\texttt{Large}}$, the average number of relevant pictures per training query is 3.79, which should be compared to 5.33 for $\texttt{Corel}^{\texttt{Small}}$ (see Table 4.1 and 4.2). Hence, the models trained over $\texttt{Corel}^{\texttt{Large}}$ should address a more difficult ranking problem, while having seen less relevant pictures to generalize from. In fact, the statistics of $\texttt{Corel}^{\texttt{Large}}$ make this task closer to real world applications, such as image search for stock photography or news wire services, and the results over $\texttt{Corel}^{\texttt{Large}}$ are hence of a greater interest from a user perspective.

Although low, the results over $\texttt{Corel}^{\texttt{Large}}$ are much higher than random

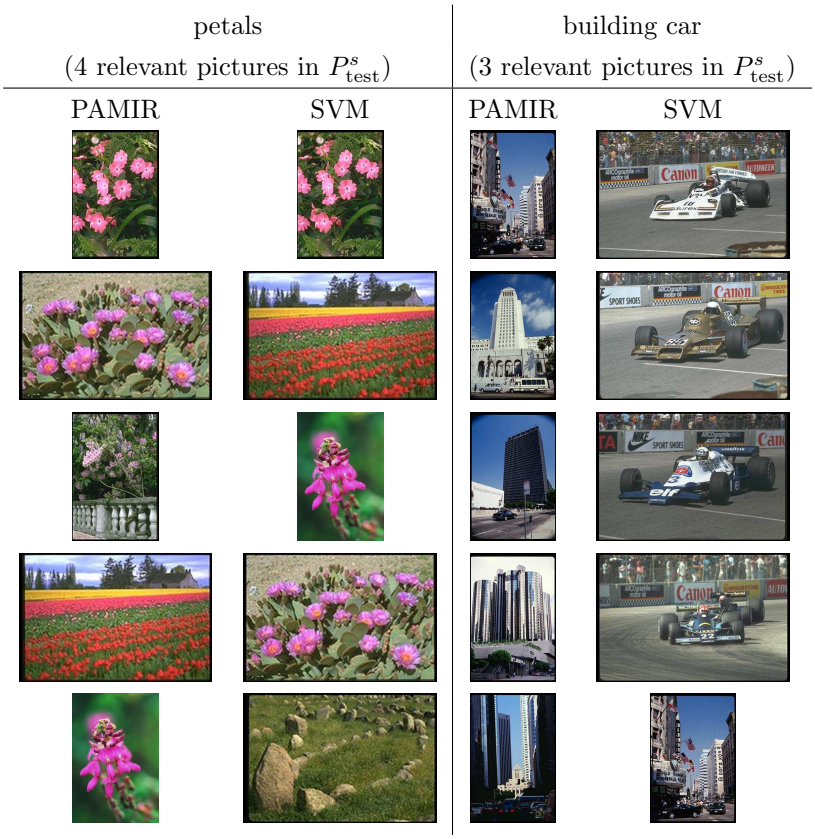

Figure 4.4. Example: the top 5 pictures obtained with PAMIR and SVM, for two queries over $\texttt{Corel}^{\texttt{Small}}$. Higher resolution images, as well as other examples, are available at `www.idiap.ch/pamir/`.

performance for all models (e.g. random performance is $\sim 0.03\%$ for P10 which is much lower than 1.44%, the worst P10 results, obtained with CMRM). All approaches can hence leverage from the training data. In fact, even if the models are far from optimal performance, they can still be useful to the user, as illustrated by the two queries shown on Figure 4.5. The first example 'tree snow people' corresponds to a relatively easy query with 13 relevant pictures in the test set. Like for the 'building car' example on $\texttt{Corel}^{\texttt{Small}}$, the SVM solution is dominated by one of the concepts, 'snow', at the expense of the others, and does not retrieve any relevant picture in the top 5. On the contrary, PAMIR, which is directly trained from multiple-word queries, yields high performance with 3 relevant pictures within the top 5 positions. The second query 'zebra herd' has less relevant pictures (4 in the test set). The results show a slight

advantage for PAMIR: our model retrieves two relevant pictures at the third and fourth positions, while the SVM retrieves one relevant picture at the fifth position. This example illustrates that both models are often confused by similar pictures (savanna scenes in this case) for concepts with few training instances (only 22 pictures contain zebras among the 14,861 pictures of P^l_{train}).

Like for $\texttt{Corel}^{\texttt{Small}}$, the results in Table 4.13 clearly show the advantage of PAMIR over the other approaches. In fact, model comparison yields similar conclusions over $\texttt{Corel}^{\texttt{Small}}$ and $\texttt{Corel}^{\texttt{Large}}$: CMTT and CMRM reach comparable performance levels, PLSA performs better than the other generative models, but not as well as the SVM. Again, PAMIR yields the best results. Furthermore, the Wilcoxon test over $\texttt{Corel}^{\texttt{Large}}$ concludes that PAMIR significantly outperforms each alternative, at the 95% confidence level, for P10 and BEP. For AvgP, the test concludes that PAMIR outperforms all generative models (CMTT, CMRM and PLSA), and yields an AvgP similar to the SVM's. Overall, the results over both sets are consistent and show the advantage of our discriminative model over the alternatives.

4.5 Conclusions

We have proposed a discriminative approach to the retrieval of images from text queries. In such a task, the model receives a picture corpus P and a text query q. It should then rank the pictures of P such that the pictures relevant to q appear above the others. Contrary to previous approaches that generally rely on an image auto-annotation framework, our learning procedure aims at selecting the model parameters likely to yield a high ranking performance over the unseen test data. For that purpose, we introduced a loss inspired from ranking SVM [Joachims, 2002] and formalized the notion of margin for our retrieval problem. We then introduced a learning algorithm building upon Passive-Aggressive (PA) minimization [Crammer et al., 2006]. The resulting model, Passive-Aggressive Model for Image Retrieval (PAMIR), has several advantages: its learning objective is related to the final retrieval performance, its training procedure allows an efficient for learning over large datasets, and the model parameterization can benefit from picture kernels recently introduced in the computer vision literature [Kondor and Jebara, 2003; Lyu, 2005; Wallraven and Caputo, 2003]. These advantages actually yield a model effective in practice, as shown by our experiments over stock photography data. For instance, over the standard *Corel* benchmark [Duygulu et al., 2002], PAMIR yields 26.3% average precision, which should be compared to 22.0% for SVM for concept classification, the best alternative. Our model has notably shown

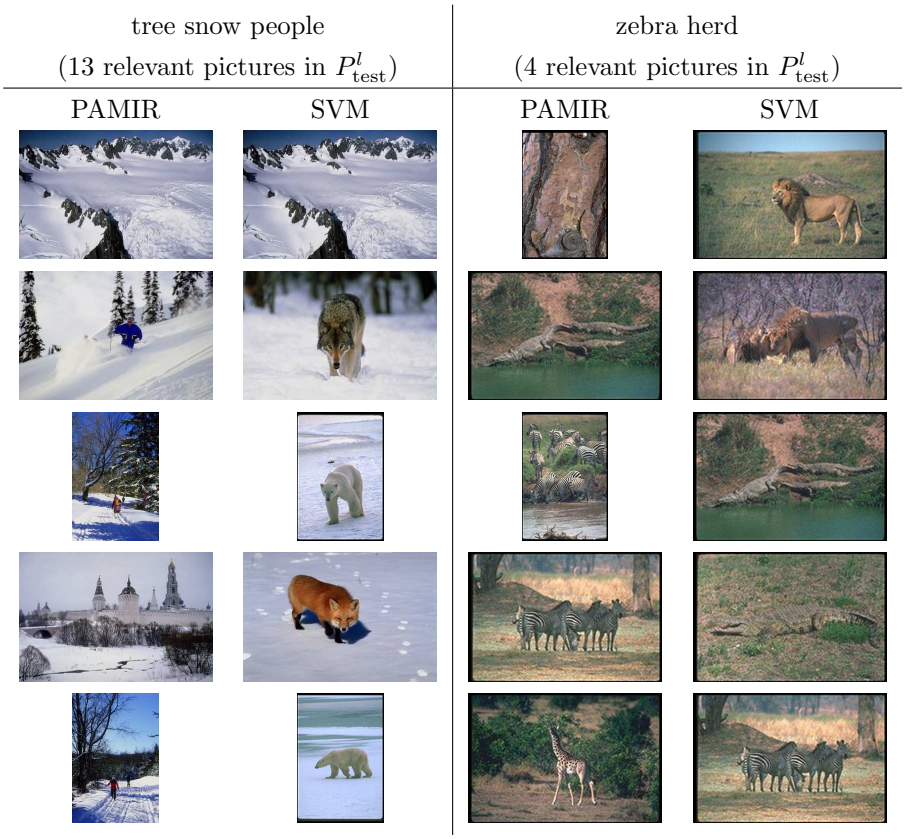

Figure 4.5. Example: the top 5 pictures obtained with PAMIR and SVM for two queries over $\texttt{Corel}^{\texttt{Large}}$. Higher resolution images, as well as other examples, are available at `www.idiap.ch/pamir/`.

to be especially advantageous over multiple-word queries and difficult queries with few relevant pictures.

Although it outperforms the alternative models, PAMIR is far from reaching perfect performance, especially over the challenging $\texttt{Corel}^{\texttt{Large}}$ data. Several directions could be explored to improve our model. First, PAMIR loss function could be changed to give more emphasis on the top of the ranking, as most users examine only the first results. An approach derived from [Joachims, 2005] could be applied to optimize measures like P10. The loss could also be modified to optimize measures considering relevance assessments with gradual relevance levels, such as Discounted Cumulative Gain [Voorhees, 2001]. Another useful extension would be the prediction of a cut-off rank, that is, a ranking position below which the user is unlikely to encounter any relevant documents. Solutions

inspired from [Brinker and Huellermeier, 2005] could help solving this problem. Also, it would worth investigating further on the use of kernels for local features, especially to model the spatial relationships between features [Sivic et al., 2005].

The proposed model, along with the reported results, hence advocates for addressing the image retrieval problem through a discriminative ranking approach, and open several possible directions of research to fully benefit from this formalism.

4.6 Contributions

This chapter summarizes [Grangier et al., 2006b] and [Grangier and Bengio, 2008]. We have also compared the effectiveness of different local features, relying on PAMIR [Grangier et al., 2006a]. Based on the same learning objective, we have proposed a Neural Network for image retrieval [Grangier and Bengio, 2006].

◇ D. Grangier, F. Monay, and S. Bengio. A discriminative approach for the retrieval of images from text queries. In *European Conference on Machine Learning (ECML)*, pages 162–173, Berlin, Germany, September 2006b.

◇ D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*, 2008. (in press).

◇ D. Grangier, F. Monay, and S. Bengio. Learning to retrieve images from text queries with a discriminative model. In *International Workshop on Adaptive Multimedia Retrieval (AMR)*, pages 42–56, Geneva, Switzerland, July 2006a.

◇ D. Grangier and S. Bengio. A neural network to retrieve images from text queries. In *International Conference on Artificial Neural Networks (ICANN)*, volume 2, pages 24–34, Athens, Greece, September 2006.

5 Discriminative Keyword Spotting

This chapter is concerned with the task of keyword spotting. This task aims at detecting the utterances of a given keyword in speech recordings and relates to numerous applications, such as voice mail retrieval or voice command detection. This work proposes a discriminative approach to this problem and introduces a learning algorithm, which aims at maximizing the Area Under the receiver operating Curve (AUC), given a set of training spotting problems. Interestingly, this AUC maximization framework yields a learning criterion related to the pairwise loss for rankings used for image retrieval in the previous chapter. Building upon our image retrieval work, the proposed algorithm relies on a large margin formulation of the spotting task, and adopts an efficient online learning strategy.

Our spotting solution contrasts with current spotting strategies. Previous work concentrated mainly on several variants of Hidden Markov Models (HMMs) to address this intrinsically sequential problem. While the HMM-based approaches constitute the state-of-the-art, they suffer several known limitations. Most of these limitations are not specific to keyword spotting HMMs, and also affect speech recognition HMMs, as pointed out previously, e.g. [Keshet et al., 2006]. We can for instance mention the predominance of the emission probabilities in the likelihood, which tends to neglect duration and transition models, or the Expectation-Maximization training procedure, which is prone to convergence to local optima. Other drawbacks are specific to the application of HMMs to the keyword spotting task. In particular, the rarity of the occurrences of the targeted keyword often requires ad-hoc modifications of the HMM topology, transition probabilities or decoding algorithm. However, the most important limitation of HMM-based approaches lies in their training objective. Typically, HMM learning aims at maximizing the likelihood of tran-

scribed utterances, and does not provide strong guarantees in terms of keyword spotting performance.

