Revisiting the Cluster-based Paradigm for Implicit Search Result Diversification

Hai-Tao Yu[a,*], Adam Jatowt[b], Roi Blanco[c], Hideo Joho[d], Joemon M. Jose[e], Long Chen[e], Fajie Yuan[e]

[a]*Faculty of Library, Information and Media Science, University of Tsukuba, Tsukuba, Japan*
[b]*Department of Social Informatics, Graduate School of Informatics, Kyoto University, Kyoto, Japan*
[c]*IRLab, Computer Science Department, University of A Coruña, Spain*
[d]*Research Center for Knowledge Communities, Faculty of Library, Information and Media Science, University of Tsukuba, Tsukuba, Japan*
[e]*School of Computing Science, University of Glasgow, Glasgow, UK*

Abstract

To cope with ambiguous and/or underspecified queries, *search result diversification* (SRD) is a key technique that has attracted a lot of attention. This paper focuses on *implicit SRD*, where the subtopics underlying a query are *unknown*. Many existing methods appeal to the greedy strategy for generating diversified results. A common practice is using a heuristic criterion for making the locally optimal choice at each round. As a result, it is difficult to know whether the failures are caused by the optimization criterion or the setting of parameters. Different from previous studies, we formulate implicit SRD as a process of selecting and ranking k exemplar documents through integer linear programming (ILP). The key idea is that: *for a specific query, we expect to maximize the overall relevance of the k exemplar documents. Meanwhile, we wish to maximize the representativeness of the selected exemplar documents with respect to the non-selected documents. Intuitively, if the selected exemplar documents concisely represent the entire set of documents, the novelty and diversity will naturally arise.* Moreover, we propose two approaches *ILP4ID* (Integer Linear Programming for Implicit SRD) and *AP4ID* (Affinity Propagation for Implicit SRD) for solving the proposed formulation of implicit SRD. In particular, ILP4ID appeals to the strategy of bound-and-branch and is able to obtain the optimal solution. AP4ID being an approximate method transforms the target problem as a maximum-a-posteriori inference problem, and the message passing algorithm is adopted to find the solution. Furthermore, we investigate the differences and connections between the proposed models and prior models by casting them as

*Corresponding author
Email addresses: yuhaitao@slis.tsukuba.ac.jp (Hai-Tao Yu), adam@dl.kuis.kyoto-u.ac.jp (Adam Jatowt), rblanco@udc.es (Roi Blanco), hideo@slis.tsukuba.ac.jp (Hideo Joho), joemon.jose@glasgow.ac.uk (Joemon M. Jose), Long.Chen@glasgow.ac.uk (Long Chen), f.yuan.1@research.gla.ac.uk (Fajie Yuan)

different variants of the cluster-based paradigm for implicit SRD. To validate the effectiveness and efficiency of the proposed approaches, we conduct a series of experiments on four benchmark TREC diversity collections. The experimental results demonstrate that: (1) The proposed methods, especially *ILP4ID*, can achieve substantially improved performance over the state-of-the-art unsupervised methods for implicit SRD. (2) The *initial runs*, *the number of input documents*, *query types*, *the ways of computing document similarity*, *the predefined cluster number* and *the optimization algorithm* significantly affect the performance of diversification models. Careful examinations of these factors are highly recommended in the development of implicit SRD methods. Based on the in-depth study of different types of methods for implicit SRD, we provide additional insight into the cluster-based paradigm for implicit SRD. In particular, how the methods relying on greedy strategies impact the performance of implicit SRD, and how a particular diversification model should be fine-tuned.

Keywords: Cluster-based IR, implicit SRD, integer linear programming, affinity propagation

---

1. Introduction

Accurately and efficiently satisfying user information requests by search engines is still far from being a solved problem. A key issue is that users tend to submit short and often ambiguous or underspecified queries; for example, the common query *Lord of the Rings* may refer to the movie series or the book. Furthermore, when it comes to the movies, users may be interested in a variety of possible aspects including the cast, reviews, price of dvds, etc. Correctly determining users' preferences is however difficult. As a remedy, one possible solution is to apply *search result diversification* (SRD) technique, which relies on providing a diversified result set so as to maximize the likelihood that an average user will find documents relevant to her particular search need. Considering the above-mentioned movie example such solution should generate an optimized result list that covers the key possible aspects like *book*, *movie*, *dvd*. According to *whether the subtopics (i.e., different information needs) underlying a query are given beforehand or not*, the task of SRD can be distinguished into *implicit SRD* and *explicit SRD*. The distinguishing characteristics of the implicit SRD is that the possible subtopics underlying a query are *unknown*. Noteworthy, finding a group of subtopic strings that covers well all the possible information needs behind the query is a challenging task. In most realistic scenarios explicit subtopics are not available [1], neither is the training data for supervised methods (e.g., [2, 3, 4, 5, 6, 7, 8, 9]). In such scenarios the technique of implicit SRD is then commonly used, instead, for the purpose of diversifying the results and satisfying users' search intents. Consequently, in this paper *we focus on the implicit diversification methods* instead of the explicit SRD or on supervised methods for search result diversification.

The state-of-the-art methods for implicit SRD can be differentiated accord-

ing to their solutions for the following key problems: (1) how to represent diversity; (2) how to balance the notions of the relevance and diversity, and (3) how to generate the final result list. For example, the well-known Maximal Marginal Relevance (*MMR*) model [10] measures the diversity of a document d_i based on the maximum similarity between d_i and the previously selected documents to approach the first challenge and in order to balance relevance and diversity, *most of the existing methods utilize a trade-off parameter* λ. Finally, for generating the desired result list *the common practice is using the reedy strategy that follows a heuristic criterion of making the locally optimal choice at each round* [10, 11, 12, 13].

Despite the success achieved by the state-of-the-art methods, there are several issues and problems that need further exploration. The key underlying drawback of the state-of-the-art approaches is that the commonly used greedy strategy works well on the premise that the preceding choices are optimal or close to the optimal solution. However, in many cases, this strategy fails to guarantee the optimal solution. A natural question arises then: *to what extent does the greedy solution affect the performance of implicit SRD*? Moreover, when conducting experimental analysis, a single weighting model (e.g., language model with Dirichlet smoothing [14]) is commonly adopted to perform the initial retrieval of results. Since the initially retrieved documents (e.g., top-m documents) are then further used to test diversification models, different initial runs should have significant impact on the performance of these diversification models. Furthermore, the effects of the key parameters: m (i.e., the number of used documents) and k (i.e., the predefined cluster number) on the performance of a diversification model are crucial and should be explored in details. The same criterion applies to the examination of the effect of the different query types on the quality of results. To the best of our knowledge all these key points have not been sufficiently investigated in most of the previous studies on implicit SRD.

