Learning to Rank Retrieval Results for Geographically Constrained Search Queries

João Vicente
joao.m.vicente@ist.utl.pt

Instituto Superior Técnico, INESC-ID
Av. Professor Cavaco Silva
2744-016 Porto Salvo, Portugal

ABSTRACT

The search for techniques capable of improving the quality of search engine results remains an important research topic. Recently, several authors have proposed to leverage geographical information, extracted from documents and from user queries, in order to present users with geographically relevant results. Other authors have proposed to use machine learning methods for building search engine ranking formulas, capable of combining multiple estimators of relevance, including geospatial distance and area overlap. This paper explores the usage of learning to rank methods for geographic information retrieval. I specifically studied the impact of using different features, different learning to rank algorithms, and different methods to represent the geographic context of documents. Experiments were made with the datasets from the previous GeoCLEF evaluation campaign, where it was possible to conclude that the usage of learning to rank methods is indeed well-suited to the Geographic Information Retrieval context.

1. INTRODUCTION

The automatic retrieval and ranking of textual documents, according to geographic criteria of relevance, is a Geographic Information Retrieval (GIR) problem that has been receiving increasing attention in the scientific community. Several effective approaches for GIR have been proposed, exploring different methods to resolve place references, given over the textual documents, into geospatial footprints, and for computing the geospatial similarity between regions associated to documents and/or user queries. However, ranked retrieval models specific for GIR often combine several relevance estimates, derived from textual and geospatial similarity. Manually tuning the involved parameters is a hard task that can lead to over-fitting. Thus, there has been an increasing interest, in both general information retrieval and GIR, on the usage of machine learning methods for building retrieval formulas, capable of estimating relevance for query-document pairs by combining multiple features. Learning to Rank for Information Retrieval (L2R4IR) is a supervised machine learning approach where the general idea is to use hand-labeled data (e.g., document collections containing relevance judgments for specific sets of queries) to train ranking models, this way leveraging on data to combine the different estimators of relevance in an optimal way.

In the context of GIR, Martins and Calado [21] have already explored the usage of learning to rank techniques, by combining a set of different features based on textual and geographical similarities. These authors used a freely-available Web service, namely Yahoo! Placemaker, to associate documents and user queries to a single encompassing geographic scope, later using relevance estimates computed with basis on these scopes. Their experimental results attested for the adequacy of L2R4IR methods in the context of GIR.

In my work, I also explored the usage of learning to rank methods for geographic information retrieval, leveraging on the datasets made available in the context of the previous GeoCLEF evaluation campaigns [18]. However, I introduce some differences in comparison with the previous work by Martins and Calado, namely:

- Noticing that documents often mention multiple places in their textual contents (i.e., they can not always be correctly summarized in terms of a single geographic scope), I experimented with the usage of features computed from the full set of place references recognized in the texts, as opposed to using a single geographic scope per document.

- Noticing that user queries can be very different between themselves, referring to either broad or very specific and constrained geographic regions, I hypothesize that it may be difficult for a single ranking model to perform equally well on all types of user queries. Therefore, I compared a single ranking model, which deals with all different types of regions associated to the queries, with models specifically designed to handle some types of regions. I experimented with the classification of queries into two groups, namely queries referring to broad geographic regions and queries referring

to narrow geographic regions, and then trained separate ranking models for handling each type of query.

- Besides the differences in terms of query regions, user queries can also use different geospatial relations (e.g., *close to* or *inside*) to associate the query topics to the query region. I experimented with the usage of features that capture these geospatial relations, although the queries from the GeoCLEF dataset were often similar between themselves and mostly involved a simple containment relationship.
- I experimented with the usage of seven different learning to rank algorithms, representative of the pointwise, pairwise and listwise L2R4IR approaches, as opposed to the previous work by Martins and Calado in which only the SVM*map* algorithm was tested.

The rest of this paper is organized as follows: Section 2 presents the main concepts and related work. Section 3 presents the learning to rank techniques used in my experiments. Section 4 introduces the multiple features upon which I leverage for estimating relevance, and the mechanism used for building the multiple ranking models. Section 5 presents the experimental validation process, together with the obtained results. Finally, Section 6 presents my conclusions and points of directions for future work.

2. CONCEPTS AND RELATED WORK

Previous works in the area of Geographic Information Retrieval (GIR) have addressed the multiple problems associated to the development of retrieval systems capable of finding information, from large document collections, that is relevant to geographically constrained user queries. These problems include (i) the pre-processing of textual documents in order to extract geographic information in the form of place references, (ii) the understanding of the geographic intentions behind user queries, and (iii) the retrieval and ranking of documents according to user queries that impose some sort of geographic restrictions.

Different GIR methods have been tested in the context of the GeoCLEF evaluation campaign, addressing the three challenges above through different mechanisms [18, 19]. The most common approaches were based on heuristic combinations of the standard IR metrics used in text retrieval (e.g., TF-IDF), with similarity metrics for geographic regions associated to the documents, based on distance, area containment or taxonomic similarity.

In the context of the first task from the list above, i.e. the extraction of geographic information from documents, authors like Leidner or Martins et al. have proposed to split the task into two sub-problems, namely (i) place reference recognition and (ii) place reference disambiguation [16, 20]. Place reference recognition refers to delimiting the text tokens referencing locations, while place reference disambiguation refers to giving unique identifiers, typically geospatial coordinates, to the location names that were found, in order to associate the recognized places to their real locations in the surface of the Earth. The main challenges in both tasks are related with the ambiguity of natural language. Amitay et al. characterized these ambiguity problems according to two types, namely geo/non-geo and geo/geo [1]. Geo/non-geo ambiguity occurs when location names have a non-geographic meaning (e.g., the word *Turkey* can refer to either the country or the bird). Geo/geo ambiguity refers to distinct locations with the same name (e.g. *London*, a city in England or in Ontario, Canada).

In particular, Leidner studied different approaches for the complete resolution of geographic references in text [16]. Most of the studied methods resolve places references by matching expressions from the texts against dictionaries of location names, and use disambiguation heuristics like default senses (e.g., the most important referenced location is choosen, estimated by the population size) or the spatial minimality (e.g., the correct disambiguation should minimize the polygon that covers all the geographic references contained in the document). More recently, Martins et al. studied the usage of machine learning approaches in the recognition and disambiguation of geographic references, using Hidden Markov Models in the recognition task, and SVM regression models with features corresponding to the heuristics surveyed by Leidner, in the disambiguation task [20].

Besides the resolution of place references in text, previous works have also proposed methods for summarizing all place references associated to a given document into an encompassing geographic scope [1, 2]. Typical methods either use a hierarchy of *part-of* relations among the recognized place references to generalize from the available information (e.g., if several cities from the same country are mentioned, this might mean that this country is the geographic scope of the document) [1], or use the overlapping area for all the geospatial polygons corresponding to the recognized place references, trying to find the most specific place that is related to all the place references made in the text [29].