In this study, we propose a model to circumvent those limitations. Our approach adopts a learning objective related to the final keyword spotting task. At training time, our keyword spotter is presented a set of spotting problems and the parameters are learned to maximize the performance over these problems, measured by the AUC, the most common measure to evaluate keyword spotters. Furthermore, the proposed learning algorithm adopts a large margin approach and provides theoretical guarantees regarding generalization performance. This framework hence contrasts with HMMs, where parameter learning is not tightly related to the keyword spotting problem. Moreover, our model enjoys further advantages compared to HMMs, including its convex optimization procedure, which avoids convergence to local optima, or its non-probabilistic framework, which offers greater flexibility for selecting the relative importance of duration modeling with respect to acoustic modeling. These advantages actually convert into higher spotting performance, according to our experiments performed over TIMIT and WSJ data. For instance, the best HMM configuration over WSJ reaches 88.4% AUC, compared to 92.2% for our discriminative spotter.

The remainder of this chapter is organized as follows: Section 5.1 describes prior work on keyword spotting, Section 5.2 introduces our discriminative approach, Section 5.3 presents different experiments comparing the proposed model to an HMM-based solution. Finally, Section 5.4 draws some conclusions and delineates possible directions for future research.

5.1 Related Work

Research in keyword spotting has paralleled the development of the Automatic Speech Recognition (ASR) domain in the last thirty years. Like ASR, keyword spotting has first been addressed with models based on Dynamic Time Warping (DTW) [Bridle, 1973; Higgins and Wohlford, 1985]. Then, approaches based on discrete HMMs have been introduced [Kawabata et al., 1988]. Finally, discrete HMMs have been replaced by continuous HMMs [Rabiner and Juang, 1993].

In all cases, the core objective of keyword spotting is to discriminate between the segments of the signal belonging to a keyword utterance and the others. For that purpose, the first approaches based on DTW proposed to compute the alignment distance between a template utterance of the keyword and all possible subsequences of the test signal [Bridle, 1973]. In this con-

text, the keyword is considered as detected for the subsequences for which the distance is below some predefined threshold. Such approaches are however greatly affected by speaker mismatch and varying recording conditions between the template sequence and the test signal. To gain some robustness, it has then been proposed to compute alignment distances not only with respect to the targeted keyword template, but also with respect to other word templates [Higgins and Wohlford, 1985]. Precisely, given a test example, such a system identifies the concatenation of templates with the lowest distance to the signal and the keyword is considered as detected if this concatenation contain the keyword template. Therefore, the keyword alignment distance is not considered as an absolute number, but relatively to the distances to the other templates, which increase robustness with respect to changes in the recording conditions.

Along with the development of the speech research, increasingly large amount of labeled speech data were collected, and DTW-based techniques started showing their limitations to leverage from large training sets. Consequently, discrete HMMs were introduced for ASR [Bahl et al., 1986], and then for keyword spotting [Kawabata et al., 1988; Wilpon et al., 1990]. A discrete HMM assumes that the observations of a sequence of discrete events (i.e. the quantized acoustic vectors of an utterance) are independent conditioned on an hidden state variable that follows a Markov process. This type of model introduces several advantages compared to DTW-based approaches, including an improved robustness to speaker and channel changes, when several training utterances of the targeted keyword are available. However, the most important evolution introduced with the HMM certainly lies in the development of phone or triphone-based modeling [Lee and Hon, 1988a; Kawabata et al., 1988; Rose and Paul, 1990], in which a word model is composed of several sub-unit models shared across words. This means that the model of a given word not only benefit from the training utterances containing this word, but also from all the utterances containing its sub-units. A further advantage of phone-based modeling is the ability to spot words unavailable at training time, as this paradigm allows one to build a new word model by composing already trained sub-unit models. This aspect is very important, as, in most applications, the set of test keywords is not known in advance.

Soon after the application of discrete HMMs to speech problems, continuous density HMMs have been introduced in the ASR community [Rabiner and Juang, 1993]. Continuous HMMs remove the need for acoustic vector quantization, as the distributions associated with the HMM states are continuous densities, generally Gaussian Mixtures. The learning of both the Gaussian Mix-

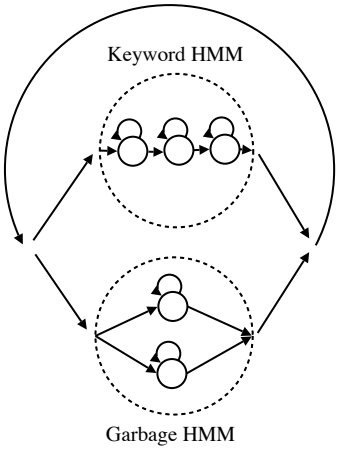

Figure 5.1. HMM topology for keyword spotting with a Viterbi best path strategy. This approach verifies whether the Viterbi best path passes through the keyword sub-model.

ture parameters and the state transition probabilities is performed in a single integrated framework, maximizing the likelihood of the training data given its transcription through the Expectation-Maximization algorithm [Bilmes, 1998]. This approach has shown to be more effective and allows for greater flexibility for speaker or channel adaptation [Rabiner and Juang, 1993]. It is now the most widely used approach for both ASR and keyword spotting. In the context of keyword spotting, different strategies based on continuous HMMs have been proposed. In most cases, a sub-unit based HMM is trained over a large corpus of transcribed data and a new model is then built from the sub-unit models. Such a model is composed of two parts, a keyword HMM and a *filler* or *garbage* HMM, which respectively model the keyword and non-keyword parts of the signal. Figure 5.1 shows such a topology. Given such a model, keyword detection is performed through Viterbi decoding, i.e. by searching for the sequence of states that yields the highest likelihood for the provided test sequence. Keyword detection is determined by checking whether the Viterbi best-path passes through the keyword model or not. In such a model, the selection of the transition probability in the keyword sets the trade-off between low false alarm rate (i.e. detecting a keyword where it is not present), and low false rejection rate (i.e. not detecting a keyword where it is indeed present). Another important aspect of this approach lies in the modeling of non-keyword parts of the signal, and several choices are possible for the garbage HMM. The simplest choice

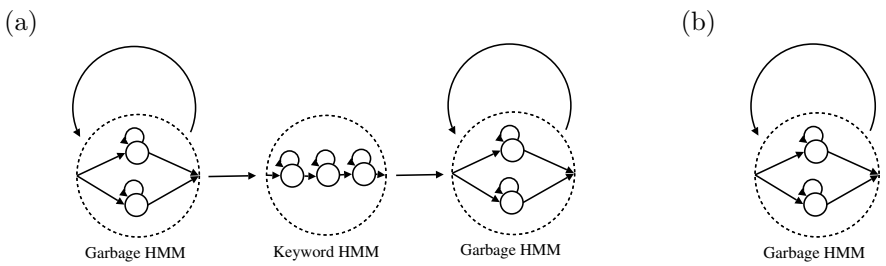

Figure 5.2. HMM topology for keyword spotting with a likelihood ratio strategy. This approach compares the likelihood of the sequence given the keyword is uttered, estimated with (a), to the likelihood of the sequence given the keyword is not uttered, estimated with (b).

models garbage with an HMM that fully connects all sub-units models [Rose and Paul, 1990], while the most complex choice models garbage with a full-large vocabulary HMM, where the lexicon excludes the keyword [Weintraub, 1993]. This latter approach obviously yields a better garbage model, using additional linguistic knowledge. This advantage however induces a higher decoding cost and requires larger amount of training data, in particular for language model training. Besides practical concerns, one can conceptually wonder whether an automatic spotting approach should require such a large linguistic knowledge. Of course, several variations of garbage models exist between the two extreme examples pointed above, see [Boite et al., 1993] for instance.

Viterbi decoding relies on a sequence of local decisions to determine the best path, which can be fragile with respect to local model mismatch. In the context of HMM-based keyword spotting, a keyword can be missed, if only its first phoneme suffers such a mismatch, for instance. To gain some robustness, likelihood ratio approaches have been proposed [Weintraub, 1995; Rose and Paul, 1990]. In this case, the confidence score outputted by the keyword spotter corresponds to the likelihood ratio estimated by an HMM requiring an occurrence of the keyword, and an HMM excluding it, see Figure 5.2. Detection is then performed by comparing the outputted score to a predefined threshold. Different variations on this likelihood ratio approach have then been devised, such as computing the ratio only on the part of the signal detected as the keyword by the keyword model [Junkawitsch et al., 1997]. Overall, all the above described methods are variations over the same HMM paradigm, which consists in training a generative model through likelihood maximization, before intro-

ducing different modifications prior to decoding in order to address the keyword spotting task. In other words, these approaches do not propose to train the model to maximize the spotting performance, and the keyword spotting task is only introduced after training.

Only few studies have proposed discriminative parameter training approaches to circumvent this weakness [Sukkar et al., 1996; Sandness and Hetherington, 2000; Weintraub et al., 1997; Benayed et al., 2003]. [Sukkar et al., 1996] proposes to maximize the likelihood ratio between the keyword and garbage models for keyword utterances and to minimize it over a set of false alarms generated by a first keyword spotter. [Sandness and Hetherington, 2000] proposes to apply Minimum Classification Error (MCE) to the keyword spotting problem. The training procedure updates the acoustic models to lower the score of non-keyword models within keyword occurrences. However, this procedure does not focus on false alarms, and does not aim at lowering the keyword score in non-keyword parts of the signal. Other discriminative approaches have focused on combining different HMM-based keyword detectors. For instance, [Weintraub et al., 1997] trains a neural network to combine likelihood ratios from different models. [Benayed et al., 2003] relies on support vector machines to combine different averages of phone-level likelihoods. Both of these approaches propose to minimize the error rate, which equally weights the two possible spotting errors, false positive (or false alarm) and false negative (missing a keyword occurrence, often called keyword deletion). This measure is however barely used to evaluate keyword spotters, due to the *unbalanced* nature of the problem. Precisely, the targeted keywords generally occurs rarely and the number of potential false alarms hence highly exceeds the number of potential missed detections. In this case, the useless model which never predicts the keyword avoids all false alarms and yields a very low error rate, with which it is difficult to compete. For that reason, the Area Under the receiver operating Curve (AUC), which plots the true positive rate versus the false alarm rate, is more informative and is commonly used to evaluate models. The maximization of the AUC would hence be an appropriate learning objective for the discriminative training of a keyword spotter. To the best of our knowledge, only [Chang, 1995] proposed an approach targeting this goal. This work introduces a methodology to maximize the Figure-Of-Merit (FOM), which corresponds to the AUC over a specific range of false alarm rates. However, the proposed approach relies on various heuristics, such as gradient smoothing and sorting approximations, which does not ensure any theoretical guarantee on FOM maximization. Also, these heuristics involve the selection of several hyperparameters, that challenges

a practical use.

In the following, we introduce a model that aims at maximizing the AUC over a set of training spotting problems, which constitutes a truly discriminative approach to the keyword spotting problem. The proposed model relies on large-margin learning and provides theoretical guarantees regarding the generalization performance. Furthermore, its efficient learning procedure ensures scalability toward large problems and simple practical use, with only 2 hyperparameters to select.

5.2 Discriminative Keyword Spotting

This section formalizes the keyword spotting problem, and introduces the proposed approach. For that purpose, we first describe the methodology generally employed to evaluate keyword spotters. This allows us to introduce a loss derived from the area under the Receiver Operating Curve (ROC), the most common measure of keyword spotting performance. Then, we present our model parameterization and the training procedure to minimize efficiently a regularized version of the loss. Finally, we explain the large margin interpretation of our method, and its generalization guarantees.

5.2.1 Problem Setting

In the keyword spotting task, we are provided with a speech utterance $\overline{x}$ along with a keyword k, and we should determine whether k is uttered in $\overline{x}$. Formally, if the keyword k is represented as a sequence of phonemes, $\overline{p}^k = (p_i)_{i=1}^L \in \mathcal{P}^*$, and the speech utterance $\overline{x}$ is represented as a sequence of frames (or acoustic vectors), $\overline{x} = (x_i)_{i=1}^T \in \mathcal{X}^*$, we should build a function

$$f^{-1/1} : \mathcal{P}^* \times \mathcal{X}^* \to \{-1, +1\},$$

which detects the keyword $f^{-1/1}(\overline{p}^k, \overline{x}) = +1$ or rejects it $f^{-1/1}(\overline{p}^k, \overline{x}) = -1$. For that purpose, we introduce a keyword spotter

$$f : \mathcal{P}^* \times \mathcal{X}^* \to \mathbb{R},$$

whose output $f(\overline{p}^k, \overline{x})$ expresses the confidence that k is uttered in $\overline{x}$. This confidence value can then be compared to a threshold b, to accept $f(\overline{p}^k, \overline{x}) > b$, or reject $f(\overline{p}^k, \overline{x}) < b$ the utterance of k in $\overline{x}$.