The aforementioned drawbacks motivate us to address the task of the implicit SRD in a novel way. In particular, we propose a concise integer linear programming (ILP) formulation for implicit SRD. Based on such formulation, we introduce two different approaches to find the desired solution. One is an approximate method based on message passing called *AP4ID*. The other is an exact method, called *ILP4ID*, which is based on the strategy of bound-and-branch, under which the exactly optimal solution can be obtained and validated. Finally, we compare the effectiveness of the proposed approaches against the state-of-the-art algorithms using the standard TREC diversity collections. The experimental results prove that both *AP4ID* and *ILP4ID* can improve performance over the baseline methods in terms of the standard diversity metrics.

The main contributions of this paper are as follows:

1. We present a concise ILP formulation for implicit SRD which allows for the exact solution of the objective function (Eq. 12) to be obtained. On the one hand, two different approaches *AP4ID* and *ILP4ID* are proposed to find the desired solution. The proposed method *ILP4ID* can lead to substantially improved performance than the state-of-the-art unsupervised

methods. The experimental results also demonstrate how much accuracy has been lost due to the usage of an approximation method (e.g., compared with the method [13]). On the other hand, the flexibility of the proposed formulation allows for further extensions by simply altering the constraints.

2. Different from prior studies, we thoroughly investigate the effects of a series of factors on the performance of a diversification model. Our main finding is that some factors, such as *different initial runs*, *the number of input documents*, *query types*, *the ways of computing document similarity* and *the predefined cluster number* greatly affect the effectiveness of diversification models for implicit SRD. Careful examinations of these factors are highly recommended in the development of implicit SRD methods. Based on the systematic evaluation of different variants of the cluster-based paradigm for implicit SRD, we provide additional insight into the cluster-based paradigm for implicit SRD. In particular, how the methods relying on greedy strategies impact the performance of implicit SRD, and how a particular diversification model should be fine-tuned.

In this paper, we extend the conference version [15] in multiple ways. First of all, we include a new approximate approach *AP4ID* for solving the proposed formulation for implicit SRD (cf. Section 3.1). Although *AP4ID* outperforms *ILP4ID* and other baseline methods only under particular cases (cf. Section 4.3.3), it sheds light on devising more efficient ways for solving implicit SRD. Secondly, we expand our experimental evaluation, reporting additional discussion of the results on the comparison with existing state-of-the-art methods. Thirdly, additional experiments are conducted to highlight the effect of the predefined cluster number on the diversification performance.

The rest of the paper is structured as follows. In the next section, we first describe the Affinity Propagation algorithm, and survey the well-known approaches for search result diversification. In Section 3, we formulate implicit SRD as an ILP problem, then *ILP4ID* and *AP4ID* are proposed. A series of experiments are conducted and discussed in Section 4. We summarize the key findings in Section 5. In Section 6, we conclude the paper and discuss the possible aspects for future work.

2. Related Work

This work is connected to two different research areas: *data clustering* and *information retrieval* (IR). In this section, we first provide a brief description of the popular Affinity Propagation (AP) algorithm for exemplar-based clustering, which lays the groundwork for the proposed methods. Then, we concisely survey the popular methods for explicit SRD and the supervised methods for search result diversification. Finally, we discuss the typical approaches for cluster-based IR and implicit SRD. Due to space constraints, we refer the reader to [16, 17] for a detailed overview of cluster-based IR and search result diversification.

2.1. Affinity Propagation for Clustering

The AP algorithm [18] has been deployed and extended in many research fields, such as detecting drug sensitivity [19], image categorization [20] and image segmentation [21].

Under the AP algorithm, clustering is viewed as identifying a subset of exemplars (i.e., representative items) given m items. A symmetric matrix U representing the pairwise similarity of each pair of items is predefined. Moreover, the diagonal values of U denotes the prior beliefs of the m items in how likely each item is to be selected as an exemplar. The m items are divided into two disjoint sets, one set consists of exemplar items, the other set consists of non-exemplar items. The AP algorithm assigns each non-exemplar item to an exemplar item, the objective is to maximize the sum of similarities between non-exemplar items and their assigned exemplar items.

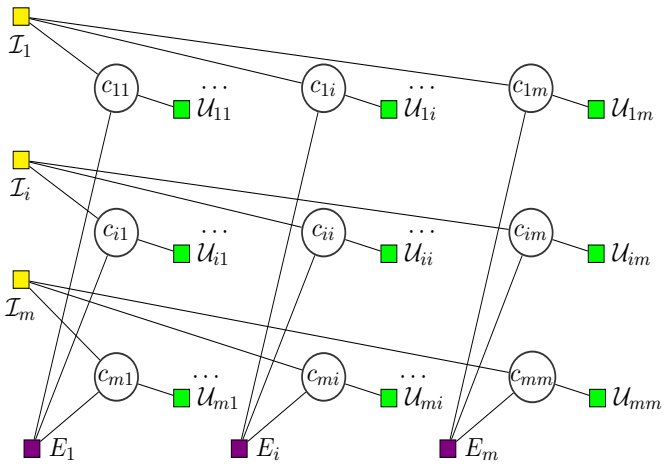


Figure 1: Factor graph representation of AP.