Currently, there are many commercial products for recognizing and disambiguating place references in text. An example is the Yahoo! Placemaker[1] web service, which was used in this work and is also capable of associating documents to their corresponding geographic scopes.

In the context of the second task, i.e. query interpretation, authors like Gravano et al. have proposed query classification methods to distinguish local queries (i.e., queries having a geographic intention) from global queries (i.e., queries without a localization component) [10]. This is a particularly challenging problem because queries are often highly ambiguous or underspecify the information they are after. Using state-of-the-art log-linear regression models, and deriving the feature representations of the queries from the results produced for them by search engines, Gravano et al. reported results of approximately 0.89 in terms of Accuracy.

Zhuang et al. proposed to leverage on a geographical click probability distribution model, built by mining past user clicks and mapping IP addresses to geographical locations, to determine whether a search query is geo-sensitive, and also to detect and disambiguate the associated geographical

[1] `http://developer.yahoo.com/geo/placemaker/`

locations [36]. Although interesting, this approach suffers from the limitation of requiring large amounts of query-log data, in order to infer the probability distributions.

Yi et al. proposed to use features derived from a probabilistic representation of the language surrounding the mention of a city name in Web queries. Probabilistic models based on those features were used to address several problems related to geographical query interpretation with high accuracy, namely (i) identifying the users' implicit geo intent and pinpointing the specific city corresponding to this intent, (ii) determining whether the geo-intent is localized around the users' current geographic location, and (iii) predicting cities for queries that have a mention of an entity that is located in a specific place [31].

In the context of the third task, several studies have addressed the retrieval of documents by combining different metrics of textual and geographic similarity in order to construct an appropriate ranking function [24, 33, 21]. Textual similarity metrics are similar to those used in standard information retrieval systems, while geographic similarity metrics can be derived from geospatial distance between locations [22], from region overlap [8], or from taxonomical distance between the nodes in a hierarchical taxonomy encoding *part-of* relations between locations [13].

According to Frontiera et al. [8], there are two fundamental principles behind most estimators of geographic relevance. The first one is Tobler's First Law of geography, which states that *everything is related to everything else, but near things are more related than distant things* [26]. The second principle corresponds to the notion that *topology matters and metric refines* [6]. In geographic spaces, topology is considered to be first-class information, whereas metric properties (e.g., distances) are used as refinements that are frequently less exactly captured. Also according to Frontiera et al., spatial similarity can be defined as a function of (i) topological spatial relationships, (ii) metric spatial characteristics and (iii) directional spatial relationships. The first class includes functions to identify if two spatial objects intersect or not, and precisely define how this intersection happens, including notions such as touching, containing, equaling or overlapping. The second class includes notions of area, perimeter, length, shape, density and dispersion, among others. The third and last class refers to orientation by compass, including concepts such as north, south, east or west.

To measure similarity, Frontiera et al. [8] proposed to rely on relative region overlap, although noticing that geospatial similarity metrics should also contemplate other perspectives. When comparing different metrics, it was reported that the approach proposed by Hill [11] and the Union approach proposed by Janée[2] achieved better results.

Yu and Cai proposed a dynamic document ranking scheme to combine the thematic and geographic similarity measures on a per-query basis [33]. The authors used query specificity (i.e., the geographic area covered by the query) to determine the relative weights of different sources of ranking evidence for each query (i.e., the weight of a geographic relevance measure, derived from geospatial distance, is inversely proportional to the geospatial area of the query).

Since Geographic Information Retrieval involves tuning combinations of multiple relevance estimators, I propose to leverage on previous works concerning the subject of learning to rank for information retrieval (L2R4IR). In the past, Martins and Calado have already experimented with the usage of L2R4IR methods in geographic information retrieval, attesting for its advantages over heuristic combinations of relevance estimators. Tie-Yan Liu presented a good survey on the subject of L2R4IR [17], categorizing the previously proposed algorithms into three groups, according to their input representation and optimization objectives:

- **Pointwise approach** - L2R4IR is seen as either a regression or a classification problem. Given feature vectors of each single document from the data for the input space, the relevance degree of each of those individual documents is predicted with scoring functions corresponding to regression or classification models. The scores can then be used to sort documents and produce the final ranked list. Several different pointwise methods have been proposed, including the Additive Groves algorithm [25].

- **Pairwise approach** - L2R4IR is seen as a binary classification problem for document pairs, since the relevance degree can be regarded as a binary value which tells which document ordering is better for a given pair of documents. Given feature vectors of pairs of documents from the data for the input space, the relevance degree of each of those documents can be predicted with classification models which try to minimize the average number of misclassified document pairs. Several different pairwise methods have been proposed, including SVM*rank* [12], RankNet [4] or RankBoost [7].

- **Listwise approach** - L2R4IR is addressed in a way that takes into account an entire set of documents, associated with a query, as instances. These methods train a ranking function through the minimization of a listwise loss function defined on the predicted list and the ground truth list. Given feature vectors of a list of documents of the data for the input space, the relevance degree of each of those documents can be predicted with scoring functions which try to directly optimize the value of a particular information retrieval evaluation metric, averaged over all queries in the training data [17]. Several different listwise methods have also been proposed, including SVM*map* [34], AdaRank [30] or Coordinate Ascent [23].

In my work, I made experiments with the application of representative learning to rank algorithms from the pointwise, pairwise and listwise approaches, namely the SVM*rank*, RankNet, RankBoost, AdaRank, SVM*map*, Coordinate Ascent and Additive Groves algorithms, in a task of geographic information retrieval where geographic information is encoded over textual documents and extracted through the usage of Yahoo! Placemaker.

[2] `http://www.alexandria.ucsb.edu/archive/2003/similarity.html`

Noticing that significant differences may exist between user queries, several authors have proposed to use multiple ranking models, specific to the type of query, in order to improve learning to rank effectiveness. For instance Geng et al. proposed a query-dependent ranking method which dynamically creates a ranking model for a given query by using the k nearest training queries and their corresponding query-document feature vectors, afterwards using this model to rank the documents with respect to the query [9]. Their experimental results showed that query-dependent ranking outperformed the baseline method of using a single ranking function, effectively leveraging the useful information from the similar queries and avoiding the negative effects from the dissimilar ones. Zhu el al. demonstrated that it is highly beneficial to divide queries into multiple groups and address ranking problems through multiple models based on query difficulty [35]. Experiments using a classification model to predict query difficulty, latter using SVMrank or RankNet to build specific ranking models for each difficulty class, indicated a significant improvement in the task of web search ranking. Similar ideas are explored in my work, by also using query-specific ranking models with basis on the geographical scope of the queries (i.e., one model for queries refering to large locations, and another for queries referring to more constrained locations).