Such a keyword spotter f can be evaluated relying on the Receiver Operating Curve (ROC). This curve plots the true positive rate (TPR) as a function of the false positive rate (FPR). The TPR measures the fraction of keyword

occurrences correctly spotted, while the FPR measures the fraction of negative utterances yielding a false alarm. The points on the curve are obtained by sweeping the threshold b from the largest value outputted by the system to the smallest one. These values hence correspond to different trade-offs between the two types of errors a keyword spotter can make, i.e. missing a keyword utterance or rising a false alarm. In order to evaluate a keyword spotter over various trade-offs, it is common to report the Area Under the ROC (AUC). This area hence corresponds to an averaged performance, assuming a flat prior over the different operational settings. Given a keyword k, a set of positive utterances X_k^+ in which k is pronounced, and a set of negative utterances X_k^- in which k is not pronounced, the AUC can be written as,

$$A_k = \frac{1}{|X_k^+||X_k^-|} \sum_{\substack{\overline{x}^+ \in X_k^+ \\ \overline{x}^- \in X_k^-}} \mathbb{1}_{f(\overline{p}^k, \overline{x}^+) > f(\overline{p}^k, \overline{x}^-)},$$

where $|\cdot|$ refers to set cardinality and $\mathbb{1}_{\cdot}$ refers to the indicator function. A_k hence estimates the probability that the score assigned to a positive utterance is greater than the score assigned to a negative utterance. This quantity is also referred to as the Wilcoxon Mann Whitney statistics [Wilcoxon, 1945; Mann and Whitney, 1947; Cortes and Mohri, 2004].

As one is often interested in the expected performance over any keyword, it is common to plot the ROC averaged over a set of evaluation keywords K_{test}, and to compute the corresponding averaged AUC,

$$A_{\text{test}} = \frac{1}{|K_{\text{test}}|} \sum_{k \in K_{\text{test}}} A_k.$$

In this study, we introduce a large-margin approach to learn a keyword spotter maximizing the averaged AUC.

5.2.2 A Loss to Maximize the AUC

In order to build our keyword spotter f, we are given training data consisting of a set of training keywords K_{train} and a set of training utterances. For each keyword $k \in K_{\text{train}}$, we denote with X_k^+ the set of utterances containing the keyword and with X_k^- the other utterances. Furthermore, for each positive utterance $\overline{x} \in X_k^+$, we are also provided with the segmentation $\overline{s}$ of the keyword phoneme sequence $\overline{p}^k$ over $\overline{x}$. Such a segmentation, which provides the start and end points of each phoneme, can either be provided by annotators or localized with an alignment algorithm, such as [Keshet et al., 2007b]. Formally,

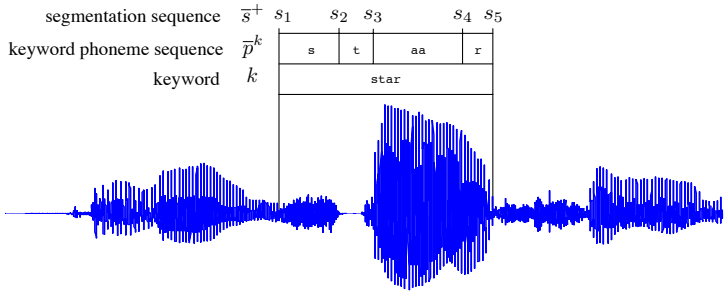

Figure 5.3. Example of our notation. The waveform of the spoken utterance "a lone star shone..." taken from the TIMIT corpus. The keyword k is the word *star*. The phonetic transcription $\overline{p}^k$ along with the segmentation sequence $\overline{s}^+$ are schematically depicted in the figure.

a segmentation is noted as,

$$\overline{s} = (s_i)_{i=1}^{L_k+1},$$

where L_k denotes the number of phoneme in k, and s_i denotes the start time of the i^{th} phoneme, which also corresponds to the end time of the $(i-1)^{\text{th}}$ phoneme. Figure 5.3 illustrates our notations.

Provided with such data, the training AUC is

$$A_{\text{train}} = \frac{1}{|K_{\text{train}}|} \sum_{k \in K_{\text{train}}} \frac{1}{|X_k^+||X_k^-|} \sum_{\substack{\overline{x}^+ \in X_k^+ \\ \overline{x}^- \in X_k^-}} \mathbb{1}_{f(\overline{p}^k, \overline{x}^+) > f(\overline{p}^k, \overline{x}^-)},$$

which can be compactly written as,

$$A_{\text{train}} = \sum_{(k, \overline{x}^+, \overline{x}^-) \in T_{\text{train}}} \beta_k \quad \mathbb{1}_{f(\overline{p}^k, \overline{x}^+) > f(\overline{p}^k, \overline{x}^-)},$$

if we define $T_{\text{train}} = \{(k, \overline{x}^+, \overline{x}^-), \forall k \in K_{\text{train}}, \forall \overline{x}^+ \in X_k^+, \forall \overline{x}^- \in X_k^-\}$ and $\beta_k = \frac{1}{|K_{\text{train}}||X_k^+||X_k^-|}$. The selection of a model maximizing this AUC is equivalent to minimizing the loss

$$\begin{aligned} L^{0/1}(f) &= 1 - A_{\text{train}} \\ &= \sum_{(k, \overline{x}^+, \overline{x}^-) \in T_{\text{train}}} \beta_k \quad \mathbb{1}_{f(\overline{p}^k, \overline{x}^+) \leq f(\overline{p}^k, \overline{x}^-)}. \end{aligned}$$

$L^{0/1}$ is however not suitable for model training since it is piecewise constant, which means that its gradient with respect to f is zero. Therefore, we introduce

an upper bound of $L^{0/1}$,

$$L(f) = \sum_{(k,\overline{x}^+,\overline{x}^-)\in T_{\text{train}}} \beta_k \quad |1 - f(\overline{p}^k, \overline{x}^+) + f(\overline{p}^k, \overline{x}^-)|_+, \tag{5.1}$$

where $|a|_+$ denotes $\max(0,\ a)$. It is easy to verify that $L(f) \geq L^{0/1}(f)$, since $\forall a, b, \quad |1 - a + b|_+ \geq \mathbb{1}_{a\leq b}$. Moreover, this loss ensures that if $L(f) = 0$, then $A_{\text{train}} = 1$, since $\forall a, b, \quad |1 - a + b|_+ = 0 \Rightarrow a > b + 1 \Rightarrow a > b$. This loss is related to the pairwise ranking loss used for the retrieval of images from text queries in Chapter 4. In fact, our AUC maximization objective can be formulated as a ranking objective. Given a keyword and a set of speech utterances, our goal is to rank the utterances containing the keyword above the others. Then, measuring the number of swapped utterance pairs over this ranking task is equivalent to measuring $1 - AUC$ over the original keyword spotting problem. This parallel further justifies our approach, since the pairwise loss has already shown to be to successful over highly unbalanced retrieval problems, see Chapter 4 or [Joachims, 2002], and keyword spotting also corresponds to an highly unbalanced setup, where the utterances containing the keyword only account for a very small part of the data.

5.2.3 Model Parameterization

Our keyword spotter f is parameterized as

$$f_{\mathbf{w}}(\overline{x}, \overline{p}^k) = \max_{\overline{s}} \mathbf{w} \cdot \phi(\overline{x}, \overline{p}^k, \overline{s}),$$

where $\mathbf{w}$ is a vector of importance weights, $\phi(\overline{x}, \overline{p}^k, \overline{s})$ is a feature vector, measuring different characteristics related to the confidence that $\overline{p}^k$ is pronounced in $\overline{x}$ with the segmentation $\overline{s}$. In other words, our keyword spotter outputs a confidence score by maximizing a weighted sum of feature functions over all possible segmentations. This maximization corresponds to a search over an exponentially large number of segmentations. Nevertheless, it can be performed efficiently by selecting decomposable feature functions, which allows the application of dynamic programming techniques, like for HMMs (see Appendix A.2).

Precisely, our feature functions are borrowed from a phoneme alignment algorithm [Keshet et al., 2007b]. These 7 features measure the match between the acoustic sequence $\overline{x}$ and the phoneme sequence $\overline{p}$, as explained in the following.

There are four **phone transition functions**, which aim at detecting transition between phonemes. For that purpose, they compute the frame distance between the frames before and after a hypothesized transition point, i.e.

$$\forall i = 1, 2, 3, 4, \quad \phi_i(\overline{x}, \overline{p}^k, \overline{s}) = \frac{1}{L} \sum_{j=2}^{L-1} d(x_{s_j - i}, x_{s_j + i})$$

where d refers to the Euclidean distance and L refers to the number of phonemes in keyword k.

The **frame classifier function** relies on a frame-based phoneme classifier to measure the match between each frame and the hypothesized phoneme class,

$$\phi_5(\overline{x}, \overline{p}^k, \overline{s}) = \frac{1}{L} \sum_{i=1}^{L} \sum_{j=s_i}^{s_{i+1}-1} \frac{1}{s_{i+1} - s_i} \, g(x_j, p_i) \tag{5.2}$$

where g refers to the phoneme classifier. Different phoneme classifiers might be applied for this feature. In our case, we conduct experiments relying on two alternative solutions. The first assessed classifier is a hierarchical large-margin classifier [Dekel et al., 2004], while the second classifier is a Bayes classifier with one Gaussian Mixture per phoneme class. In the first case, g is defined as the phoneme confidence outputted by the classifier, while, in the second case, g is defined as the log posterior of the class $g(x, p) = \log(P(p|x))$. The presentation of the training setup, as well as, the empirical comparison of both solutions, are deferred to Section 5.3.

The **phone duration function** measures the adequacy of the hypothesized segmentation $\overline{s}$, with respect to a duration model,

$$\phi_6(\overline{x}, \overline{p}^k, \overline{s}) = \frac{1}{L} \sum_{i=1}^{L} \log \mathcal{N}(s_{i+1} - s_i; \mu_{p_i}, \sigma_{p_i}),$$

where $\mathcal{N}()$ refers to the likelihood of a Gaussian duration model, whose mean μ_p and variance σ_p^2 parameters for each phoneme p are estimated over the training data.

The **speaking rate function** measures the stability of the speaking rate,

$$\phi_7(\overline{x}, \overline{p}^k, \overline{s}) = \frac{1}{L} \sum_{i=2}^{L} (r_i - r_{i-1})^2,$$

where r_i refers to the estimate of the speaking rate for the i^{th} phoneme,

$$r_i = \frac{s_{i+1} - s_i}{\mu_{p_i}}.$$

This set of seven functions has been used in our experiments. Of course, this set can easily be extended to incorporate further features, such as confidences from a triphone frame classifier or the output of a more refined duration model.

5.2.4 An Iterative Training Algorithm

As the model parameterization is introduced, our objective is now to identify the vector $\mathbf{w}$ minimizing a regularized version of the loss $L(f_{\mathbf{w}})$ to avoid

overfitting,

$$L^{\text{Reg}}(f_{\mathbf{w}}) = \|\mathbf{w}\|^2 + \quad C \sum_{(k,\overline{x}^+,\overline{x}^-)\in T_{\text{train}}} \beta_k \quad l(\mathbf{w}; \overline{p}^k, \overline{x}^+, \overline{x}^-),$$

where

$$l(\mathbf{w}; \overline{p}^k, \overline{x}^+, \overline{x}^-) = |1 - \max_{\overline{s}} \mathbf{w} \cdot \phi(\overline{x}^+, \overline{p}^k, \overline{s}) + \max_{\overline{s}} \mathbf{w} \cdot \phi(\overline{x}^-, \overline{p}^k, \overline{s})|_+$$

and C is a hyperparameter setting the importance of the training loss versus the regularizer. One can note that $L^{\text{Reg}}(f_{\mathbf{w}})$ is not a convex function of $\mathbf{w}$. In order to benefit from the guarantees of convex optimization [Boyd and Vandenberghe, 2004], we propose to substitute this objective function with the following convex upper bound

$$L^{\text{CReg}}(f_{\mathbf{w}}) = \|\mathbf{w}\|^2 + \quad C \sum_{(k,\overline{x}^+,\overline{x}^-)\in T_{\text{train}}} \beta_k \quad l^{\text{C}}(\mathbf{w}; \overline{p}^k, \overline{x}^+, \overline{x}^-),$$

where

$$l^{\text{C}}(f_{\mathbf{w}}; \overline{p}^k, \overline{x}^+, \overline{x}^-) = |1 - \mathbf{w} \cdot \phi(\overline{x}^+, \overline{p}^k, \overline{s}^+) + \max_{\overline{s}} \mathbf{w} \cdot \phi(\overline{x}^-, \overline{p}^k, \overline{s})|_+,$$

and $\overline{s}^+$ refers to the segmentation of $\overline{p}^k$ over $\overline{x}^+$ provided by the training data. One can check that $L^{\text{CReg}}(f_{\mathbf{w}}) \geq L^{\text{Reg}}(f_{\mathbf{w}})$, since

$$\mathbf{w} \cdot \phi(\overline{x}^+, \overline{p}^k, \overline{s}^+) \leq \max_{\overline{s}} \mathbf{w} \cdot \phi(\overline{x}^+, \overline{p}^k, \overline{s}).$$

In addition to convexity, L^{CReg} has also a computational advantage as it divides the training cost due to dynamic programming by two: the maximization over all possible segmentations should no longer be performed for each positive example $\overline{x}^+$ but only for each negative example $\overline{x}^-$.