Fig. 1 shows the factor graph representation of AP, where the binary variable $c_{ij:i\neq j}$ denotes whether the j-th item chooses the i-th item as its exemplar. The factor nodes are defined as follows:

$$\mathcal{U}(c_{ij}) = c_{ij}U_{ij} \tag{1}$$

$$\mathcal{I}_i(c_{i:}) = \begin{cases} -\infty & if \ \sum_{j=1}^{m} c_{ij} \neq 1 \\ 0 & otherwise \end{cases} \tag{2}$$

$$\mathcal{E}_j(c_{:j}) = \begin{cases} -\infty & if \, c_{jj} \neq 1 \, but \, \exists i : c_{ij} = 1 \\ 0 & otherwise \end{cases} \tag{3}$$

For convenience, let $c_{:j} = \{c_{1j}, ..., c_{mj}\}$ and $c_{i:} = \{c_{i1}, ..., c_{im}\}$. The factor function $\mathcal{I}_i(c_{i:})$ enforces the constraint that each item must and can only select

one exemplar. The factor function $\mathcal{E}_j(c_{:j})$ enforces the consistence constraint: the j-th item must choose itself as an exemplar if there is one or more items that have the j-th item as their exemplar. Finally, the objective function of AP is expressed as Eq. 4, namely, a problem of searching for the optimal setting of $\mathbf{c}$ that maximizes the sum of similarities between exemplar items and non-exemplar items, while respecting the constraints (Eqs. 2 and 3).

$$\sum_{i,j}^{m} \mathcal{U}(c_{ij}) + \sum_{i=1}^{m} \{\mathcal{I}_i(c_{i:}) + \mathcal{E}_i(c_{:i})\} \tag{4}$$

To solve the objective function by Eq. 4, Givoni and Frey [22] use the max-sum message passing algorithm and show how to perform inference by only using two types of messages. The *responsibility* message ρ_{ij} (Eq. 5) refers to the message sent from a variable node c_{ij} to the factor node E_j and it is interpreted as the accumulated evidence for how well-suited the j-th item is to serve as the exemplar for the i-th item.

$$\rho_{ij} = U_{ij} - \max_{k \neq j} \{U_{ik} + \alpha_{ik}\} \tag{5}$$

The *availability* message α_{ij} (Eq. 6) refers to the message sent from a candidate exemplar (the j-th item) to the i-th item. It reflects the accumulated evidence for how appropriate it would be for the i-th item to choose the j-th item as its exemplar.

$$\alpha_{ij} = \begin{cases} \sum_{k \neq j} \max\{\rho_{kj}, 0\} & if\, i = j \\ \min\{0, \rho_{jj} + \sum_{k \notin \{i,j\}} \max\{\rho_{kj}, 0\}\} & if\, i \neq j \end{cases} \tag{6}$$

After iteratively updating these two messages, AP determines the exemplars by a combined usage of responsibility and availability messages. For instance, the value of k that maximizes $\rho_{ik} + \alpha_{ik}$ either identifies the exemplar itself if $k = i$, or identifies the k-th item who is the exemplar of the i-th item[1].

Inspired by AP, we formulate the implicit SRD as a process of selecting and ranking exemplar documents. Two different approaches *AP4ID* and *ILP4ID* are proposed to get the solution. The approximate approach *AP4ID* is essentially a modification of the original AP algorithm. *ILP4ID* builds upon the *bound-and-branch method* strategy to obtain the optimal solution. Thus *ILP4ID* can be used as a complementary method, which also provides clues for solving data clustering problems when the exact solution is to be expected.

2.2. Explicit SRD and Supervised Methods

The methods [11, 12, 23, 24, 25] for explicit SRD assume that the possible aspects representing different information needs of a query are given beforehand. For example, the xQuAD framework [11] downweights each subtopic

[1]For detailed information please refer to [18, 22].

based on the degree of its relevance to the already selected documents, thus the subtopics with less relevant documents will have a higher priority in the next round. Dang and Croft [12] studied result diversification by considering the notion of proportionality, they argued that the number of documents assigned to a specific subtopic should be proportional to this subtopic's popularity. At each step, the document that best maintains the overall proportionality is selected, and then the so-called *quotient* of the corresponding subtopic will be updated. Hu et al. [25] explored how to incorporate the hierarchical relationships among pre-collected subtopics of a query to perform search result diversification. Different from the aforementioned methods, the studies [26, 27] perform search result diversification by aggregating the output of a set of rankers optimized for diversity or not. The work by Liang et al. [26] showed that fusing results of different rankers does aid diversification. Moreover, Liang et al. [28] also explored how to perform search result diversification for streams of short texts (e.g., Twitter messages). The experimental results show that diversification for streams of short texts is quite different from diversification for long documents, and specific models have to be carefully designed.

Another popular direction is to use machine learning technologies to train the diversification model [4, 5, 6, 7, 8, 9, 29]. The advantages are straightforward. On one hand, it is easy to incorporate a large number of features into a specific diversification method. On the other hand, decades of work on machine learning can be leveraged to optimize the ranking functions. Compared with the unsupervised methods for either explicit SRD or implicit SRD, we can observe that the diversification models [4, 5, 6, 7, 8, 9, 29] based on machine learning technologies can achieve significantly improved performance.

However, there are some major challenges when deploying either the explicit methods or the supervised approaches for search result diversification. First, it is not easy to find a group of subtopic strings that accurately reflect the possible information needs underlying an ambiguous and/or underspecified query. In most realistic scenarios explicit subtopics are not available [1]. Second, gathering sufficient labeled data is often a challenging task, which requires considerable human efforts. Consequently, in this paper we mainly focus on the implicit diversification methods rather than the explicit approaches and supervised models for search result diversification.

2.3. Cluster-based IR and Implicit SRD

We begin by introducing some notations that are used throughout this paper. For a given query q, $D = \{d_1, ..., d_m\}$ represents the top-m documents of an initial retrieval run. $r(q, d_i)$ denotes the relevance score of a document d_i w.r.t. q. The similarity between two documents d_i and d_j is denoted as $s(d_i, d_j)$.

A large body of work on cluster-based approaches for IR build upon the *cluster hypothesis* [30], which states that "closely associated documents tend to be relevant to the same requests". Some cluster-based methods rely on document clusters created offline by using the entire corpus [31, 32]. The methods utilizing *query-specific document clusters* are more popular, where the clusters are generated from documents by an initial retrieval performed in response to

a query. For instance, [33, 34] propose to enhance the ad-hoc retrieval performance, where document clusters are used to smooth documents' representations (e.g., language models). Recently, Levi et al. [35] investigated how to apply cluster-based document retrieval or standard document retrieval in a selective manner on a per-query basis. Meanwhile, the cluster-based retrieval paradigm has been explored in the context of search result diversification, such as [36] and [37]. Raiber and Kurland [37] studied how to incorporate various types of cluster-related information based on Markov Random Fields. Naini et al. [38] explored the practical issues when performing distributed diversification.