3. LEARNING TO RANK FOR GIR

In my work, I followed a general approach which is common to most supervised learning to rank methods, consisting of two separate steps, namely training and testing. Figure 1 provides an illustration.

Given a set of queries $Q = \{q_1, \ldots, q_{|Q|}\}$ and a collection of documents $D = \{d_1, \ldots, d_{|D|}\}$, where each query is associated with specific documents from the collection, a training dataset for learning to rank is created as a set of query-document pairs, each $(q_i, d_j) \in Q \times D$, upon which a relevance judgment indicating the match between q_i and d_j is assigned by a labeler. This relevance judgment can be a binary label, i.e., relevant or non-relevant. For each instance (q_i, d_j), a feature extractor produces a vector of features that describe the match between q_i and d_j. Features can range from classical IR estimators computed from the documents associated with the geographic location of the queries (e.g., term frequency, inverse document frequency, BM25, etc.) to geospatial functions which measure the distance or overlap between two regions. The inputs to the learning algorithm comprise training instances, their feature vectors, and the corresponding relevance judgments. The output is a ranking function, f, where $f(q_i, d_j)$ is supposed to either give the true relevance judgment for (q_i, d_j), or produce a ranking score for d_j so that when sorting documents according to these scores, the more relevant ones appear at the top of the ranked list of results.

During the training process, the learning algorithm attempts to learn a ranking function capable of sorting documents in a way that optimizes a particular bound on an information retrieval performance measure (e.g., Mean Average Precision). In the test phase, the learned ranking function is applied to determine the relevance between each document $d_j \in D$ and a new query q. In this paper, I experimented with the following learning to rank algorithms:

- SVM*rank* - This is a pairwise method which builds a ranking model in the form of a linear combination function, through the formalism of Support Vector Machines (SVMs) [12]. The idea is to minimize a hinge-loss function defined over a set of training queries, their associated pairs of documents, and the corresponding pairwise relevance judgments (i.e., pairwise preferences resulting from a conversion from the ordered relevance judgments over the query-document pairs). In my experiments, I used a linear kernel and the set the parameter was set to 980.
- RankNet - This is a probabilistic pairwise learning to rank method based on the formalism of artificial neural networks [4]. For every pair of correctly ranked documents, each document is propagated through the network separately. The differences between each two outputs are mapped to a probability through the logistic function. The cross entropy loss is then computed from that probability and the true label for each pair. Next, all weights in the network are updated using the error back propagation and the gradient descent methods. In my experiments, I used neural networks with 2 hidden layers and 50 nodes per layer. The number of training epochs was set to 300.
- RankBoost - This is a pairwise boosting technique for ranking [7]. Training proceeds in rounds, starting with all the pairs of documents being assigned to equal weight. At each round, the learner selects the weak ranker that achieves the smallest pairwise loss on the training data, with respect to the current weight distribution. Pairs that are correctly ranked have their weight decreased and those that are incorrectly ranked have their weight increased, so that the learner will focus more on the hard samples in the next round. The final model is essentially a linear combination of weak rankers. Weak rankers can theoretically be of any type, but they are most commonly chosen as a binary function with a single feature and a threshold. This function assigns a score of one when the feature value exceeds the threshold, and zero otherwise. In my experiments, the number of threshold candidates was set to 50 and the number of training rounds was set to 1000.
- AdaRank - The idea here is similar to that of RankBoost, except that AdaRank is a list-wise approach [30]. Hence, this method directly maximizes any desired IR metric computed over ranked lists of results, whereas the objective in RankBoost is simply to minimize the pairwise loss. In my experiments, the number of training rounds was set to 1000, and I trained the models to optimize the highly informative Mean Average Precision (MAP) metric [32].
- SVM*map* - This is a listwise method which builds on the formalism of Structured Support Vector Machines [27], attempting to optimize the metric of Average Precision [34]. The idea is to minimize a loss function which measures the difference between the performance of a perfect ranking (i.e., when the Average

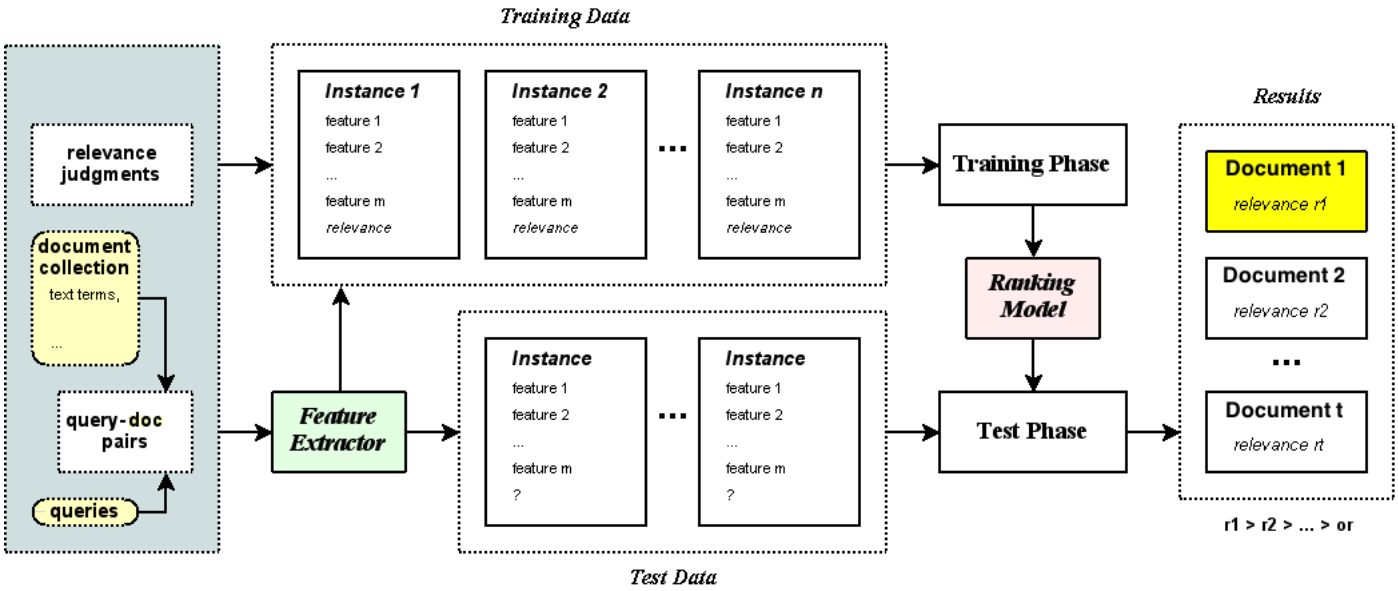

Figure 1: The general procedure involved in learning to rank tasks, adapted from [17].