The minimization of $L^{\text{CReg}}(f_{\mathbf{w}})$ with respect to $\mathbf{w}$ can either be performed by cutting plane methods [Tsochantaridis et al., 2005] or by *passive-aggressive* optimization [Crammer et al., 2006]. In this work, we rely on the latter solution which yields an efficient, online training procedure close to the one used in Chapter 4. This procedure learns the weight vector iteratively by visiting the training triplets of T_{train} one after another. The procedure starts with the null vector $\mathbf{w}_0 = 0$. Then, at iteration $i \geq 1$, it considers the i^{th} training triplet $(k_i, \overline{x}_i^+, \overline{x}_i^-)$ and predicts the segmentation of the negative utterance $\overline{x}^-$,

$$\overline{s}_i^- = \arg\max_{\overline{s}} \mathbf{w}_{i-1} \cdot \phi(\overline{x}_i^-, \overline{p}_i^k, \overline{s}).$$

Defining

$$\Delta\phi_i = \phi(\overline{x}_i^+, \overline{p}_i^k, \overline{s}_i^+) - \phi(\overline{x}_i^-, \overline{p}_i^k, \overline{s}_i^-),$$

the next weight vector is then selected as a trade-off between minimizing the loss over the current triplet and remaining close to the previous weight,

$$\mathbf{w}_i = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_{i-1}\|^2 + \; c\; \beta_{k_i}\; |1 - \mathbf{w} \cdot \Delta\phi_i|_+, \tag{5.3}$$

where the hyperparameter c controls this trade-off. Equation (5.3) can actually be solved in closed form [Crammer et al., 2006], yielding

$$\begin{aligned} &\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \Delta\phi_i, \\ \text{where} \quad &\alpha_i = \min\left(c\; \beta_{k_i}, \frac{|1 - \mathbf{w}_{i-1} \cdot \Delta\phi_i|_+}{\|\Delta\phi_i\|^2}\right). \end{aligned}$$

This update is referred to as *passive-aggressive*, since the algorithm *passively* keeps the previous weight ($\mathbf{w}_i = \mathbf{w}_{i-1}$) if the loss over the current training triplet is already zero ($|1 - \mathbf{w}_{i-1} \cdot \Delta\phi_i|_+ = 0$), while it *aggressively* updates it to cancel this loss otherwise. At the end of the training procedure, when all training triplets have been visited, the best weight $\mathbf{w}^*$ among $\{\mathbf{w}_0, \dots \mathbf{w}_{|T_{\text{train}}|}\}$ is selected over a set of validation triplets T_{valid}. The hyperparameter c is also selected relying on the validation data.

The pseudo-code of the algorithm is given in Algorithm 5.1. This passive aggressive approach hence requires the selection of two hyperparameters, the aggressiveness c and the effective number of iterations n, which corresponds to the index of the selected weight $\mathbf{w}^*$. These two parameters implicitly selects C in L^{CReg}, since they determine the trade-off between minimizing the training loss l^C and the regularizer. In fact, during training, the training loss is descreasing, while, at the same time an upper bound on $\|\mathbf{w}\|$ is increasing. Moreover, c influences the convergence rate, see Chapter 4 and [Crammer et al., 2006]. Hence, both n and c set the trade-off between the smoothness of the solution and the training accuracy.

Algorithm 5.1: Passive-Aggressive Training

Input: Training set T_{train}, validation set T_{valid}; parameter c;
Initialize $\mathbf{w} = 0$.
foreach $(k_i, \overline{x}_i^+, \overline{x}_i^-) \in T_{train}$ **do**
  $\overline{s}_i^- = \arg\max_{\overline{s}}\; \mathbf{w}_{i-1} \cdot \phi(\overline{p}_i^k, \overline{x}_i^-, \overline{s})$
  $\Delta\phi_i \;=\; \phi(\overline{p}_i^k, \overline{x}_i^+, \overline{s}_i^+) - \phi(\overline{p}_i^k, \overline{x}_i^-, \overline{s}_i^-)$
  if $\mathbf{w}_{i-1} \cdot \Delta\phi_i < 1$ **then**
    $\alpha_i = \min\left(c\; \beta_{k_i}, \dfrac{1 - \mathbf{w}_{i-1} \cdot \Delta\phi_i}{\|\Delta\phi_i\|^2}\right)$
    $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \cdot \Delta\phi_i$
  end
end

Output: $\mathbf{w}^*$ achieving the highest AUC over T_{valid}.

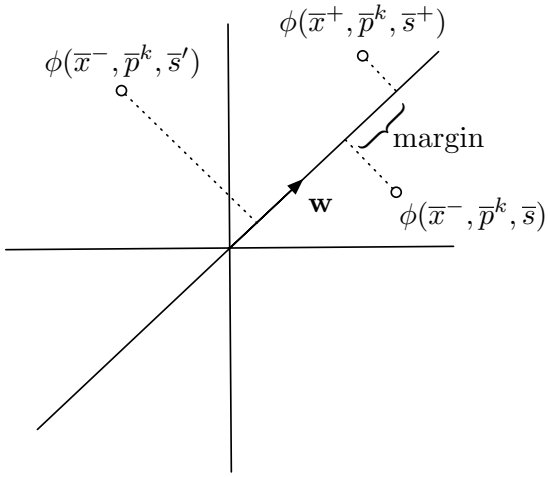

Figure 5.4. The feature vectors are ordered according to their projection onto $\mathbf{w}$. A positive example $\phi(\overline{x}^+, \overline{p}^k, \overline{s}^+)$ should appear above any negative example $\phi(\overline{x}^+, \overline{p}^k, \overline{s}), \forall \overline{s}$, with a margin of at least $\frac{1}{\|\mathbf{w}\|}$.

5.2.5 Large Margin Keyword Spotting

This section explains why the minimization of $L^{\text{Reg}}(f_{\mathbf{w}})$ corresponds to a large margin approach. $L^{\text{Reg}}(f_{\mathbf{w}})$ combines two terms, the regularizer and the loss. The loss sums $l^{\text{C}}(f_{\mathbf{w}}; \overline{p}^k, \overline{x}^+, \overline{x}^-)$ over the training triplets $(\overline{p}^k, \overline{x}^+, \overline{x}^-) \in T_{\text{train}}$. For each term, the lowest possible value $l^{\text{C}}(f_{\mathbf{w}}; \overline{p}^k, \overline{x}^+, \overline{x}^-) = 0$ is reached when,

$$\mathbf{w} \cdot \phi(\overline{x}^+, \overline{p}^k, \overline{s}^+) - \max_{\overline{s}} \mathbf{w} \cdot \phi(\overline{x}^-, \overline{p}^k, \overline{s}) > 1,$$

which is equivalent to

$$\forall \overline{s}, \quad \mathbf{w} \cdot \phi(\overline{x}^+, \overline{p}^k, \overline{s}^+) - \mathbf{w} \cdot \phi(\overline{x}^-, \overline{p}^k, \overline{s}) > 1.$$

These inequalities can be rewritten as,

$$\forall \overline{s}, \quad \mathbf{u} \cdot \phi(\overline{x}^+, \overline{p}^k, \overline{s}^+) - \mathbf{u} \cdot \phi(\overline{x}^-, \overline{p}^k, \overline{s}) > \frac{1}{\|\mathbf{w}\|}, \tag{5.4}$$

where $\mathbf{u}$ corresponds to the unit vector directing $\mathbf{w}$, i.e $\mathbf{u} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$. Hence, the loss part of our objective function ensures that the projection of the positive feature vector $\phi(\overline{x}^+, \overline{p}^k, \overline{s}^+)$ along the direction $\mathbf{u}$ is ranked above the projection of any negative feature vector $\phi(\overline{x}^-, \overline{p}^k, \overline{s})$, with a least a margin of $\frac{1}{\|\mathbf{w}\|}$, see Figure 5.4. Therefore, the regularizer $\|\mathbf{w}\|^2$ can be interpreted as a margin maximization term. The large margin framework ensures good generalization properties, as analyzed in [Vapnik, 1995]. For instance, one can remark that

the satisfaction of (5.4) implies that the score of the positive utterance $\overline{x}^+$ remains above the score of the negative one $\overline{x}^-$, even when introducing noise terms ν, ν', whose amplitudes $\|\nu\|, \|\nu'\|$ remain below $\frac{1}{2\|\mathbf{w}\|}$,

$$\forall \overline{s}, \quad \mathbf{u} \cdot (\phi(\overline{x}^+, \overline{p}^k, \overline{s}^+) + \nu) > \mathbf{u} \cdot (\phi(\overline{x}^-, \overline{p}^k, \overline{s}) + \nu').$$

5.3 Experiments and Results

We conducted two types of experiments to evaluate the proposed discriminative approach. First, we learned the parameters of our model over the training set of TIMIT, and compared its performance against an HMM baseline over the test set of TIMIT. Second, we measured the robustness of both models with respect to changing recording conditions. For this second set of experiments, we evaluated the models learned over TIMIT on Wall Street Journal (WSJ) data.

5.3.1 The TIMIT Experiments

The TIMIT corpus [Garofolo, 1993] consists in read speech from 630 American speakers, with 10 utterances per speaker. The corpus provides manually aligned phoneme and word transcriptions for each utterance. It also provides a standard split into training and testing data. From the training part of the corpus, we extract three disjoint sets consisting of 1500, 300 and 200 utterances. The first set acts as the training set of the phoneme classifier used by our fifth feature function ϕ_5. The second set acts as the training set for our discriminative keyword spotter, while the third set acts as the validation set to select the hyperparameter c and the best weight $\mathbf{w}$ seen during training. The test set is solely used for evaluation purposes.

Mel Frequency Cepstral Coefficients (MFCC), along with their first (Δ) and second derivatives ($\Delta\Delta$), are extracted every 10 ms. These features are used by our first five feature functions $\phi_1, \ldots, \phi_5$. For our fifth feature function ϕ_5, two types of phoneme classifiers are experimented, a large margin phoneme classifier and a GMM model [Bilmes, 1998]. Both classifiers are trained to predict 39 phoneme classes [Lee and Hon, 1988b] over the first part of the training set. The large margin classifier corresponds to a hierarchical classifier relying on the Gaussian kernel [Dekel et al., 2004]. This classifier exploits the dependency between phoneme classes, as formalized by the phonetic tree of American English [Rabiner and Schafer, 1978]. In this case, the margin score assigned to each frame for a given phone is used as the function g in ϕ_5, see Equation (5.2). The GMM model corresponds to a Bayes classifier combining

one GMM per class and the phoneme prior probabilities, both learned from the training data. In this case, the log posterior of a phone given the frame vector is used as the function g in ϕ_5, see Equation (5.2). The hyperparameters of both phoneme classifiers are selected to maximize the frame accuracy over part of the training data held out during parameter fitting. The discriminative keyword spotter relying on the features from the hierarchical phoneme classifier is referred as `Discriminative/Hier` in the following, while the model relying on the GMM log posteriors is referred as `Discriminative/GMM`.

We compare the results of both models against an HMM baseline, in which each phoneme is modeled with a left-right HMM of 5 emitting states. The density of each state is modeled with a 40-Gaussian GMM. Training is performed over the whole TIMIT training set. *Embedded training* is applied, i.e. after an initial training phase relying on the provided segmentation, a second training phase which dynamically determines the most likely segmentation is applied. The hyperparameters of this model (i.e. the number of states per phoneme, the number of Gaussians per state, as well as the number of Expectation-Maximization iterations) are selected to maximize the likelihood of an held-out validation set.