Regarding implicit SRD, in order to obtain the optimal ranked list L^*, the most intuitive way is to apply the *greedy best first strategy*. At the beginning, this strategy initializes L with the most relevant document d_1^*, and then it selects the subsequent documents one by one via a specific heuristic criterion:

$$d_j^* = \underset{d_j \in D \setminus L_{j-1}}{\operatorname{argmax}} \{\lambda r(q, d_j) + (1-\lambda)\mathcal{W}(d_j, L_{j-1})\} \tag{7}$$

where $L_{j-1} = \{d_1^*, ..., d_{j-1}^*\}$, $\mathcal{W}(d_j, L_{j-1})$ measures how far d_j disperses w.r.t. L_{j-1}. At every round, it involves examining each document that has not been selected, computing a gain using the above heuristic criterion, and selecting the one with the maximum gain. A typical instance of this strategy is the *MMR* model [10], in which $\mathcal{W}(d_j, L_{j-1})$ is defined as - $\max_{d_i \in L_{j-1}} s(d_i, d_j)$. In other words, the diversity under *MMR* is measured through the maximum similarity between d_j and the previously selected documents. Furthermore, Guo and Sanner [39] present a probabilistic latent view of MMR, where the need of manually tuning λ is removed. Later on, the greedy optimization of *Exp*-1-*call*@k [40] for implicit SRD was proposed. The well-known Modern Portfolio Theory (MPT) [41] model takes into account the expected relevance and relevance variance of a document, and the correlations with the already selected documents. It sequentially selects documents that maximize the following criterion:

$$\mathbb{E}(d_k) - b \cdot w_k \cdot \sigma_k^2 - 2b \sum_{i=1}^{k-1} w_i \cdot \sigma_i \cdot \sigma_k \cdot \rho_{ik} \ , \tag{8}$$

where $\mathbb{E}(d_k)$ is the expected relevance of d_k, and σ_k is the standard deviation, w denotes the rank-specific weigh, and ρ_{ik} denotes the correlation coefficient between d_i and d_k.

Another line of studies (referred to as *top-k retrieval* in [13, 42, 36]) for implicit SRD performs a two-step process. The first step is to select an optimal subset $S \subset D$ of k documents according to a specific objective function. At the second step, the selected documents in S are ordered in a particular way, e.g., in a decreasing order of relevance. Moreover, Gollapudi and Sharma [42] propose a set of natural axioms analyzing the properties of a diversification function. A more general model (referred to as Desirable Facility Placement *DFP*) by

Zuccon et al. [13] is given as:

$$S^* = \underset{S \subset D, |S|=k}{\operatorname{argmax}} \lambda \cdot \mathcal{R}(S) + (1-\lambda) \cdot \mathcal{D}(S) \tag{9}$$

$$\mathcal{R}(S) = \sum_{d \in S} r(d) \tag{10}$$

$$\mathcal{D}(S) = \sum_{d' \in D \setminus S} \max_{d \in S} \{s(d, d')\} \ , \tag{11}$$

where $\mathcal{R}(S)$ denotes the overall relevance. $\mathcal{D}(S)$ denotes the diversity of the selected documents, which is captured by measuring the representativeness of the selected documents w.r.t. the non-selected ones and $\lambda \in [0, 1]$ is a trade-off parameter. To obtain S^*, they use the *greedy best k strategy*. It initializes S with an arbitrary solution (e.g., the k most relevant documents), and then iteratively refines S by swapping a document in S with another one in $D \setminus S$. At each round, interchanges are made only when the current solution can be improved. The process terminates after convergence or after a fixed number of iterations.

Our work is a further endeavor to the cluster-based retrieval paradigm. The studies most related to ours are [23, 13, 36, 37]. However, the ILP formulation by Yu and Ren [23] is proposed to perform explicit SRD, which requires pre-collected subtopics as the input. For implicit SRD, the methods [13, 36, 37] appeal to approximate methods for generating clusters. Our formulation of implicit SRD based on ILP allows to obtain the optimal solution, which makes it possible to investigate how much accuracy has been lost due to approximations (e.g., compared with *AP4ID* and *DFP*).

A number of successful ILP formulations have been developed for natural language processing tasks, such as semantic role labelling [43], syntactic parsing [44] and summarisation [45]. The ILP formulation we present is, to the best of our knowledge, the first one for implicit SRD. In fact, the above ILP formulation is quite flexible, and different variants can be derived by simply changing the constraints. For example, when removing the constraint by Eq. 16, the relevance expression (by Eq. 13) and the coefficients m-k and k in Eq. 12, the above formulation boils down to an equivalent ILP formulation of AP. It would be interesting to make an in-depth comparison between AP and its ILP formulation in the future, which helps to know *to what extent AP diverges from the optimal solution*.

3. Proposed Methods

In this section, we first describe the approaches *ILP4ID* and *AP4ID* proposed for addressing implicit SRD. We then discuss the differences and connections between the proposed approaches and the previous methods by viewing them as different variants of the cluster-based paradigm for implicit SRD.

3.1. ILP Formulation for Implicit SRD

We formulate the task of implicit SRD as a process of selecting and ranking k exemplar documents from the top-m documents of an initial retrieval. We call a document *exemplar* if it is selected to represent a group of documents based on some measure of similarity. On one hand, we expect to maximize the overall relevance of the k exemplar documents w.r.t. a query. On the other hand, we wish to maximize the *representativeness* of the exemplar documents w.r.t. the non-selected documents. The underlying intuition is that if the selected exemplars concisely represent the entire set of documents, the novelty and diversity will naturally arise.

To clearly describe the way of identifying the expected k exemplar documents, we introduce the binary square matrix $\mathbf{x} = [x_{ij}]_{m \times m}$ such that $m = |D|$, x_{ii} indicates whether document d_i is selected as an exemplar or not, and $x_{ij:i \neq j}$ indicates whether document d_i "chooses" document d_j as its exemplar. The process of selecting k exemplar documents is then expressed as the following ILP problem:

$$\max_{\mathbf{x}} \lambda \cdot (m\text{-}k) \cdot \mathcal{R}^{'}(\mathbf{x}) + (1\text{-}\lambda) \cdot k \cdot \mathcal{D}^{'}(\mathbf{x}) \tag{12}$$

$$\mathcal{R}^{'}(\mathbf{x}) = \sum_{i=1}^{m} x_{ii} \cdot r(q, d_i) \tag{13}$$

$$\mathcal{D}^{'}(\mathbf{x}) = \sum_{i=1}^{m} \sum_{j=1:j \neq i}^{m} x_{ij} \cdot s(d_i, d_j) \tag{14}$$

$$s.t.\ x_{ij} \in \{0, 1\},\ i \in \{1, ..., m\}, j \in \{1, ..., m\} \tag{15}$$