Precision equals one) and the minimum performance of an incorrect ranking (i.e., when it is less than one). In my experiments, I set the parameter C to 12 and γ to 0,250.

- Coordinate Ascent - This is a state-of-the-art listwise method, proposed by Metzler and Croft, which uses coordinate ascent to optimize the ranking model parameters [23]. Coordinate ascent is a well-known technique for unconstrained optimization, which optimizes multivariate objective functions by sequentially doing optimization in one dimension at a time. The method cycles through each parameter and optimizes over it, while fixing all the other parameters. In the context of this paper, the term Coordinate Ascent refers to this particular ranking technique, rather than the general optimization method. In my experiments, I set the number of different random initializations to 50, and I trained the models to optimize the Mean Average Precision (MAP) metric

- Additive Groves - This is pointwise algorithm which builds a ranking model through additive models and regression trees [25]. The algorithm starts by initializing a grove (i.e., an additive model containing a small number of large trees) with a single small tree. Iteratively, the grove is gradually expanded by adding a new tree or enlarging the existing trees of the model. The trees in the groves are trained with the set of documents which were misclassified by the other previously trained trees. Trees are retrained until the overall predictions converge to a stable function. Since a single grove can easily suffer from overfit when the trees start to become very large, the authors introduced a bagging procedure in this approach in order to deal with overfitting. Bagging is a method which improves a model's performance by reducing variance. On each bagging iteration, the algorithm takes a sample from the training set, and uses it to train the full model of the grove. In this algorithm, the implementation automatically tune the parameters through a grid search procedure.

The survey paper by Tie-Yan Liu discusses most of the above ranking methods in more detail [17].

4. RANKING FEATURES

The considered set of features is divided in three groups, namely (i) textual features, (ii) geographical features and (iii) query-level collaborative features. The textual features are similar to those used in standard text retrieval systems and also in previous learning to rank experiments (e.g., TF-IDF and BM25 scores). The geographical features are based on those reported in previous works related to geographic information retrieval [22] (e.g. the area of overlap between the geographical scope of the topic and the geographic references recognized in the text).

The rest of this section describes the full set of considered features. The methods responsible for their computation were implemented through the full-text search and geospatial capabilities of *Microsoft SQL Server "Denali" 2011*.

4.1 Textual Similarity Features

Similarly to previous works in the area of geographic information retrieval, I also used ranking features based on the textual similarity between the query and the contents of the documents. For each query-document pair, I used the Okapi BM25 document-scoring function to compute textual similarity features. Okapi BM25 is a state-of-the-art IR ranking mechanism, composed of several simpler scoring functions with different parameters and components (e.g., term frequency and inverse document frequency). It can be

computed through the formula shown in Equation 1, where $Terms(q)$ represents the set of terms from query q, $Freq(i, d)$ is the number of occurrences of term i in document d, $|d|$ is the number of terms in document d, and $\mathcal{A}$ is the average length of the documents in the collection. The Okapi BM25 parameters k_1 and b were set to the default values of 1.2 and 0.75, respectively.

$$BM25(q,d) = \sum_{i \in Terms(q)} \log\left(\frac{N - Freq(i) + 0.5}{Freq(i) + 0.5}\right) \times \frac{(k_1+1) \times \frac{Freq(i,d)}{|d|}}{\frac{Freq(i,d)}{|d|} + k_1 \times (1 - b + b \times \frac{|d|}{\mathcal{A}})} \quad (1)$$

I also experimented with other textual features commonly used in ad-hoc IR systems, such as *Term Frequency* and *Inverse Document Frequency*.

Term Frequency (TF) corresponds to the number of times that each individual term in the query occurs in the document. Equation 2 describes the TF formula, where $Terms(q)$ represents the set of terms from query q, $Freq(i, d)$ is the number of occurrences of term i in document d, and $|d|$ represents the number of terms in document d.

$$TF_{q,d} = \sum_{i \in Terms(q)} \frac{Freq(i,d)}{|d|} \quad (2)$$

The Inverse Document Frequency (IDF), given by Equation 3, is the sum of the inverse document frequencies for each query term. In the formula, $|D|$ is the size of the document collection and $f_{i,D}$ corresponds to the number of documents in the collection where the i_{th} query term occurs.

$$IDF_q = \sum_{i \in Terms(q)} \log \frac{|D|}{f_{i,D}} \quad (3)$$

4.2 Geographical Similarity Features

The considered set of geographic similarity features is mostly based on the geospatial distance and relative area overlap, summarizing the existence of multiple locations through their minimum, maximum and average values.

In what concerns geospatial distance, I considered the distance between the centroid point for a geographic region associated to the document and the centroid point for the geographic region associated to the query, expressed in Kilometers.

I also considered the Hausdorff distance [28] between a region associated to document and the region associated to the query, given by the formula shown in Equation 4:

$$sim(S_d, S_t) = max(max\{\forall p_1 \in S_d | min\{\forall p_2 \in S_t | dist(p_1,p_2)\}\}, max\{\forall p_1 \in S_t | min\{\forall p_2 \in S_d | dist(p_1,p_2)\}\}) \quad (4)$$

In terms of areas, I considered the area of the geographic region associated to the query, the area of the geographic region associated to the document, and the area of overlap between the geographic region associated to the query and the geographic region associated to the document, in squared Kilometers.

I also considered the relative degree of overlap between the region associated to the document and the region associated to the query, given by the formulas originally proposed in [11], in [3], in [8] and introduced by G. Janée shown in Equations 5, 6, 7, and 8 respectively.

$$sim(S_d, S_t) = \frac{2 \times area(S_d \cap S_t)}{area(S_d) + area(S_t)} \quad (5)$$

$$sim(S_d, S_t) = \frac{\frac{area(S_d \cap S_t)}{area(S_t)} + \frac{area(S_d \cap S_t)}{area(S_d)}}{2} \quad (6)$$

$$sim(S_d, S_t) = \begin{cases} \frac{area(S_d)}{area(S_t)} & \text{if } S_d \subset S_t \\ \frac{area(S_t)}{area(S_d)} & \text{if } S_d \supset S_t \\ \frac{\frac{area(S_d \cap S_t)}{area(S_t)}}{\frac{area(S_d \cap S_t)}{area(S_d)}} & \text{if } S_d \cap S_t \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$sim(S_d, S_t) = \frac{area(S_d \cap S_t)}{area(S_d \cup S_t)} \quad (8)$$