The phone models of the trained HMM are then used to build a keyword spotting HMM, composed of two sub-models: the keyword model and the garbage model, as illustrated on Figure 5.1. The keyword model is an HMM, which estimates the likelihood of an acoustic sequence given that the sequence represented the keyword phoneme sequence. The garbage model is an HMM composed of all phoneme HMMs fully connected to each others, which estimates the likelihood of any phoneme sequence. The overall HMM fully connects the keyword model and the garbage model. The detection of a keyword in a given utterance is performed by checking whether the Viterbi best path passes through the keyword model, as explained in Section 5.1. In this model, the keyword transition probability sets the trade-off between the true positive rate and the ROC curve can be plotted by varying this probability. This model is referred as `HMM/Viterbi` in the following. We also experiment an alternative decoding strategy, in which the system outputs the ratio of the likelihood of the acoustic sequence knowing the keyword is uttered versus the likelihood of the sequence knowing the keyword is *not* uttered, as discussed in Section 5.1. In this case, the first likelihood is determined by an HMM forcing an occurence of the keyword, and the second likelihood is determined by the garbage model, as illustrated on Figure 5.2 . This likelihood-ratio strategy is referred as `HMM/Ratio` in the following.

Table 5.1. Area Under the Curve (AUC) over the TIMIT Corpus

Model	AUC
`HMM/Viterbi`	94.2
`HMM/Ratio`	95.2
`Discriminative/GMM`	97.1
`Discriminative/Hier`	99.6

The evaluation of discriminative and HMM-based models is performed over 80 keywords, randomly selected among the words occurring in the test set of TIMIT. This random sampling of the keyword set aims at evaluating the expected performance over any keyword. For each keyword k, we consider a spotting problem, which consists of a set of positive utterances X_k^+ and a set of negative utterance X_k^-. Each positive set X_k^+ contains between 1 and 20 sequences, depending on the number of occurrences of k in the TIMIT test set. Each negative set contains 20 sequences, randomly sampled among the sequences of TIMIT that does not contain k. This setup represents an unbalanced problem, with only 10% of the sequences being labeled as positive.

Table 5.1 reports the AUC results, averaged over the 80-word test set, for the evaluated models. These results show the advantage of our approach. The two HMM based solutions are outperformed by the keyword spotters relying on our discriminative learning approach. The improvement introduced by our discriminative training algorithm can be observed when comparing the performance of `Discriminative/GMM` to the performance of the HMM spotters. In that case, both spotters rely on GMMs to estimate the frame likelihood given a phoneme class. In our case we use that probability to compute the ϕ_5 feature, while the HMM uses it as the state emission probability.

Moreover, our keyword spotter can benefit from effective non-probabilistic frame classifiers, as illustrated with `Discriminative/Hier`. This model relies on the output of a large margin frame-based classifier [Dekel et al., 2004], which yield an additional improvement compared to `Discriminative/GMM`. In order to verify whether the differences observed on averaged AUC could be due only to a few keywords, we applied the Wilcoxon test [Rice, 1995] to compare the results of both HMM approaches (`HMM/Viterbi` and `HMM/Ratio`) with the results of both discriminative approaches (`Discriminative/GMM` and `Discriminative/Hier`). At the 90% confidence level, the test rejected this hypothesis, showing that the performance gain of the discriminative approach

Table 5.2. Comparing `Discriminative/Hier` and `HMM/Ratio` for each keyword over the TIMIT corpus

Best Model	Keywords
`HMM/Ratio`	materials **(1 keyword)**
`Discriminative/Hier`	absolute admitted apartments apparently argued controlled depicts dominant drunk efficient followed freedom introduced millionaires needed obvious radiation rejected spilled street superb sympathetically weekday **(23 keywords)**
No differences	aligning anxiety bedrooms brand camera characters cleaning climates creeping crossings crushed decaying demands dressy episode everything excellent experience family firing forgiveness fulfillment functional grazing henceforth ignored illnesses imitate increasing inevitable January mutineer package paramagnetic patiently pleasant possessed pressure recriminations redecorating secularist shampooed solid spreader story strained streamlined stripped stupid surface swimming unenthusiastic unlined urethane usual walking **(56 keywords)**

is consistent over over the keyword set.

Table 5.2 further presents the per-keyword performance and compares the results of the best HMM configuration, `HMM/Ratio` to the performance of the best discriminative configuration, `Discriminative/Hier`. Out of 80 keywords, the discriminative model outperforms the HMM for 23 keywords, the HMM outperforms our solution for 1 keyword, and both models yield the same AUC for 56 keywords. The discriminative model seems to be especially advantageous for short keywords, as it outperform the HMM for most of the keywords of 5 phonemes or less (e.g. drunk, spilled, street). The 56 cases without reported differences between the models correspond to keywords which are correctly detected by both models, i.e. 100% AUC. This advocates for further comparisons over a more challenging task.

5.3.2 The WSJ Experiments

WSJ [Paul and Baker, 1992] is a large corpus of American English. It consists in read and spontaneous speech corresponding to the reading and the dictation of articles from the Wall Street Journal. In the following, we evaluate the models trained over the TIMIT dataset relying on the `si_tr_s` subset of WSJ. This

Table 5.3. Area Under the Curve (AUC) over the WSJ Corpus

Model	AUC
`HMM/Viterbi`	86.8
`HMM/Ratio`	88.4
`Discriminative/GMM`	92.2
`Discriminative/Hier`	91.4

set corresponds to the recordings of 200 speakers. Compared to TIMIT, this set introduce several variations, both regarding the type of sentences recorded and the recording conditions [Paul and Baker, 1992]. These experiments hence evaluate the robustness of the different approaches when they encounter differing conditions for training and testing. Like for TIMIT, the evaluation is performed over 80 keywords randomly selected from the corpus transcription. For each keyword k, the evaluation is performed over a set X_k^+, containing between 1 and 20 positive sequences, and a X_k^-, containing 20 randomly selected negative sequences. This setup also represents an unbalanced problem, with 27% of the sequences being labeled as positive.

Table 5.3 reports the AUC results, averaged over the 80-word test set, for the evaluated models. Overall, the results of this WSJ experiments show that the differences between the TIMIT training conditions and the WSJ testing conditions affect the performance of all models. However, the measured performance still yield acceptable performance in all cases (86.8% AUC in the worse case). Comparing the individual model performance, the WSJ results confirm the conclusions of TIMIT experiments and the discriminative spotters outperform the HMM-based alternatives. For the HMM models, `HMM/Ratio` outperforms `HMM/Viterbi` like in the TIMIT experiments. For the discriminative spotters, `Discriminative/GMM` outperforms `Discriminative/Hier`, which was not the case over TIMIT. Since these two models only differ in the frame-based classifier used as the 5^{th} feature function, this result certainly indicates that the hierarchical frame-based classifier on which `Discriminative/Hier` relies is less robust to the acoustic condition changes than the GMM alternative. Like for TIMIT, we checked whether the differences observed on the whole set could be due to a few keywords. The Wilcoxon test rejected this hypothesis at the 90% confidence level, for the 4 tests comparing `Discriminative/GMM` and `Discriminative/Hier` to `HMM/Viterbi` and `HMM/Hier`.

We further compared the best discriminative spotter, `Discriminative/GMM`,

Table 5.4. Comparing `Discriminative/GMM` and `HMM/Ratio` for each keyword over the WSJ corpus

Best Model	Keywords
`HMM/Ratio`	artificially Colorado elements Fulton itinerary longer lunchroom merchant mission multilateral narrowed outlets Owens piper replaced reward sabotaged shards spurt therefore **(20 keywords)**
`Discriminative/Hier`	Adams additions Allen Amerongen apiece buses Bushby Colombians consistently cracked dictate drop fantasy fills gross Higa historic implied interact kings list lobby lucrative measures Melbourne millions Munich nightly observance owning plus proudly queasy regency retooling Rubin scramble Seidler serving significance sluggish strengthening Sutton's tariffs Timberland today truths understands withhold Witter's **(50 keywords)**
No differences	aftershocks Americas farms Flamson hammer homosexual philosophically purchasers sinking steel-makers **(10 keywords)**

and the best HMM spotter `HMM/Ratio` over each keyword. These results are summarized in Table 5.4. Out of the 80 keywords, the discriminative model outperforms the HMM for 50 keywords, the HMM outperforms the discriminative model for 20 keywords and both models yield the same results for 10 keywords. Like for the TIMIT experiments, our model is shown to be especially advantageous for short keywords, with 5 phonemes or less (e.g. Adams, kings, serving).

Overall, the experiments over both WSJ and TIMIT highlight the advantage of our discriminative learning strategy.

5.4 Conclusions

This chapter introduced a discriminative approach to the keyword spotting problem. In this task, the model receives a keyword and a speech recording and should decide whether the keyword has been uttered in the recording. Keyword spotting corresponds to an unbalanced detection problem, since, in standard setups, most of tested utterances do not contain the targeted keyword. In that unbalanced context, the Area Under the receiver operating Curve (AUC) is generally used for evaluation. This work proposed a learning algorithm, which aims at maximizing the AUC over a set of training spotting problems.

Our strategy is based on a large margin formulation of the task, and relies on an efficient iterative training procedure. The resulting model contrasts with standard approaches based on Hidden Markov Models (HMMs), for which the training procedure does not rely on a loss directly related to the spotting task. Compared to such alternatives, our model is shown to yield significant improvements over various spotting problems on the TIMIT and the WSJ corpus. For instance, the best HMM configuration over TIMIT reaches 95.3% AUC, compared to 99.6% for the best discriminative spotter.

Several potential directions of research can be identified from this work. In its current configuration, our keyword spotter relies on the output of a pre-trained frame-based phoneme classifier. It would be of a great interest to learn the frame classifier and the keyword spotter jointly, so that all model parameters are selected to maximize the performance on the final spotting task. For that purpose, solutions based on large-margin sequence classifiers such as [Keshet et al., 2006] or [Sha and Saul, 2007] could be investigated.

Also, our work currently represents keywords as sequence of phonemes, without considering the neighboring context. Possible improvement might results from the use of phoneme in context, i.e. triphones. We hence plan to investigate on relying on triphones in a discriminative framework, and to compare the resulting model to triphone-based HMMs. More generally, our model parameterization offers greater flexibility to incorporate new features, compared to probabilistic approaches such as HMMs. Therefore, in addition to triphones, features extracted from the speaker identity, the channel characteristics or the linguistic context could possibly be included to improve performance.

Beyond the spotting of spoken keywords, this work might be extended to the field of Computer Vision, where a similar word spotting problem exists. In fact, the detection of written keywords within images is required for different applications, such as car navigation systems, advertising survey and multimedia indexing [Chen et al., 1995, 2004]. Of course, this vision problem is more challenging since the search over all possible alignments should be replaced by a search over the projections of a planar object (i.e. the surface potentially displaying the keyword) onto the image space.

Overall, this chapter shows that tasks which have not been formalized as ranking problems, such as keyword spotting, can benefit from recent learning techniques developed for retrieval rankings.

5.5 Contributions

This research on keyword spotting was performed in collaboration with Joseph Keshet. I was initially focusing on the learning objective for AUC maximization, while Joseph Keshet brought his expertise on sequence modeling. The resulting ideas and work presented in this chapter can be considered as a joint contribution. The work presented in this Chapter constitutes an extension of the conference paper [Keshet et al., 2007a]. Based on the same learning objective, a Neural Network for learning the inter-frame distance for template-based keyword detection has then been proposed [Grangier and Bengio, 2007].

◇ J. Keshet, D. Grangier, and S. Bengio. Discriminative keyword spotting. In *International Workshop on Non-LInear Speech Processing (NOLISP)*, pages 47–50, Paris, France, May 2007a.

◇ D. Grangier and S. Bengio. Learning the inter-frame distance for discriminative template-based keyword detection. In *International Conference on Speech Processing (INTERSPEECH)*, pages 902–905, Antwerp, Belgium, August 2007.

6 Conclusions and Perspectives

This last chapter concludes summarizes the thesis and delineates possible future directions of research.

6.1 General Summary

Throughout this thesis, we have explored the application of machine learning techniques to the ad-hoc retrieval problem. This task, which consists in ranking the documents of a given corpus with respect to queries, is currently receiving a growing attention from the machine learning community, mainly due to the growing web-search industry. Within this effort, our general goal was to introduce discriminative approaches to address different retrieval problems. Our methodology focused on proposing scalable learning strategies to leverage from large training sets.