$$\sum_{i=1}^{m} x_{ii} = k \tag{16}$$

$$\sum_{j=1}^{m} x_{ij} = 1, i \in \{1, ..., m\} \tag{17}$$

$$x_{jj} - x_{ij} \geq 0, i \in \{1, ..., m\}, j \in \{1, ..., m\} \tag{18}$$

In particular, the restriction given by Eq. 16 guarantees that k documents are selected. The restriction by Eq. 17 means that each document must have only one representative exemplar. The constraint given by Eq. 18 enforces that if there is one document d_i selecting d_j as its exemplar (i.e., $x_{ij} = 1$), then d_j must be an exemplar (i.e., $x_{jj} = 1$). $\mathcal{R}^{'}(\mathbf{x})$ represents the overall relevance of the selected exemplar documents. $\mathcal{D}^{'}(\mathbf{x})$ denotes diversity. In other words, the diversity is expressed through selecting documents that represent the intrinsic diverse information revealed by the input documents. In view of the fact that there are k numbers (each number is in $[0, 1]$) in the relevance part $\mathcal{R}^{'}(\mathbf{x})$, and m-k numbers (each number is in $[0, 1]$) in the diversity part $\mathcal{D}^{'}(\mathbf{x})$, the coefficients m-k and k are added in order to avoid possible skewness issues, especially when $m \gg k$. Finally, the two parts are combined through the parameter λ

as shown in Eq. 12. As shown by previous studies [17, 46], the diversification problem is NP-hard. It is hardly surprising that the proposed ILP formulation (Eqs. 12-18) for implicit SRD is also NP-hard. Fortunately, ILPs are a well studied optimization problem and a number of mature techniques, such as the cutting-plane strategy [47] and the branch-and-bound strategy [48], have been developed to obtain the optimal solution.

Given the above ILP formulation (Eqs. 12-18) for implicit SRD, we adopt two different approaches to find the desired solution. The first approach appeals to the strategy of bound-and-branch, which is a traditional way of solving ILP problems. In particular, the off-the-shelf Gurobi solver (Version 6.5.1)[2] is adopted in this study. Once the k exemplar documents are selected, they are further ranked in the decreasing order of their respective contributions to objective function given by Eq. 12. We denote this approach as *ILP4ID*, namely, *a naive integer linear programming approach for implicit SRD*. The second approach relying on message passing extends the AP algorithm by incorporating more factors, such as the part of measuring the overall relevance by Eq. 13 and the restriction by Eq. 16. We detail this approach in Section 3.2.

3.2. Affinity Propagation for Implicit SRD

Besides the strategy of bound-and-branch for solving the proposed ILP formulation (Eqs. 12-18) for implicit SRD, we develop an approximate method based on message passing. The key idea is to transform the proposed ILP formulation as a maximum-a-posteriori inference problem.

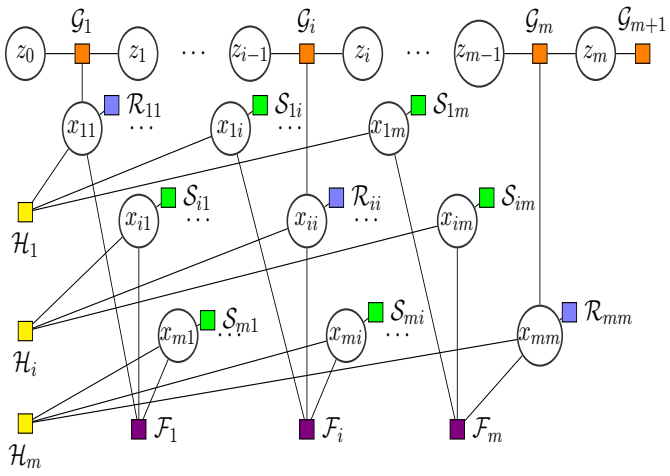


Figure 2: Factor graph of implicit SRD.

According to the study [18, 22], the derivation of AP algorithm is given by rewriting the clustering objective as minimizing a particular energy function,

[2]http://www.gurobi.com/index

where the max-sum algorithm can be used to search over configurations in the factor graph. In view of the fact that there is an equivalent ILP expression for the clustering problem of AP, analogously, we can modify the original AP algorithm for solving the proposed ILP formulation for implicit SRD. Specifically, the problem of maximizing Eq. 12 subject to constraints (Eqs. 15, 16, 17 and 18) can be expressed by a factor graph in Fig. 2, where the global objective function is factored into simpler local functions.

Specifically, the factor potentials are given as follows:

$$\mathcal{R}_{ii}(x_{ii}) = x_{ii} \cdot \lambda \cdot (m-k) \cdot r(q, d_i) \tag{19}$$

$$\mathcal{S}_{ij}(x_{ij:j\neq i}) = x_{ij} \cdot (1-\lambda) \cdot k \cdot s(d_i, d_j) \tag{20}$$

$$\mathcal{H}_i(x_{i:}) = \begin{cases} -\infty & if \sum_{j=1}^{m} x_{ij} \neq 1 \\ 0 & otherwise \end{cases} \tag{21}$$

$$\mathcal{F}_j(x_{:j}) = \begin{cases} -\infty & if\, x_{jj} \neq 1\, but\, \exists i : x_{ij} = 1 \\ 0 & otherwise \end{cases} \tag{22}$$

$$\mathcal{G}_i(x_{ii}, z_{i-1}, z_i) = \begin{cases} 0 & if\, z_i = z_{i-1} + x_{ii} \\ -\infty & otherwise \end{cases} \tag{23}$$

The factor function $\mathcal{R}_{ii}(i \in \{1, ..., m\})$ is specific to the diagonal variable x_{ii}, which reflects the relevance of document d_i if it is selected. The factor function $\mathcal{S}_{ij}(i \in \{1, ..., m\} : i \neq j)$ is specific to the non-diagonal variable x_{ij}, which reflects the similarity between documents d_i and d_j. $\mathcal{H}_i(x_{i:})$, $\mathcal{F}_j(x_{:j})$ and $\mathcal{G}_i(x_{ii}, z_{i-1}, z_i)$ are constraint factors. In particular,

(1) Like the factor function $\mathcal{I}_i(c_{i:})$ of the AP algorithm, $\mathcal{H}_i(x_{i:})$ enforces the constraint that each document can only select one document as its representative exemplar.

(2) Like the factor function $\mathcal{E}_j(c_{:j})$ in the AP algorithm, $\mathcal{F}_j(x_{:j})$ enforces a consistence constraint, i.e., document d_j must choose itself as its representative exemplar if there is one or more documents that choose d_j as their exemplar.