Finally, I also considered the hierarchical distance measure originally proposed in [14], proposed an approach based in the hierarchical distance between a query and a document, by computing the ancestors of the geographic scopes from the query and the document in a hierarchical taxonomy encoding geo-political subdivisions. The function Hierarchical Distance (HD) is computed over the taxonomy given by the ancestors of the geographic locations associated to the query and to the document. The formula for the hierarchical distance is shown in Equation 9, where the function $TaxonomyLevel()$ returns the distance from a given node in the geographic taxonomy to the root node (i.e., in a world gazetteer, the taxonomy level for */World/Europe/Portugal* is 2) and the function $Ancestors()$ returns, for a given node, its set of ancestor nodes in the taxonomy. In my experiments, the taxonomical information is provided by the Yahoo! GeoPlanet[3] Web service.

$$HD(S_d, S_t) = \frac{1}{TaxonomyLevel(S_d)} + \frac{1}{TaxonomyLevel(S_t)} + \sum_{x \epsilon Ancestors(S_d) - Ancestors(S_t)} \frac{1}{TaxonomyLevel(x)} + \sum_{y \epsilon Ancestors(S_t) - Ancestors(S_d)} \frac{1}{TaxonomyLevel(y)}$$

[3] `http://developer.yahoo.com/geo/geoplanet/`

(9)

The above similarity features are computed with basis on the geographical scopes assigned by Yahoo! Placemaker to summarize the set of place references made in each document, as well as with the individual place references that are made in text. In the case of the individual place references, I considered separate features corresponding to the minimum, maximum and average values for each of the features listed above (e.g., the minimum, maximum and average geospatial distances between the geographical scope of the query, and the set of place references made in the document).

I was also considered three binary features corresponding to the type of geospatial relation used in the query, taking the value of one if the query contains a specific relation between its topic and the referenced locations, and the value of zero otherwise. One feature corresponds to the contained relation, another to the *nearby* relation, and another to cardinal relations (e.g., *south of* or *east of*).

4.3 Query-Level Collaborative Features

Taking inspiration on the micro-collaborative ranking technique recently proposed in [5], I also used a set of features that leverage on information from a set of collaborators for the query, under the assumption that:

- The query can be expanded with information available from the document collection, this way building the set of collaborators.
- The query collaborators, although often noisy and sharing redundant information with the query, may also exhibit multifaceted information that complements the query, thus leading to better relevance estimates.

The collaborators for the query can be obtained by either (a) searching for geographically related documents according to the relative overlap metric proposed by [11], or (b) searching for textually similar documents according to the BM25 metric. The textual contents of the 5 most similar documents (i.e., the query-level collaborators) are used to expand the contents of the original query topics, which are then processed with Yahoo! Placemaker. Afterwards, the textual contents and the geographical scopes for the expanded queries are used to build collaborative features (i.e., textual and geographical query-level collaborative features) similar to those described in the previous subsections.

5. EXPERIMENTAL EVALUATION

This section presents the experimental evaluation of the proposed learning to rank methods for GIR, detailing the evaluation protocol and the obtained results.

5.1 Dataset and Experimental Protocol

For training and validating the different ranking models, I used the dataset that was previously made available in the context of the GeoCLEF[4] task at the Cross Language Evaluation Forum[5] (CLEF). Started in 2005 and performed every year until 2008, GeoCLEF aimed at the evaluation of Geographic Information Retrieval systems [19]. The dataset build from the four editions of GeoCLEF consists in a total of 100 query topics (i.e., there were 25 different topics in each edition), with binary relevance judgments obtained through a pooling approach for documents taken from a collection of 169,477 English newswire articles from the American newspaper Los Angels Times published in the year of 1994, and from the English newspaper Glasgow Herald published in the year of 1995. Table 1 presents a statistical characterization of the GeoCLEF dataset.

The GeoCLEFF dataset has a total of 2,742 query-document pairs labeled as relevant, and 58,770 query-document pairs labeled as non-relevant. This distribution is highly unbalanced, and thus it was necessary to perform some normalization before training the ranking models. In order to balance the training data, I filtered the set of non-relevant judgments for each query topic, in order to obtain a set of non-relevant documents equaling in number to the set of relevant documents. Using the BM25 feature, I retrieved the top ranked $n/2$ non-relevant documents for each query, where n is the number of relevant query-documents pairs, and then complemented this set with $n/2$ randomly selected non-relevant query-document pairs. Five GeoCLEF queries which had no relevant documents associated to them were removed from the final balanced dataset.

In order to validate the usage of learning to rank for geographic information retrieval, I used the GeoCLEF dataset in a leave-one-out cross-validation methodology, in which the queries were first divided into four sets, each one corresponding to a different edition of GeoCLEF. Four different experiments used three of the sets to train the ranking models, which were then evaluated over the remaining queries. The averaged results from the four cross-validation experiments were finally used as the evaluation result.

Each of the query topics from the GeoCLEF dataset was manually assigned to one of two groups according to their scope, namely (i) a group with query topics associated to large geographic regions (e.g., regions corresponding to entire continents or countries) and (ii) a group with query topics associated to small geographic regions (e.g., regions bellow the level of country, such as cities or states). Each group of queries was used to train separate ranking models, which were selected for usage at query time according to the type of query. I also trained global ranking models considering the full set of query-document pairs, using it to rank results for queries which could not be directly associated to a geographic scope (i.e., queries like *rivers with vineyards*).

5.2 Metrics

To measure the quality of the results, I used three different performance metrics, namely the Precision at rank position k (P@k), the Mean Average Precision (MAP), and the Nor-

[4] `http://ir.shef.ac.uk/geoclef/`
[5] `http://www.clef-campaign.org/`

	2005	2006	2007	2008
Number of topics	25	25	25	25
Average terms per topic title	6.64	5.76	6.08	5.48
Topics mentioning location names	21	19	13	14
Topics associated to large regions	13	10	12	12
Topics associated to small regions	12	12	5	12
Number of topics with a part-of relation	23	15	11	17
Number of topics with a near-to relation	0	3	2	0
Number of topics with a cardinal relation	2	4	4	7
Topics with relevant documents	25	24	22	24
Judged topic-document pairs	14,243	17,651	15,249	14,329
Relevant topic-document pairs	993	371	631	727
Non-Relevant topic-document pairs	13,250	17,280	14,638	13,602

Table 1: Statistical characterization of the GeoCLEF dataset.