The first problem that we addressed is the retrieval of text documents from text queries. Since this task is generally addressed by ranking documents according to their similarity to the query, we proposed a model to learn a function measuring text similarity from data. Specifically, we proposed to leverage from large hyperlinked corpora to infer an effective text similarity measure. Our method assumes that documents sharing hyperlinks generally share similar content and identifies a similarity measure agreeing with this assumption. Our model parameterization learns the term weighting function for bag-of-word vector with a neural network. The network is trained from the document proximity properties conveyed by the hyperlinks. In a transfer learning setup, we applied the learned similarity measure to the targeted text retrieval application and we observed that the hyperlinked training data yield significant performance improvements, compared to standard term weighting strategies. This research

has been published in [Grangier and Bengio, 2005a] and [Grangier and Bengio, 2005b].

The next work addressed the task of image retrieval from text queries. Our objective was to propose a discriminative model for this task. For that purpose, we introduced a learning procedure optimizing a criterion related to the ranking performance over a set of training queries and images. Our learning approach builds upon recent work on the online learning of kernel-based classifiers. This results in an efficient, scalable algorithm, which can benefit from recent kernels developed for image comparison. Our method contrasts with state-of-the-art approaches that mostly rely on generative annotation models which learns a joint distribution over text and visual features through likelihood maximization. Compared to such models, we showed that our method offers both greater scalability and higher performance over benchmark data. Different aspects of this work have been published separately [Grangier et al., 2006a,b; Grangier and Bengio, 2006, 2008].

In a third part of the thesis, we addressed the task of keyword spotting, i.e. the detection of keywords in speech utterances. Although keyword spotting is not formally a ranking problem, we proposed to use a ranking objective on this task. This objective is similar to the one used in our image retrieval work and states that, when ordering speech utterances according to the spotter confidence, the utterances containing the keyword should appear above the others. Interestingly, this formulation of the problem actually yields a large margin model optimizing the area under the receiver operating curve, the most common measure used to evaluate keyword spotters. From a performance per-

---

◇ D. Grangier and S. Bengio. Inferring document similarity from hyperlinks. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 359–360, Bremen, Germany, November 2005a.

◇ D. Grangier and S. Bengio. Exploiting hyperlinks to learn a retrieval model. In *NIPS Workshop on Learning to Rank*, pages 12–17, Whistler, Canada, December 2005b.

◇ D. Grangier, F. Monay, and S. Bengio. Learning to retrieve images from text queries with a discriminative model. In *International Workshop on Adaptive Multimedia Retrieval (AMR)*, pages 42–56, Geneva, Switzerland, July 2006a.

◇ D. Grangier, F. Monay, and S. Bengio. A discriminative approach for the retrieval of images from text queries. In *European Conference on Machine Learning (ECML)*, pages 162–173, Berlin, Germany, September 2006b.

◇ D. Grangier and S. Bengio. A neural network to retrieve images from text queries. In *International Conference on Artificial Neural Networks (ICANN)*, volume 2, pages 24–34, Athens, Greece, September 2006.

◇ D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*, 2008. (in press).

spective, the proposed model has shown to be effective compared to generative alternatives based on Hidden-Markov Models. This work has resulted in the following publications, [Keshet et al., 2007a; Grangier and Bengio, 2007].

Over this different retrieval applications, we have illustrated the advantage of relying on learning objectives closely related to the final task. The different proposed models have shown significant performance gains, compared to approaches relying on intermediate goals. This work also advocates for the use of simple, efficient learning algorithms that can leverage from large training sets, since, in machine learning, "nothing is as valuable as data, except maybe more data". Of course, this thesis is only a glimpse into the potential applications of machine learning to retrieval problems. In the following, we outline potential directions of research that our work could initiate.

6.2 Potential Future Work

Several possible directions of research arise from this thesis. In the following, we describe the ones we consider the most promising.

Our work on text retrieval focused on a transfer learning setup and proposed to build a better measure of text similarity from the semantic proximity information conveyed by hyperlinks. Of course, hyperlinks are not the only source of such information. In the recent years, approaches to exploit the relationships between translations of the same document [Vinokourov et al., 2003], or between the successive paragraphs of a text [Keller and Bengio, 2006] have been proposed. Building upon this research, it would be of a great interest to introduce a learning algorithm that could leverage from all these sources in a single unified framework.

Another interesting aspect deserving further investigations is user-feedback and personalization. The interactions of the users with a retrieval system provides valuable data to improve the ranker and to adapt it to different kinds of user profiles. The online algorithms proposed in this thesis are especially adapted for this truly online setup. Of course, online adaptation presents challenging problems, since user feedback is typically noisy and biased towards the existing ranker. A possible solution to overcome such difficulties could rely on

---

◇ J. Keshet, D. Grangier, and S. Bengio. Discriminative keyword spotting. In *International Workshop on Non-LInear Speech Processing (NOLISP)*, pages 47–50, Paris, France, May 2007a.

◇ D. Grangier and S. Bengio. Learning the inter-frame distance for discriminative template-based keyword detection. In *International Conference on Speech Processing (INTERSPEECH)*, pages 902–905, Antwerp, Belgium, August 2007.

techniques developed for reinforcement learning [Sutton and Barto, 1998], as this field always deals with such a bias.

Throughout this thesis, our approach to the ranking problem focused on the pairwise loss, stating that the score of the ranking function should be higher for relevant items than for non-relevant ones. Other loss functions have been proposed to account for gradual relevance levels and to emphasize the top position of the ranking [Jarvelin and Kekalainen, 2000]. These functions are of a great interest for applications such as web search, where the very first positions are of crucial importance. The optimization of such losses is typically more costly than the pairwise loss, since their gradient cannot be computed without ranking all the corpus documents [Burges et al., 2006; Joachims, 2005]. In order to derive efficient online learning strategies for minimizing these losses, it would be necessary to investigate on approximations or bounds which can be computed from only a small fraction of the corpus documents. Such approximations or bounds would actually represent a great advance towards the use of machine learning as the main tool to identifying effective rankers for large scale datasets.

Generally, the growing interest of the machine learning community for retrieval applications opens promising perspectives for many further research opportunities. Moreover, the web search industry is currently radically transforming the retrieval field, providing a seemingly unlimited source of data. Unlimited ? It is worth remembering when machine learning dissertation used to define the objective of this field as “identifying an effective function from *limited* training data”, see Chapter 1...

A Appendix

A.1 Regularization in Passive-Aggressive through Early Stopping

This appendix shows that an upper bound on the norm $\|\mathbf{w}^i\|$ grows with the number of iterations i of the Passive-Aggressive algorithm, described in Chapter 4, Section 4.2.4. Precisely, it shows that $\|\mathbf{w}^i\| \leq 2\ c\ \rho\ i$, where ρ corresponds to the radius of the training data $\rho = \max_{(q,p)\in D_{train}} \gamma(q,p)$.

The proof, inspired from [Collobert and Bengio, 2004], is conducted by induction over the iteration i. At the first iteration, the property is satisfied, since $\mathbf{w}^0 = 0$. The update rule of $\mathbf{w}^t$ also preserves the property. If we assume the property to be verified at iteration $i-1$, i.e. $\|\mathbf{w}^{i-1}\| \leq 2\ c\ \rho\ (i-1)$, we have $\|\mathbf{w}^i\| \leq 2\ c\ \rho\ (i-1) + \|\tau_i v^i\|$, according to the update rule (4.10). By definition, τ_i is positive and smaller than c and hence $\|\mathbf{w}^i\| \leq 2\,c\,\rho\,(i-1)+c\|v^i\|$. Furthermore, v^i is defined as $\gamma(q^i, p^{i+}) - \gamma(q^i, p^{i-})$, which implies that $\|v^i\| \leq 2\rho$. Consequently, $\|\mathbf{w}^i\| \leq 2\ c\ \rho\ i$. This concludes the proof.

A.2 Dynamic Programming Procedure for the Discriminative Spotter

In this appendix, we present a reccursive procedure to efficiently compute the confidence score of our keyword spotter described in Chapter 5, Section 5.2.3, i.e.

$$f_{\mathbf{w}}(\overline{x}, \overline{p}^k) = \max_{\overline{s}} \mathbf{w} \cdot \phi(\overline{x}, \overline{p}^k, \overline{s}). \tag{A.1}$$

This procedure relies on dynamic programming, similarly to Viterbi decoding for Hidden Markov Models. To introduce our approach, we first require a few definitions. For all i and any $s_{i-1} < s_i < s_{i+1}$, we introduce the vector $h(i; s_{i-1}, s_i, s_{i+1})$ of $\mathbb{R}^7$ in which each component is defined as

$$\forall j = 1,2,3,4, \quad h^i_j(i; s_{i-1}, s_i, s_{i+1}) \quad = \quad \frac{1}{L}\, d(x_{s_i - j}, x_{s_i + j}),$$

and

$$
\begin{aligned}
h_5(i; s_{i-1}, s_i, s_{i+1}) &= \frac{1}{L} \sum_{j=s_i}^{s_{i+1}-1} \frac{1}{s_{i+1} - s_i}\, g(x_j, p_i), \\
h_6(i; s_{i-1}, s_i, s_{i+1}) &= \frac{1}{L} \log \mathcal{N}(s_{i+1} - s_i; \mu_{p_i}, \sigma_{p_i}), \\
h_7(i; s_{i-1}, s_i, s_{i+1}) &= \begin{cases} 0 & \text{if } i = 1, \\ \frac{1}{L}\,(r_i - r_{i-1})^2 & \text{otherwise.} \end{cases}
\end{aligned}
$$

where $r_i = \frac{s_{i+1}-s_i}{\mu_{p_i}}$. According to the definition of ϕ, we have

$$
\forall \overline{s}, \quad \phi(\overline{x}, \overline{p}^k, \overline{s}) = \sum_{i=1}^{L} h(i; s_{i-1}, s_i, s_{i+1}). \tag{A.2}
$$

We further introduce,

$$
\forall l > 1, \quad \mathrm{H}(l; s_1, s_l, s_{l+1}) = \max_{s_2, \ldots, s_{l-1}} \sum_{i=1}^{l} \mathbf{w} \cdot h(i; s_{i-1}, s_i, s_{i+1}).
$$

We can notice that, according to Equation (A.1) and Equation (A.2),

$$
f_{\mathbf{w}}(\overline{x}, \overline{p}^k) = \max_{s_1, s_L, s_{L+1}} \mathrm{H}(L; s_1, s_L, s_{L+1}) \tag{A.3}
$$

We can now introduce the core of our recursive algorithm: for any $l > 2$,

$$
\begin{aligned}
&\mathrm{H}(l; s_1, s_l, s_{l+1}) \\
&= \max_{s_{l-1}} \left\{ \max_{s_2, \ldots, s_{l-2}} \sum_{i=1}^{l} \mathbf{w} \cdot h(i; s_{i-1}, s_i, s_{i+1}) \right\} \\
&= \max_{s_{l-1}} \left\{ \max_{s_2, \ldots, s_{l-2}} \left\{ \sum_{i=1}^{l-1} \mathbf{w} \cdot h(i; s_{i-1}, s_i, s_{i+1}) \right\} + \mathbf{w} \cdot h(l; s_{l-1}, s_l, s_{l+1}) \right\} \\
&= \max_{s_{l-1}} \{ \mathrm{H}(l-1; s_1, s_{l-1}, s_l) + \mathbf{w} \cdot h(l; s_{l-1}, s_l, s_{l+1}) \}
\end{aligned}
$$

This reccursion suggests Algorithm A.1 for solving the maximization in Equation (A.1). In this algorithm, we introduce range lists, which allows to simplify the notation for iteration indexes. These ranges are computed from $|\overline{x}|$ the acoustic sequence length, L the number of phoneme in the targeted keyword, minp the minimum allowed duration for a phoneme and maxp the minimum allowed duration for a phoneme. $\mathrm{range}_{\mathrm{start}}$ denotes the allowed start points of the first phoneme, between 1 and $|\overline{x}| - L\mathrm{minp} + 1$, where $L\mathrm{minp}$ corresponds to the minimum keyword duration. $\mathrm{range}(s_i)$ denotes the allowed end points for a phoneme started at s_i, between $s_i + \mathrm{minp}$ and $s_i + \mathrm{maxp}$.

Algorithm A.1: Dynamic Programming Procedure

Input: acoustic sequence $\overline{x}$ and keyword phoneme sequence $\overline{p}$

Step 1. Precomputation of H for the whole sequence.

foreach $s_1 \in \text{range}_{\text{start}}$ **do**
  foreach $s_2 \in \text{range}(s_1)$ **do**
    $\text{H}(1; s_1, s_1, s_2) = \mathbf{w} \cdot h(i; s_0, s_1, s_2)$
  end
  foreach $i \in \{2, \ldots, L\}$ **do**
    foreach $s_i \in \text{range}(s_{i-1})$ **do**
      foreach $s_{i+1} \in \text{range}(s_i)$ **do**
        $\text{H}(i; s_1, s_i, s_{i+1}) =$
        $\max_{s_{i-1}} \{\text{H}(i-1; s_1, s_{i-1}, s_i) + \mathbf{w} \cdot h(i; s_{i-1}, s_i, s_{i+1})\}$
      end
    end
  end
end

Step 2. Compute $f_{\mathbf{w}}(\overline{x}, \overline{p}^k)$ from Equation (A.3).