(3) To fulfill the constraint of Eq. 16, i.e., the total number of selected exemplars is exactly the predefined size k, we incorporate a Hidden Markov Model (HMM) model using the strategy proposed by Lazic [49]. Under this model, $z_i(i = 0, ..., m)$ are hidden variables, $x_{ii}(i = 1, ..., m)$ is interpreted as noisy observations. By setting $z_0 = 0$ and enforcing the constraint factor $\mathcal{G}_i(x_{ii}, z_{i-1}, z_i)$ as in Eq. 23, the hidden variable $z_m = \sum_{i=1}^{m} x_{ii}$ corresponds to the total number of selected documents. Moreover, an arbitrary potential (i.e., the size of S) on z_m is incorporated via the factor G_{m+1}.

Note that it is possible to build an equivalent factor graph representation w.r.t. the above ILP in a different way (see, for instance, the study by Dueck [50]). A further exploration is beyond the scope of this paper.

Continuing, the graphical model in Fig.2 together with Eqs. 19-23 result in the following max-sum objective function:

$$\begin{aligned}\mathcal{M}(\mathbf{x}) = &\sum_{i=1}^{m}\sum_{j=1:j\neq i}^{m}\mathcal{S}_{ij}(x_{ij}) + \sum_{i=1}^{m}\{\mathcal{R}_{ii}(x_{ii}) \\ &+ \mathcal{H}_i(x_{i:}) + \mathcal{F}_i(x_{:i}) + \mathcal{G}_i(x_{ii}, z_{i-1}, z_i)\}\end{aligned} \tag{24}$$

Now the problem of implicit SRD has been transformed into an inference problem over the binary random variables $\mathbf{x}$, i.e, searching a setting of $\mathbf{x}$ that achieves the largest joint likelihood. Following the work [18, 22], we employ the max-sum message passing algorithm to perform inference on the factor graph in Fig. 2. After iteratively propagating the messages after a fixed number of iterations or after the local decisions remain constant for some number of iterations, the optimal result of x_{ij} can be determined by collecting all the received messages and computing the beliefs w.r.t. each state. Fig. 3 shows the messages exchanged between variable nodes and factor nodes.

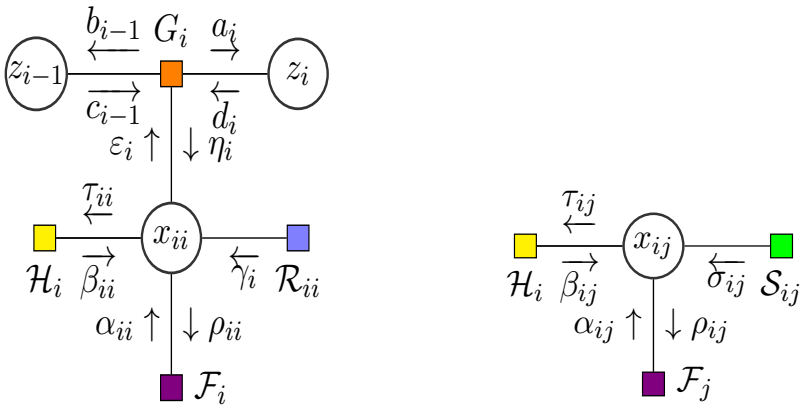


Figure 3: The messages passed between variable nodes and factor nodes.

The message update equations w.r.t. the messages in Fig. 3 are summarized as follows (the detailed derivation can be found in the *supplementary material*):

$$\alpha_{ij} = \begin{cases} i = j & \sum\limits_{i':i'\neq i} \max\{0, \rho_{i'j}\} \\ i \neq j & \min\{0, \rho_{jj} + \sum\limits_{i':i'\notin\{i,j\}} \max\{0, \rho_{i'j}\}\} \end{cases} \tag{25}$$

$$\rho_{ij} = \begin{cases} i = j & \eta_i + \lambda \cdot r(q, d_i) \text{-} \max\limits_{j':j'\neq i} \{(1\text{-}\lambda) \cdot s(d_i, d_{j'}) + a(i, j')\} \\ i \neq j & (1\text{-}\lambda) \cdot s(d_i, d_j) \text{-} \max\{\eta_i + \lambda \cdot r(q, d_i) + a(i, i), \\ & \max\limits_{j':j'\notin\{i,j\}} \{(1\text{-}\lambda) \cdot s(d_i, d_{j'}) + a(i, j')\}\} \end{cases} \tag{26}$$

$$\eta_i = \max_{z_i}\{b_i(z_i) + a_{i-1}(z_i - 1)\} - \max_{z_i}\{b_i(z_i) + a_{i-1}(z_i)\} \tag{27}$$

$$\begin{aligned} a_i(z_i) = &\max\{a_{i-1}(z_i - 1) + \alpha_{ii} - \max\{\eta_i + \lambda \cdot r(q, d_i) + a(i, i), \\ &\max_{j^{'}:j^{'}\notin\{i,j\}}\{(1-\lambda)\cdot s(d_i, d_{j^{'}}) + a(i, j^{'})\} + \lambda \cdot r(q, d_i),\ a_{i-1}(z_i)\} \end{aligned} \tag{28}$$

$$\begin{aligned} b_{i-1}(z_{i-1}) = &\max\{b_i(z_{i-1}), b_i(z_{i-1} + 1) + \alpha_{ii} \\ &- \max\{\eta_i + \lambda \cdot r(q, d_i) + a(i, i), \\ &\max_{j^{'}:j^{'}\notin\{i,j\}}\{(1-\lambda)\cdot s(d_i, d_{j^{'}}) + a(i, j^{'})\} + \lambda \cdot r(q, d_i)\} \end{aligned} \tag{29}$$

Algorithm 1 details the message passing algorithm for finding the optimal configuration of $\mathbf{x}$.