Query-Specific Models	P@5	P@10	P@15	P@20	MAP	NDCG@5	NDCG@20
SVM*rank*	0,6399	0,5971	0,5511	0,5208	0,7194	0,6860	0,7644
RankNet	0,6576	0,6031	0,5422	0,5098	0,7298	0,7221	0,7731
RankBoost	0,6301	0,5833	0,5453	0,5220	0,6972	0,6849	0,7601
AdaRank	0,5633	0,5462	0,5047	0,4735	0,6802	0,6089	0,7088
SVM*map*	0,6912	0,6179	0,5716	0,5325	0,7473	0,7430	0,7926
Coordinate ascent	0,6788	0,6135	0,5613	0,5079	0,7630	0,7521	0,7893
Additive Groves	**0,7307**	**0.6467**	0,5874	0,5452	0,7474	0,7682	0,8020
Single Model	P@5	P@10	P@15	P@20	MAP	NDCG@5	NDCG@20
SVM*rank*	0,6685	0,6281	0,5877	0,5368	0,7393	0,7175	0,7906
RankNet	0,6134	0,5713	0,5461	0,5083	0,7107	0,6692	0,7570
RankBoost	0,6562	0,6031	0,5627	0,5215	0,7050	0,7039	0,7689
AdaRank	0,5926	0,5579	0,5191	0,4878	0,6684	0,6336	0,7186
SVM*map*	0,6478	0,5961	0,5619	0,5311	0,7302	0,7112	0,7883
Cordinate ascent	0,6949	0,6402	0,59150	0,5468	0,7412	0,7439	0,8059
Additive Groves	0,7292	0,6370	**0,5996**	**0,5543**	**0,7684**	**0,7823**	**0,8251**
BM25 baseline	0.2105	0.2684	0.3011	0.3111	0.4395	0.2365	0.4296

Table 2: The results obtained with the full set of features.

malized Discounted Cumulative Gain (NDCG).

Precision at rank k is used when a user wishes only to look at the first k retrieved documents. The precision is calculated at that rank cutoff position through Equation 10.

$$P@k[r] = \frac{r\,(k)}{k} \tag{10}$$

In the formula, $r(k)$ is the number of relevant documents retrieved in the top k positions. $P@k$ only considers the top-ranked documents as relevant and computes the fraction of such documents in the top-k elements of the ranked list.

The Mean of the Average Precision over the set of test queries is defined as the mean over the precision scores for all retrieved relevant documents. For each query, the Average Precision (AP) for a list of results r is given by:

$$AP[r] = \frac{\sum_{k=1}^{n} P@k[r] \times I\{g_{r_k} = \max(g)\}}{\sum_{k=1}^{n} I\{g_{r_k} = \max(g)\}} \tag{11}$$

In the formula, n is the number of documents associated with query q and $I\{g_{r_k} = \max(g)\}$ is an indicator function that returns 1 if the relevance grade g_{r_k} for document r_k is maximum, and zero otherwise. In my case, $\max(g) = 1$ (i.e., I have 2 different grades for relevance, 0 or 1).

The Normalized Discounted Cumulative Gain (NDCG) emphasizes the fact that highly relevant documents should appear on top of the ranked list. The metric is given by the following equation, where Z_k is a normalization factor corresponding to the maximum score that could be obtained when looking at the the top k retrieved documents.

$$NDCG@k[r] = Z_k \sum_{i=1}^{k} \frac{2^{g_{r_k}} - 1}{log_2(1+k)} \tag{12}$$

5.3 Usage of Different L2R4IR Algorithms

To validate the hypothesis that L2R4IR can effectively be used in GIR, I trained separate ranking models using the features described in Section 4 and reusing existing implementations of state-of-the-art learning to rank algorithms,

namely the SVM*rank*[6] implementation by Thorsten Joachims, the SVM*map*[7] implementation by Yue et al. [34], the Additive Groves[8] implementation by Sorokina et al [25] and the different implementations provided in the RankLib[9] Java package by Van Dang.

I also compared an approach that uses two different learning to rank models (i.e., one specifically designed to address queries associated to small geographic regions, and another for addressing queries associated to large regions) against a single model which is trained for all types of queries.

Table 2 presents the results that were obtained when considering the complete set of features described in Section 4. I compare the approach that considers query-specific ranking models, against the approach that uses a single ranking model. The results show that using query-specific models does not improve significantly the results against using a single model, in almost all of the considered learning to rank algorithms. The results also show that the different learning to rank algorithms achieve a similar performance, with the Additive Groves algorithm slightly outperforming the others. Table 2 also presents the results obtained through a baseline method corresponding to the usage of BM25 scores computed between the query and the textual contents of the documents, showing that all the proposed combination methods based on learning to rank with a rich set of features outperform the baseline BM25 textual retrieval approach.

Figure 2 shows the obtained results in terms of the Average Precision metric, for each of the individual GeoCLEF query topics and when considering the best-performing learning to rank algorithm (i.e., Additive Grove models). In the figure, the dashed lines correspond to the MAP scores, the red bars correspond to queries associated to large geographic regions, the blue bars correspond to queries associated to small geographic regions, and the green bars correspond to queries that were not associated to a class according to their geographic region. The figure compares the results obtained with a single ranking model for all queries against the results obtained with query-specific models, showing that the query-based and the single model approaches perform poorly in almost the same all queries.

A Fisher's two-sided paired randomization statistical significance test was used to see if the differences between (a) Additive Groves versus Coordinate Ascent using the single ranking model and (b) a single Additive Groves model versus query-specific Additive Groves models, were indeed significant. The test results indicated that the differences between both learning to rank algorithms were not significant, with the significance level for the MAP metric being high at 0.183060. When comparing the results of the query-based versus the single ranking model, the results showed that single ranking model significantly different, for the MAP metric, having a $p - value$ of 0.063190.

[6] `http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html`
[7] `http://projects.yisongyue.com/svmmap/`
[8] `http://additivegroves.net/#downloads`
[9] `http://www.cs.umass.edu/~vdang/ranklib.html`

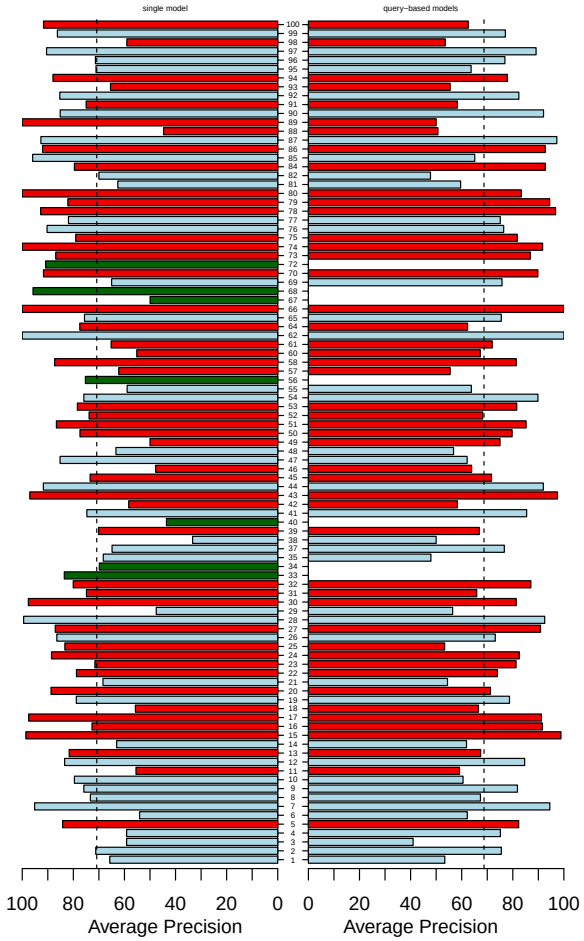

Figure 2: The obtained results in terms of average precision for each of the individual query topics, using Additive Groves models with all the features.