Output: confidence score $f_{\mathbf{w}}(\overline{x}, \overline{p}^k)$

The complexity of this algorithm is dominated by Step 1, and, more specifically, by the most inner loop of step 1. Its complexity is $\mathcal{O}(|x|\ L\ d^3)$, where d refers to $\text{maxp} - \text{minp} + 1$. This complexity is the product of the number of iterations of the loop over s_1, i, s_i and s_{i+1}, multiplied by the cost of the maximization over s_{i-1}.

Bibliography

A. Amir, G. Iyengar, J. Argillander, M. Campbell, A. Haubold, S. Ebadollahi, F. Kang, M. R. Naphade, A. Natsev, J. R. Smith, J. Tesic, and T. Volkmer. IBM research TRECVID-2005 video retrieval system. In *TREC Video Workshop*, pages 51–68, Gaithersburg, MD, USA, November 2005.

J.A. Aslam and E. Yilmaz. A geometric interpretation and analysis of r-precision. In *Conference on Information and Knowledge Management (CIKM)*, pages 664–671, Bremen, Germany, November 2005.

H. Attias, J. Platt, A. Acero, and L. Deng. Speech denoising and dereverberation using probabilistic models. In *Neural Information Processing Systems (NIPS)*, pages 758–764, Vancouver, Canada, December 2001.

B., J. Lu, and G. Huang. A constrained non-negative matrix factorization in information retrieval. In *Conference on Information Reuse and Integration*, pages 273–277, Las Vegas, NV, USA, October 2003.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, Harlow, England, 1999.

L. R. Bahl, P. F. Brown, P. de Souza, and R. L. Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 49–52, Tokyo, Japan, April 1986.

K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *International Conference on Computer Vision (ICCV)*, pages 408–415, Vancouver, Canada, July 2001.

K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research (JMLR)*, 3:1107–1135, 2003.

Y. Benayed, D. Fohr, J. P. Haton, and G. Chollet. Confidence measures for keyword spotting using support vector machines. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 588–591, Hong Kong, China, April 2003.

J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report TR-97-021, International Computer Science Institute, Berkeley, CA, USA, 1998.

C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, London, England, 1995.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, Berlin, Germany, 2006.

D. M. Blei and J. D. Lafferty. Correlated topic models. In *Neural Information Processing Systems (NIPS)*, pages 774–781, Vancouver, Canada, December 2005.

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

J. M. Boite, H. Bourlard, B. D'hoore, and M. Haesen. Keyword recognition using template concatenation. In *European Conference on Speech and Communication Technologies (EUROSPEECH)*, pages 1273–1276, Berlin, Germany, September 1993.

S. Boughorbel, J.P. Tarel, and F. Fleuret. Non-mercer kernels for svm object recognition. In *British Machine Vision Conference*, pages 137 – 146, London, UK, September 2004.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

J. S. Bridle. An efficient elastic-template method for detecting given words in running speech. In *British Acoustic Society Meeting*, pages 1–4, London, UK, April 1973.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *International World Wide Web Conference*, pages 107–117, Brisbane, Australian, April 1998.

K. Brinker and E. Huellermeier. Calibrated label-ranking. In *NIPS Workshop on Learning to Rank*, pages 5–10, Whistler, Canada, December 2005.

C. Buckley and E.M. Voorhees. Evaluating evaluation measure stability. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 33–40, Athens, Greece, July 2000.

H. Bunke and W. P. Wang. *Handbook of Character Recognition and Document Image Analysis*. World Scientific Publishing Company, Singapore, 1997.

C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, pages 89–96, Bonn, Germany, August 2005.

C.J.C. Burges, R. Ragno, and Q.V. Le. Learning to rank with non-smooth cost functions. In *Neural Information Processing Systems (NIPS)*, pages 193–200, Vancouver, Canada, December 2006.

G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 163–168, San Diego, CA, USA, June 2005.

E. Chang. *Improving Word Spotting Performance with Limited Training Data*. PhD thesis, Massachusetts Institute of Technology (MIT), 1995.

S.-F. Chang, W. Hsu, W. Jiang, L. Kennedy, D. Xu, A. Yanagawa, and E. Zavesky. Trecvid-2006 video search and high-level feature extraction. In *TREC Video Workshop*, Gaithersburg, MD, USA, November 2006.

D. Chen, J-M. Odobez, and H. Bourlard. Text detection and recognition in images and video frames. *Pattern Recognition*, 37(3):595–608, 2004.

F.R. Chen, L.D. Wilcox, and D.S. Bloomberg. A comparison of discrete and continuous hidden markov models for phrase spotting in text images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 398–402, Montreal, Canada, August 1995.

D. Cohn and T. Hofmann. The missing link a probabilistic model of document content and hypertext connectivity. In *Neural Information Processing Systems (NIPS)*, pages 430–436, Vancouver, Canada, December 2000.

R. Collobert and S. Bengio. Links between perceptrons, MLPs and SVMs. In *International Conference on Machine Learning (ICML)*, pages 23–30, Banff, Canada, July 2004.

W. S. Cooper. Getting beyond boole. *Information Processing and Management*, 24(3):243 – 248, 1988.

Corbis. Corbis stock photography. URL http://www.corbis.com.

Corel. Corel stock photography data. URL http://www.emsps.com/photocd/corelcds.htm.

C. Cortes and M. Mohri. Confidence intervals for the area under the roc curve. In *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2004.

C. Cortes and M. Mohri. AUC optimization vs. error rate minimization. In *Neural Information Processing Systems (NIPS)*, pages 23–30, Vancouver, Canada, December 2003.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research (JMLR)*, 7:551–585, 2006.

B. D. Davison. Topical locality in the web. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 272 – 279, Athens, Greece, July 2000.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 6(41):391–407, 1990.

O. Dekel, J. Keshet, and Y. Singer. An online algorithm for hierarchical phoneme classification. In *Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*, pages 146–158, Martigny, Switzerland, June 2004.

S. Dumais. Latent semantic indexing (LSI): TREC-3 report. In *NIST Text Retrieval Conference (TREC)*, pages 219–230, Seattle, WA, USA, July 1995.

P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European Conference on Computer Vision (ECCV)*, pages 97–112, Copenhagen, Denmark, May 2002.

J. Eichhorn and O. Chapelle. Object categorization with SVM: Kernels for local features. Technical Report 137, Max Planck Institute, 2004.

S.L. Feng and V. Lavrenko R. Manmatha. Multiple bernoulli relevance models for image and video annotation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1002–1009, Washigton, DC, USA, June 2004.

Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research (JMLR)*, 4:933–969, 2003.

J. Garofolo, J. Lard, and E. Voorhees. Overview of the TREC-9 spoken document retrieval track. In *NIST Text Retrieval Conference (TREC)*, Gaithersburg, MD, USA, November 2000.

J. S. Garofolo. TIMIT acoustic-phonetic continuous speech corpus. Technical Report LDC93S1, Linguistic Data Consortium, Philadelphia, PA, USA, 1993.

J.S. Garofolo, G.P. Auzanne, and E.M. Voorhees. The TREC SDR track: A success story. In *NIST Text Retrieval Conference (TREC)*, pages 107–129, Gaithersburg, MD, USA, November 1999.

Google Book Search. Google book search. URL http://books.google.com.

Y. Grandvalet, J. Mariethoz, and S. Bengio. A probabilistic interpretation of SVMs with an application to unbalanced classification. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2005.

D. Grangier and S. Bengio. Inferring document similarity from hyperlinks. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 359–360, Bremen, Germany, November 2005a.

D. Grangier and S. Bengio. Exploiting hyperlinks to learn a retrieval model. In *NIPS Workshop on Learning to Rank*, pages 12–17, Whistler, Canada, December 2005b.

D. Grangier and S. Bengio. A neural network to retrieve images from text queries. In *International Conference on Artificial Neural Networks (ICANN)*, volume 2, pages 24–34, Athens, Greece, September 2006.

D. Grangier and S. Bengio. Learning the inter-frame distance for discriminative template-based keyword detection. In *International Conference on Speech Processing (INTERSPEECH)*, pages 902–905, Antwerp, Belgium, August 2007.

D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*, 2008. (in press).

D. Grangier, F. Monay, and S. Bengio. Learning to retrieve images from text queries with a discriminative model. In *International Workshop on Adaptive Multimedia Retrieval (AMR)*, pages 42–56, Geneva, Switzerland, July 2006a.

D. Grangier, F. Monay, and S. Bengio. A discriminative approach for the retrieval of images from text queries. In *European Conference on Machine Learning (ECML)*, pages 162–173, Berlin, Germany, September 2006b.

D. Grossman and O. Frieder. *Information Retrieval: Algorithms and Heuristics*. Springer, Berlin, Germany, 2004.

A. Gruber, M. Rosen-Zvi, and Y. Weiss. Hidden topic markov models. In *Conference on Artificial Intelligence and Statistics (AISTAT)*, San Juan, Puerto Rico, March 2007.

D. Harman. Overview of the second text retrieval conference. In *NIST Text Retrieval Conference (TREC)*, pages 1–21, Gaithersburg, MD, USA, November 1993.

D. Harman, E. A. Fox, R. A. Baeza-Yates, and W. C. Lee. Inverted files. In W. F. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, pages 28–43. Prentice Hall, 1992.

V. Hatzivassiloglou, J. L. Klavans, and E. Eskin. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 203–212, New Brunswick, NJ, June 1999.

R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In P. J. Bartlett, B. Schoelkopf, and D. Schuurmans,

editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, Cambridge, MA, 2000.

A. L. Higgins and R. E. Wohlford. Keyword recognition using template concatenation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1233–1236, Tampa, FL, USA, March 1985.

T. Hofmann. personal communication, 2005.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 41–48, Athens, Greece, July 2000.

T. Jebara and R. Kondor. Bhattacharyya and expected likelihood kernels. In *Conference on Learning Theory*, pages 57–71, Washignthon, DC, USA, August 2003.

F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, USA, 1998.

J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *International Conference on Image and Video Retrieval*, pages 24–32, Dublin, Ireland, July 2004.

J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 119–126, Toronto, Canada, July 2003.

T. Joachims. A support vector method for multivariate performance measures. In *International Conference on Machine Learning (ICML)*, pages 377 – 384, Bonn, Germany, August 2005.

T. Joachims. Optimizing search engines using clickthrough data. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, Edmonton, Canada, July 2002.

T. Joachims. Making large-scale support vector machine learning practical. In A. Smola B. Scholkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*, pages 41–56. MIT Press, Cambridge, MA, 1998.

J. Junkawitsch, G. Ruske, and H. Hoege. Efficient methods in detecting keywords in continuous speech. In *European Conference on Speech and Communication Technologies (EUROSPEECH)*, pages 259–262, Rhodes, Greece, September 1997.

T. Kawabata, T. Hanazawa, and K. Shikano. Word spotting method based on hmm phoneme recognition. *Journal of the Acoustical Society of America (JASA)*, 1(84):62, 1988.

M. Keller and S. Bengio. A multi-task learning approach to document representation using unlabeled data. Technical Report 44, IDIAP Research Institute, 2006.

J. Keshet, S. Shalev-Shwartz, S. Begio, Y. Singer, and D. Chazan. Discriminative kernel-based phoneme sequence recognition. In *International Conference on Spoken Language Processing (INTERSPEECH)*, Pittsburg, PA, USA, September 2006.

J. Keshet, D. Grangier, and S. Bengio. Discriminative keyword spotting. In *International Workshop on Non-LInear Speech Processing (NOLISP)*, pages 47–50, Paris, France, May 2007a.

J. Keshet, S. Shalev-Shwartz, Y. Singer, and Dan Chazan. Large margin algorithm for speech and audio segmentation. *IEEE Transactions on Audio, Speech and Language (TASL)*, 8(15):2373 – 2382, 2007b.

T.G. Kolda and D. P. O'Leary. A semi-discrete matrix decomposition for latent semantic indexing information retrieval. *ACM Transactions on Information Systems*, 16(4):322–346, 2004.

R. Kondor and T. Jebara. A kernel between bags of vectors. In *International Conference on Machine Learning (ICML)*, pages 361–368, Washington, DC, USA, August 2003.