Algorithm 1 Message passing algorithm for implicit SRD

```
Initialize γ, σ, ρ ← 0, α ← 0, η ← 0, a ← 0, b ← 0,
  count ← 0;
while !convergence() and count ≤ threshold do
    η ← ψη + (1 − ψ)eval(Equation-27);
    ρ ← ψρ + (1 − ψ)eval(Equation-26);
    α ← ψα + (1 − ψ)eval(Equation-25);
    (a,b)-update;
    count++;
end while
---------------------------------------------
Procedure (a,b)-update
Initialize z_0 = 0,b_m(z_m) = G_{m+1}(z_m);
for i = 1:m do
    a_i(z_i) = eval(Equation-28)
end for
for i = m:1 do
    b_{i−1}(z_{i−1}) = eval(Equation-29)
end for
---------------------------------------------
```

The symbols in bold $\boldsymbol{\gamma}$, $\boldsymbol{\sigma}$, $\boldsymbol{\rho}$, $\boldsymbol{\alpha}$, $\boldsymbol{\eta}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ are used to represent matrices $[\gamma]_{1\times m}$, $[\sigma]_{m\times m}$, $[\rho]_{m\times m}$, $[\alpha]_{m\times m}$, $[\eta]_{1\times m}$, $[a]_{(m+1)\times(m+1)}$ and $[b]_{(m+1)\times(m+1)}$, respectively. The function *eval*() computes the result according to the input equation. The function *convergence*() is used to check whether the algorithm has converged or the local decisions stay constant. When updating the messages, it is important to take into account the message oscillations that arise in some circumstances. In particular, each message is set to ψ times its value from the previous iteration plus $1 - \psi$ times its prescribed updated value, where $\psi \in [0, 1)$. The updating procedure may be terminated after a fixed number of iterations (e.g., the *threshold* in Algorithm 1), or after the local decisions remain constant for certain number of iterations. In our experiments, we set

$threshold = 3000$ and $\psi = 0.85$. Following [23], we sum together all incoming messages for diagonal variables $\{x_{ii}\}$, and the corresponding belief values are used as indicators for ranking the selected documents.

Analogous to the AP algorithm, we also define ρ_{ij} and α_{ij} as the responsibility message and availability message, respectively. The update of availability message in Algorithm 1 is the same as in the AP algorithm. The update of responsibility message differs due to the incorporation of document relevance, as well as the HMM part of the factor graph for restricting the number of exemplars to be k.

In contrast to the AP algorithm, the proposed exemplar-selection process takes into account the relevance of selected exemplars w.r.t. a query, and restricts the number of selected exemplars to be k. However, both the AP algorithm and the proposed exemplar-selection process are NP-hard, which can be proved by reduction, for example, from the set cover problem. We denote this proposed approach as *AP4ID*, namely, *affinity propagation for implicit SRD*.

3.3. Models' Connections: The Perspective of Cluster Hypothesis

Looking back at the model *DFP* given by Eqs. 9, 10 and 11, if we view S as the set of exemplar documents, and $D \setminus S$ as the complementary set of non-selected documents, calculating $\max_{d \in S}\{s(d, d^{'})\}$ can be then interpreted as selecting the most representative exemplar $d \in S$ for $d^{'} \in D \setminus S$. Thus $\mathcal{D}(S)$ is essentially equivalent to $\mathcal{D}^{'}(\mathbf{x})$. In addition, $\mathcal{R}(S)$ is also equivalent to $\mathcal{R}^{'}(\mathbf{x})$. Therefore, *DFP* can be viewed as a special case of the ILP formulation for implicit SRD by Eqs. 12 - 18 when the coefficients m-k and k are not used. Since *ILP4ID* is able to obtain the exact solution w.r.t. the formulated objective function, its performance can be regarded as the *upper-bound* of *DFP*.

Moreover, the study by Zuccon et al. [13] also shows that there are close connections between *DFP* and the models like *MMR* [10], *MPT* [41] and Quantum Probability Ranking Principle (QPRP) [51]. Namely, *MMR*, *MPT* and *QPRP* can be rewritten as different variants of *DFP* (the reader can refer to [13] for detailed derivation). Analogously, *MMR*, *MPT* and *QPRP* can also be rewritten as different variants of our ILP formulation for implicit SRD by Eqs. 12 - 18. The detailed derivation can be obtained based on the work [13]. However, it should be noted that the space of feasible solutions for *DFP*, *ILP4ID* and *AP4ID* is different from the one for *MMR* or *MPT* or *QPRP*. This is because *DFP*, *ILP4ID* and *AP4ID* rely on a two-step diversification, while *MMR*, *MPT* and *QPRP* directly generate the ranked list of documents in a greedy manner.

Going further, according to the description in Section 3.1, effectively selecting exemplar documents is the core of both *ILP4ID* and *AP4ID* when performing implicit SRD, which can be interpreted as a clustering process whilst balancing both relevance and diversity. Therefore, in the context of implicit SRD, we utilize the aforesaid cluster hypothesis [30] as a general paradigm for comparing *MMR*, *MPT*, *QPRP*, *DFP*, *ILP4ID* and *AP4ID*, which makes it easy to understand the essence of each particular model. To the best of our knowledge, this paper is the first to investigate the unsupervised methods for implicit SRD based on the cluster hypothesis [30].

4. Experiments

In this section we report a series of experiments conducted to evaluate the performance of the proposed approaches by comparing them to the state-of-the-art implicit diversification methods. In the following, we first detail the test collections and the topics as well as the evaluation metrics used in the experiments. We then describe the configuration of each method to be evaluated, including the parameter setting and the ways of computing relevance scores, document similarity, etc. Finally, we describe the experimental results.

4.1. Test Collections and Metrics

Four standard test collections released in the diversity tasks of TREC Web Track from 2009 to 2012 are adopted for the experiments (50 queries per each year). Each query is structured as a set of a representative subtopics. Moreover, each query is further categorized as either "faceted" or "ambiguous" [52]. Queries numbered 95 and 100 in TREC 2010 are discarded due to the lack of judgment data, resulting in 198 queries being finally used.

The evaluation metrics we adopt are nERR-IA (normalized Intent-Aware Expected Reciprocal Rank) [46], α-nDCG (novelty-biased Discounted Cumulative Gain) [53], P-IA (intent-aware precision) [46] and Strec (subtopic recall) [54]. Noteworthy, *nERR-IA is used as the main effectiveness measure in this study same as in TREC Web Track*. Our rationale for the adopted metrics is that: nERR-IA and α-nDCG being representative position-sensitive metrics evaluate not only the diversity of a result list but also the ability of ranking relevant documents at top rank positions. Other similar metrics, such as MAP-IA and D#-nDCG [55], are not used. On the contrary, P-IA and Strec are not position-sensitive, which do not account for ranking a relevant document at position $r1$ or $r2$. Thus P-IA [46] and Strec are used to indicate the effectiveness of ranking relevant documents at top rank positions. In particular, the performance is evaluated using the top-20 ranked documents and the officially released script *ndeval*[3] with the default settings.