5.4 Impact of Different Ranking Features

In a separate experiment, I attempted to measure the impact of the different types of ranking features over the quality of the results. Using the best performing supervised learning to rank algorithm, namely Additive Groves, I separately measured the results obtained by the query-based and the single model approaches when using different sets of features. Table 3 shows the obtained results, separating the cases in which I considered (a) textual similarity together with the full set of geographic features obtained from the geographic scopes, the individual place references made in the texts, and the query topics (i.e., the query-specific features),(b) textual similarity together with geographic features computed from the geographic scopes of the documents, and (c) textual similarity alone.

The results show that the combination of all features achieves the best results. By considering the individual place references, as opposed to using a single encompassing geo-

Query-Specific Models	P@5	P@10	P@15	P@20	MAP	NDCG@5	NDCG@20
Text + Individual Places + Scopes	**0,7307**	**0,6467**	0,5874	0,5452	0,7474	0,7682	0,8020
Text + Geographic Scopes	0,6174	0,5715	0,5366	0,4944	0,7122	0,6840	0,7530
Text Similarity	0,5765	0,5405	0,5300	0,4854	0,6844	0,6366	0,7253
Single Model	P@5	P@10	P@15	P@20	MAP	NDCG@5	NDCG@20
Text + Individual Places + Scopes	0,7292	0,6370	**0,5996**	**0,5543**	**0,7684**	**0,7823**	**0,8251**
Text + Geographic Scopes	0,6286	0,5721	0,5410	0,4994	0,6763	0,6605	0,7304
Text Similarity	0,5849	0,5796	0,5379	0,4994	0,6765	0,6347	0,7344

Table 3: The obtained results when considering different sets of features.

graphic scope, the results improved. However, the results also show that textual similarity features together with geographic scopes have a high impact in the system, since their scores are almost as high as when considering all the features. Despite their complexity, the individual place reference features contribute only marginally to the performance of the ranking models.

A Fisher's two-sided paired randomization statistical significance test was used to see if the differences between Additive Groves using all the features, versus Additive Groves using geographic scopes and textual similarities, also with a single ranking model, were indeed significant. The test results indicated that the differences between the two feature groups were indeed significant, with the significance level for the MAP metric at 0.017960, and for NDCG@20 at the value of 0.034760.

In what concerns on Query-Level Collaborative approach, Table 4 presents the obtained results when considering different sets of features. The groups of features that were considered included (a) the full set of features described in Section 3, (b) the full set of features except for those corresponding to the query-level collaborative features based on textual collaborators, (c) the full set of features except for those corresponding to the query-level collaborative features based on geographic collaborators, (d) textual similarity together with the full set of geographic features obtained from the individual place references made in the texts, (e) textual similarity together with geographic features computed from the geographic scopes of the documents, and (f) textual similarity alone. The table also presents the results obtained through a baseline method corresponding to the usage of BM25 scores computed between the query and the textual contents of the documents, showing that all the proposed combination methods based on learning to rank outperform the baseline.

According to the results given in Table 4, we have that the combination of all the features described in Section 4 achieves the best results. The query-level collaborative features, obtained through geographically similar (i.e., CG features) or textually similar (i.e., CT features) documents, can be used to build better ranking models. By considering the individual place references (i.e., GP features), as opposed to using a single encompassing geographic scope (i.e., GS features), I can also improve the results. However, my experiments also showed that textual similarity features (i.e., T features) have a high impact in the system. The query-level collaborative features can be used to build better ranking models. By considering the individual place references, as opposed to using a single encompassing geographic scope, I can also improve the results. However, my experiments also showed that textual similarity features have a high impact in the system, since their scores are almost as high as when considering all features.

A Fisher's two-sided paired randomization statistical significance test was used to see if the differences between full set of features against all individual experiments were significant. The test results indicated that the differences between full set of features and all the experiments that includes geographic features were not significant different. However, when comparing the results of the full set of features against only textual features, the results showed that full features was significant, for the MAP metric, having a $p - value$ of 0.000180.

5.5 Greedy Forward Feature Selection

I also experimented with the usage of a greedy forward feature selection mechanism with the Coordinate Ascent algorithm, using the default parameters. I implemented the greedy forward feature selection mechanism inside RankLib, in order to search for the minimum set of features that produces the best result in terms of MAP. The first step is to choose the best isolated feature. The next step, is to combine this feature with all the others and train, test and evaluate the four folds, to see if this way one can can improve the MAP. This process is repeated iteratively, saving on each iteration the best minimum set of features. The process stops when one can not improve the MAP.

The feature which achieved the best results individually was the maximum overlap according to Janée's metric for all place details in query and document, with 0.5965 of MAP. I combined all features with this one, and the best pair was the combination of maximum of Janée area with BM25 feature, achieving 0.6548 of MAP. The process was repeated until I could not improve the MAP score. The new features added were the Inverse Document Frequency, the area of Hill, the Frontiera metric and the Minimum Distance of Geographic Scopes 0.6697, 0.6787, 0.6815 and 0.6821 of MAP, respectively.

The values in Table 5 show that the combination of the best set of features achieves the best results. However, Fisher's two-sided paired randomization statistical significance test was used to see if the differences between the full set of features against the best set of features was indeed significant. The test results indicated that the differences between them

	P@5	P@10	P@15	P@20	MAP	NDCG@5	NDCG@20
T + GP + GS + CT + CG	0,6132	**0,5857**	**0,5440**	0,5014	**0,6828**	0,6709	0,7288
T + GP + GS + CT	**0,6280**	0,5696	0,5414	**0,5066**	0,6819	0,6708	0,7338
T + GP + GS + CG	0,6032	0,5736	0,5399	0,5047	0,6819	0,6497	0,7310
T + GP + GS	0,6189	0,5657	0,5339	0,5020	0,6798	**0,6960**	**0,7469**
T + GS	0,6155	0,5646	0,5174	0,4885	0,6678	0,6680	0,7158
T	0,5916	0,5319	0,5003	0,4705	0,6268	0,5907	0,6701
BM25	0.2105	0.2684	0.3011	0.3111	0.4395	0.2365	0.4296

Table 4: Results for different sets of collaborative features.

	P@5	P@10	P@15	P@20	MAP	NDCG@5	NDCG@20
Full Set of Features	0,5631	0,5454	0,5383	0,5071	0,6672	0,6198	0,6589
Best Set of Features	**0,6051**	**0,5537**	**0,5394**	**0,5202**	**0.6821**	**0,6592**	**0,7107**
BM25 baseline	0.2105	0.2684	0.3011	0.3111	0.4395	0.2365	0.4296

Table 5: Results with greedy-forward feature selection.

were not significant different, with a $p - value$ of 0.1060 for the MAP metric.