S. N. Kramer. *History Begins at Sumer: Thirty-Nine Firsts in Recorded History.* Thames & Hudson, London, UK, 1958.

F. W. Lancaster. *Information retrieval systems: Characteristics, testing and evaluation.* Wiley, New York, NY, USA, 1979.

V. Lavrenko, M. Choquette, and W. B. Croft. Cross-lingual relevance models. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 175–182, Tampere, Finland, August 2002.

Q. Le and A. J. Smola. Direct optimization of ranking measures. *Journal of Machine Learning Research (JMLR)*, 2007. (submitted).

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998a.

Y. LeCun, L. Bottou, G. B. Orr, and K. R. Mueller. Efficient backprop. In G. B Orr and K. R. Mueller, editors, *Neural Networks: Trick of the Trade*, chapter 1, pages 9–50. Springer, 1998b.

K. F. Lee and H. F. Hon. Large-vocabulary speaker-independent continuous speech recognition using HMM. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 160–166, New York, NY, USA, April 1988a.

K. F. Lee and H. W. Hon. Speaker independent phone recognition using hidden markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing (TASSP)*, 11(37):1641–1648, 1988b.

Y. Li and J. Shawe-Taylor. Advanced learning algorithms for cross-language patent retrieval and classification. *Information Processing and Management*, 43(5):1183–1199, 2007.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 3(2):159–165, 1958.

Siwei Lyu. Kernels for unordered sets: the gaussian mixture approach. In *European Conference on Machine Learning (ECML)*, pages 255–267, Porto, Portugal, October 2005.

J. Mamou, B. Ramabhadran, and O. Siohan. Vocabulary independent spoken term detection. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 615–622, Amsterdam, Netherland, July 2007.

T. Mandl. Tolerant information retrieval with backpropagation networks. *Neural Computing and Applications*, 9(4):280–289, 2000.

H.B. Mann and D.R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18 (1):50–60, 1947.

B. Moghaddam and Y. Ming-Hsuan. Learning gender with support faces. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24 (5):707–711, 2002.

F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: constraining the latent space. In *ACM Multimedia*, pages 348–351, New York, NY, USA, October 2004.

H. Mueller, S. Marchand-Maillet, and T. Pun. The truth about corel: Evaluation in image retrieval. In *International Conference on Image and Video Retrieval*, pages 38–49, London, UK, July 2002.

M.R. Naphade. On supervision and statistical learning for semantic multimedia analysis. *Journal of Visual Communication and Image Representation*, 15 (3):348–369, 2004.

T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7):971–987, 2002.

L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

J. Y. Pan, H. J. Yang, P. Duygulu, and C. Faloutsos. Automatic image captioning. In *International Conference on Multimedia and Expo (ICME)*, pages 1987–1990, Taipei, Taiwan, June 2004.

D.B. Paul and J.M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Human Language Technology Conference (HLT)*, pages 357–362, New York, NY, USA, 1992.

M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

P. Quelhas, F. Monay, J. M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. J. Van Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision (ICCV)*, pages 883–890, Beijing, China, October 2005.

L. Rabiner and B.H. Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

L. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1978.

R. Kumar Rajendran and C. Shih-Fu. Image retrieval with sketches and compositions. In *International Conference on Multimedia and Expo (ICME)*, pages 717–720, New York City, NY, USA, July 2000.

A. Rakotomamonjy. Optimizing AUC with support vector machine. In *European Conference on Artificial Intelligence, Workshop on ROC Curve*, pages 71–80, Pisa, Italy, September 2004.

J.A. Rice. *Rice, Mathematical Statistics and Data Analysis*. Duxbury Press, Belmont, CA, USA, 1995.

M. Richardson and P. Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2002.

S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *NIST Text Retrieval Conference (TREC)*, pages 109–126, Gaithersburg, MD, USA, November 1994.

R. C. Rose and D. B. Paul. A hidden markov model based keyword recognition system. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 29–32, Albuquerque, NM, USA., April 1990.

S. Rosset, J. Zhu, and T. Hastie. Margin maximizing loss functions. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2004.

G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

E. D. Sandness and I. Lee Hetherington. Keyword-based discriminative training of acoustic models. In *International Conference on Spoken Language Processing (ICSLP)*, volume 3, pages 135–138, Beijing, China, October 2000.

M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2003.

F. Sha and L. Saul. Large margin HMMs for automatic speech recognition. In *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2007.

J. Shawe-Taylor and N. Cristianini. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, UK, 2000.

J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV)*, pages 370–377, Beijing, China, October 2005.

A.F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *ACM Workshop on Multimedia Information Retrieval (MIR)*, pages 321–330, Santa Barbara, CA, USA, October 2006.

A.W.M Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(12):1349–1380, 2000.

R. A. Sukkar, A. R. Seltur, M. G. Rahim, and C. H. Lee. Utterance verification of keyword strings using word-based minimum verification error training. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 518–521, Atlanta, GA, USA, May 1996.

R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1998.

M. Szummer and R. W. Picard. Indoor-outdoor image classication. In *Workshop on Content-based Access of Image and Video Databases*, pages 42–51, Bombay, India, January 1998.

V. Takala, T. Ahonen, and M. Pietikainen. Block-based methods for image retrieval using local binary patterns. In *Scandinavian Conference on Image Analysis (SCIA)*, pages 882–891, Joensuu, Finland, June 2005.

K. Tieu and P. Viola. Boosting image retrieval. *International Journal of Computer Vision (IJCV)*, 56(1):17 – 36, 2004.

TREC. Text retrieval conference datasets. URL http://trec.nist.gov.

L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005.

S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita. Dimensionality reduction using non-negative matrix factorization for information retrieval. In *Conference on Systems, Man, and Cybernetics*, pages 960–965, Tucson, AZ, USA, October 2001.

A. Vailaya, A. Jain, and H. J. Zhang. On image classification: city vs. landscape. In *Workshop on Content-based Access of Image and Video Libraries*, pages 3–8, Bombay, India, January 1998.

C. J. van Rijsbergen. *Information Retrieval*. Butterworth, London, UK, 1979.

V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, Berlin, Germany, 1982.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, Berlin, Germany, 1995.

A. Vinciarelli. Effect of recognition errors on information retrieval performance. In *International Workshop on Frontiers in Handwriting Recognition*, pages 275 – 279, Tokyo, Japan, October 2004.

A. Vinokourov, J. Shawe-Taylor, and N. Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2003.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, Kauai, HI, USA, December 2001.

J. Vogel and B. Schiele. Natural scene retrieval based on a semantic modeling step. In *International Conference on Image and Video Retrieval*, pages 207–215, Dublin, Ireland, July 2004.

E. Voorhees. Overview of TREC 2006. In *NIST Text Retrieval Conference (TREC)*, pages 1–16, Gaithersburg, MD, USA, November 2006.

E. Voorhees. Overview of the ninth text retrieval conference. In *NIST Text Retrieval Conference (TREC)*, pages 1–13, Gaithersburg, MD, USA, November 2000.

E. M. Voorhees. Evaluation by highly relevant documents. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 74 – 79, New Orleans, LA, USA, September 2001.

C. Wallraven and B. Caputo. Recognition with local features: the kernel recipe. In *International Conference on Computer Vision (ICCV)*, pages 257–264, Nice, France, October 2003.

M. Weintraub. Keyword-spotting using SRI's DECIPHER large-vocabularyspeech-recognition system. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 463–466, Minneapolis, MN, USA, May 1993.

M. Weintraub. LVCSR log-likelihood ratio scoring for keyword spotting. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 297–300, Detroit, MI, USA, May 1995.

M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke. Neural-network based measures of confidence for word recognition. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 887–890, Munich, Germany, April 1997.

Wikipedia. Wikipedia, the free encyclopedia. URL http://www.wikipedia.org.

F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing (TASSP)*, 38 (11):1870–1878, 1990.

I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes*. Morgan Kaufmann, San Francisco, CA, USA, 1999.

YouTube. Youtube. URL http://www.youtube.com.

Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 271–278, Amsterdam, Netherland, July 2007.

Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A general boosting method and its application to learning ranking functions for

web search. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2007.

David Grangier

Pré Fleury 3
CH-1006 Lausanne, Switzerland

Email: david.grangier (at) idiap (dot) ch

French Citizen
French and English Speaker

Researcher in Machine Learning

Education

10/2003 - **Ecole Polytechnique Fédérale de Lausanne (EPFL)**, Switzerland
Doctoral Studies in Machine Learning
Supervised by Samy Bengio.

09/2002 - 09/2003 **Nice Sophia Antipolis University**, France
Diplôme d'Etudes Appronfondies (M. Sc) Network and Distributed Systems
Graduated with Mention Bien (= 70% of maximum grade).

03/2002 - 09/2003 **Eurecom Institute**, Sophia Antipolis, France
Postgraduate Studies in Multimedia Communications
Specialization in Speech Processing with Christian Wellekens
Graduated with Hitachi Distinction (Best project award).

09/2000 - 03/2002 **ENST Bretagne**, Brest, France
Bachelor of Telecommunications Engineering
Graduated with rank 19 out of 214.

Professional Experience

06/2007 - 12/2007 **Google Inc.**, Internet Search and Online Advertising, USA
Research Intern: Machine Learning for Image Search
Supervised by Samy Bengio.

03/2003 - 03/2008 **IDIAP** Research Institute, Switzerland
Research Assistant: Machine Learning for Information Retrieval
Supervised by Hervé Bourlard and Alessandro Vinciarelli.

06/2002 - 09/2002 **Silogic SA**, IT Service Consultancy, France
Intern: Research on Speaker Identification for Embedded Applications
Supervised by Eric Bréhault.

Technical Knowledge

Machine Learning	Neural Networks and Kernel Machines Implementation
Multimedia	Speech Processing, Image Analysis, Information Retrieval Systems
Telecom. & Networks	Signal Processing, IP Networks and Internet Protocols
Computer Sc.	C/C++, Python, Perl, Unix Shell, Visual Basic

Publications

Peer Reviewed Journals

[1] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*, 2008. (in press).

Peer Reviewed Conferences

[2] D. Grangier and S. Bengio. Learning the inter-frame distance for discriminative template-based keyword detection. In *International Conference on Speech Processing (INTERSPEECH)*, pages 902–905, Antwerp, Belgium, August 2007.

[3] D. Grangier and S. Bengio. A neural network to retrieve images from text queries. In *International Conference on Artificial Neural Networks (ICANN)*, volume 2, pages 24–34, Athens, Greece, September 2006.

[4] D. Grangier, F. Monay, and S. Bengio. A discriminative approach for the retrieval of images from text queries. In *European Conference on Machine Learning (ECML)*, pages 162–173, Berlin, Germany, September 2006.

[5] D. Grangier and S. Bengio. Inferring document similarity from hyperlinks. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 359–360, Bremen, Germany, November 2005.

[6] D. Grangier and A. Vinciarelli. Effect of segmentation method on video retrieval performance. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 5–8, Amsterdam, The Netherlands, July 2005.

Research Reports and Workshops

[7] J. Keshet, D. Grangier, and S. Bengio. Discriminative keyword spotting. In *International Workshop on Non-LInear Speech Processing (NOLISP)*, pages 47–50, Paris, France, May 2007.

[8] D. Grangier, F. Monay, and S. Bengio. Learning to retrieve images from text queries with a discriminative model. In *International Workshop on Adaptive Multimedia Retrieval (AMR)*, pages 42–56, Geneva, Switzerland, July 2006.

[9] D. Grangier and S. Bengio. A discriminative decoder for the recognition of phoneme sequences. IDIAP-RR 67, IDIAP, 2005.

[10] D. Grangier and S. Bengio. Exploiting hyperlinks to learn a retrieval model. In *NIPS Workshop on Learning to Rank*, pages 12–17, Whistler, Canada, December 2005.

[11] D. Grangier and A. Vinciarelli. Effect of recognition errors on text clustering. IDIAP-RR 82, IDIAP, 2004.

[12] David Grangier, Alessandro Vinciarelli, and Herve Bourlard. Information retrieval on noisy text. IDIAP-COM 08, IDIAP, 2003.

Publications available at `www.idiap.ch/~grangier/`

Honors

IDIAP Paper Award	Best IDIAP student paper of the year (2006)
ECML Paper Award	Google student paper award at European Conference on Machine Learning (2006)
ICANN Paper Award	European Neural Network Society student paper award at the International Conference on Neural Network (2006)
Hitachi Prize	Best Eurecom master thesis project (2003)

Other Activities

Reviewer	Reviewer or co-reviewer for conferences like NIPS, ICML, or CVPR.
Workshop Organizer	Co-organizer of the NIPS 2006 Workshop on Learning to Compare Examples (LCE) Workshop website `www.idiap.ch/lce/`