6. CONCLUSIONS

My work explored the usage of learning to rank methods in the context of Geographic Information Retrieval, extending the previous work by Martins and Calado [21]. I compared query-specific learned ranking models, where queries are grouped with basis on their geographic area of interest, against global ranking models, concluding that they have almost the same performance. Moreover, I also demonstrated that by considering multiple place references per document, instead of summarizing documents to a single geographic scope, one can improve the results. Experiments on the GeoCLEF dataset with several representative learning to rank algorithms produced encouraging results. Further experiments are nonetheless required, and I plan on performing additional tests with more features.

For future work, and since rank aggregation is currently a very hot topic of research, I also plan on experimenting with unsupervised learning to rank methods for combining the different similarity scores. Recent works in the area of information retrieval have described several advanced unsupervised learning to rank methods, such as the ULARA algorithm recently proposed by Klementiev et al. [15].

Acknowledgements

This work was partially supported by the Fundação para a Ciência e a Tecnologia (FCT), through the project grant with reference PTDC/EIA-EIA/109840/2009 (SInteliGIS).

References

[1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: geotagging web content. In *Proceedings of the 27th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, 2004.

[2] Ivo Anastácio, Bruno Martins, , and Pável Calado. A comparison of different approaches for assigning geographic scopes to documents. In *Proceedings of the 1st Portuguese Symphosium on Informatics*, 2009.

[3] Kate Beard and Vyjayanti Sharma. Multidimensional ranking for data in digital spatial libraries. *International Journal on Digital Libraries*, 1, 1997.

[4] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, 2005.

[5] Zheng Chen and Heng Ji. Collaborative ranking: A case study on entity linking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011.

[6] Max J. Egenhofer and Matthew P. Dube. Topological relations from metric refinements. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009.

[7] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4, December 2003.

[8] Patricia Frontiera, Ray Larson, and John Radke. A comparison of geometric approaches to assessing spatial similarity for GIR. *International Journal of Geographical Information Science*, 22, 2008.

[9] Xiubo Geng, Tie-Yan Liu, Tao Qin, Andrew Arnold, Hang Li, and Heung-Yeung Shum. Query dependent ranking using k-nearest neighbor. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[10] Luis Gravano, Vasileios Hatzivassiloglou, and Richard Lichtenstein. Categorizing web queries according to geographical locality. In *Proceedings of the 12th international conference on Information and knowledge management*, 2003.

[11] Linda Ladd Hill. *Access to geographic concepts in online bibliographic files: effectiveness of current practices and the potential of a graphic interface*. PhD thesis, University of Pittsburgh, 1990.

[12] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 9th ACM Conference on Knowledge Discovery and Data Mining*, 2002.

[13] C. B. Jones, H. Alani, and D. Tudhope. Geographical information retrieval with ontologies of place. In *Proceedings of the 5th International Conference on Spatial Information Theory*, 2001.

[14] Christopher B. Jones, Harith Alani, and Douglas Tudhope. Geographical information retrieval with ontologies of place. In *Proceedings of the 5th International Conference on Spatial Information Theory*, 2001.

[15] Alexandre Klementiev, Dan Roth, Kevin Small, and Ivan Titov. Unsupervised rank aggregation with domain-specific expertise. In *Proceedings of the 21st International Joint Conference on Artifical intelligence*, 2009.

[16] Jochen L Leidner. *Toponym Resolution in Text : Annotation, Evaluation and Applications of Spatial Grounding of Place Names*. PhD thesis, University of Edinburgh, 2008.

[17] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3, 2009.

[18] T. Mandl, P. Carvalho, F. Gey, R. Larson, D. Santos, and C. Womser-Hacker. Geoclef 2008: The clef 2008 cross-language geographic information retrieval track overview. In *CLEF*, 2008.

[19] Thomas Mandl, Fredric Gey, Giorgio Di Nunzio, Nicola Ferro, Mark Sanderson, Diana Santos, and Christa Womser-Hacker. An evaluation resource for geographic information retrieval. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, 2008.

[20] Bruno Martins, Ivo Anastácio, and Pável Calado. A machine learning approach for resolving place references in text. In *Proceedings of AGILE-2010, the 13th AGILE International Conference on Geographical Information Science*, 2010.

[21] Bruno Martins and Pável Calado. Learning to rank for geographic information retrieval. In *Proceedings of the 6th ACM Workshop on Geographic Information Retrieval*, 2010.

[22] Bruno Martins, Nuno Cardoso, Marcirio Chaves, Leonardo Andrade, and Mário J. Silva. The University of Lisbon at GeoCLEF 2006. In *Proceedings of the 6th Cross-Language Evaluation Forum*, 2007.

[23] Donald Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10, 2007.

[24] Simon Overell. *Geographic Information Retrieval: Classification, Disambiguation and Modelling*. PhD thesis, Imperial College London, 2009.

[25] D. Sorokina, R. Caruana, and M. Riedewald. Additive groves of regression trees. In *Proceedings of the 18th European Conference on Machine Learning (ECML'07)*, 2007.

[26] W. R. Tobler. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 46, 1970.

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6, 2005.

[28] M. D. Wills. Hausdorff distance and convex sets. *Journal of Convex Analysis*, 14(1), 2007.

[29] A. G. Woodruff and C. Plaunt. Gipsy: automated geographic indexing of text documents. *Journal of the American Society Information Sciences*, 45(9), 1994.

[30] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[31] Xing Yi, Hema Raghavan, and Chris Leggetter. Discovering users' specific geo intention in web search. In *Proceedings of the 18th international conference on World Wide Web*, 2009.

[32] Emine Yilmaz and Stephen Robertson. On the choice of effectiveness measures for learning to rank. *Information Retrieval*, 13, 2010.

[33] Bo Yu and Guoray Cai. A query-aware document ranking method for geographic information retrieval. In *Proceedings of the 4th ACM Workshop on Geographical Information Retrieval*, 2007.

[34] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th ACM SIGIR international Conference on Research and Development in Information Retrieval*, 2007.

[35] Zeyuan Allen Zhu, Weizhu Chen, Tao Wan, Chenguang Zhu, Gang Wang, and Zheng Chen. To divide and conquer search ranking by learning query difficulty. In *Proceeding of the 18th ACM conference on Information and Knowledge Management (CIKM '09)*, 2009.

[36] Ziming Zhuang, Cliff Brunk, and C. Lee Giles. Modeling and visualizing geo-sensitive queries based on user clicks. In *Proceedings of the 1st international workshop on Location and the Web (LocWeb '08)*, 2008.