The ClueWeb09 Category B collection is indexed with the Terrier 4.0 platform[4]. Two ad-hoc weighting models are deployed for investigating the effect of initial runs, i.e., *language model with Dirichlet smoothing* [14] (denoted as *DLM*) and *BM25* [56] based on the default setting of Terrier 4.0.

4.2. Baselines and Model Configuration

The models *MMR* [10], *MPT* [41], 1-*call*@k [40] and *DFP* [13] introduced in Section 2 are used as baseline methods. Similar to 1-*call*@k, He et al. [36] have also used the Latent Dirichlet Allocation (LDA) topic model for document clustering, while Raiber and Kurland [37] have utilized a supervised method (i.e., SVM^{rank}) to utilize the cluster information. Due to these reasons, [36]

[3] http://trec.nist.gov/data/web10.html
[4] http://terrier.org/

and [37] are not compared in this study. When it comes to 1-*call*@k, we follow the same setting as in [40]. The LDA model (α=2.0, β=0.5) is trained based on the top-m results for each query and the obtained subtopic distributions are used for the similarity and diversity computation. In particular, the topic number is set to: 15 (when $m \leq 100$), 20 (when $100 < m \leq 300$), 25 (when $300 < m \leq 500$) and 40 (when $500 < m \leq 1000$). For *MPT*, the relevance variance between two documents is approximated by the variance with respect to their term occurrences. For *DFP* (the iteration threshold is $1,000$), *AP4ID* and *ILP4ID*, the k is initially set to 20. We examine the effect of different k settings in section 4.4.

For *MMR*, *DFP*, *AP4ID* and *ILP4ID*, we calculate the similarity between a pair of documents in two ways. One is the Jensen-Shannon Divergence (denoted as JSD) between document language models (e.g., DLM), which is a symmetric and smoothed version of KL divergence. The other is the cosine similarity based on tf-idf weight vectors (denoted as COS). The relevance values returned by *DLM* and *BM25* are then normalized to the range $[0, 1]$ using the MinMax normalization [57]. Using the same methods to compute both the relevance score and the document-to-document similarity in all the studied approaches enables us to conduct a fair comparison when investigating the impact of a specific component (e.g., the adopted optimization strategy) on the performance.

4.3. Experimental Evaluation

In the following experiments, we first describe the differences of the used initial runs by *DLM* and *BM25*. We then compare the optimization effectiveness between DFP and $ILP4ID$. Later, we investigate the models from different perspectives, including the effectiveness and efficiency.

4.3.1. Analysis of Initial Runs

Since the diversification models take the documents initially retrieved by either *DLM* or *BM25* as an input, a thorough exploration of the results when using *DLM* and *BM25* is necessary in order to understand the effectiveness of each diversification model. Table 1 shows the performance in terms of nERR-IA@20, α-nDCG@20, P-IA@20 and Strec@20, where the superscript $*$ indicates statistically significant differences when compared to the best result based on the Wilcoxon signed-rank test with $p < 0.05$.

Table 1: Performance of the initial retrieval. For each measure, the best result is indicted in bold.

Initial retrieval model	nERR-IA@20	α-nDCG@20	P-IA@20	Strec@20
DLM	0.1596*	0.2235*	0.0969*	0.4648*
$BM25$	**0.2168**	**0.2784**	**0.1155**	**0.5158**

From Table 1, we can observe that *BM25* has significantly better performance than *DLM*. To examine how many relevant documents there are in each initial run, we can look at Fig. 4, which shows the averaged number of documents that provide information relevant to at least one subtopic in the initial run. The x-axis denotes the cutoff values (i.e., the top-m documents to be used).

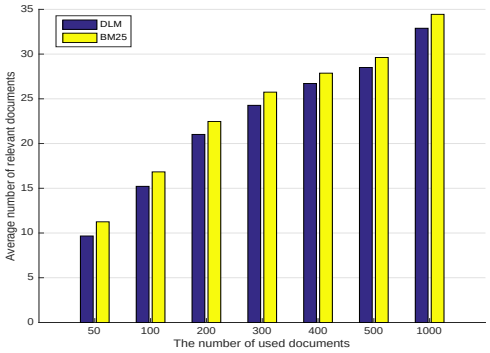


Figure 4: The statistics of the average number of relevant documents within the adopted initial runs.

Fig. 4 demonstrates that the results by *BM25* provide more relevant documents than that of *DLM*. At the same time, Fig. 4 also indicates to what extent the noisy documents will be mixed when we increase the number of used documents.

In the following experiments, the results of *DLM* and *BM25* are also used as naive baselines without diversification, which helps to show the *positive/negative* effects of deploying a diversification model. Using different ad-hoc weighting models, we can investigate the effect of an initial run. In particular, the experiments over the retrieval with *BM25* will allow to study the effect of using a high-quality initial run, while the ones with *DLM* will let us analyze the effect of providing a poor quality initial retrieval.

4.3.2. Optimization Effectiveness

Before investigating the performance of the aforementioned methods in performing implicit SRD, we first investigate the effectiveness of *ILP4ID*, *AP4ID* and *DFP* when solving the formulated objective functions (Eq. 9 and Eq. 12). In particular, we set $\lambda = 0$ for *ILP4ID*, *AP4ID* and *DFP*, and remove the coefficient k for both *AP4ID* and *ILP4ID*. Essentially, *ILP4ID*, *AP4ID* and *DFP* are enforced to work in the same way, namely by selecting predefined k exemplar documents without ranking.

For a given topic, we compute the representativeness (denoted as $\mathcal{D}$) of the subset S of k exemplar documents, which is defined as $\mathcal{D}(S)$ in Eq. 11. The higher the representativeness is, the more effective the adopted algorithm is when selecting the expected k exemplar documents. As an illustration, we use the top-50, 100 and 500 documents of the initial retrieval by *BM25*, respectively.

Fig. 5, Fig. 6 and Fig. 7 show the pair-wise comparisons of the performance of *ILP4ID*, *AP4ID* and *DFP* in finding the best k exemplars, respectively. Take Fig. 5 (a) for example, for each topic, we compute the difference between $\mathcal{D}_{ILP4ID}$ and $\mathcal{D}_{DFP}$ that is the difference between the representativeness by *ILP4ID* and the one by *DFP*. Specifically, the x-axis represents the queries, and

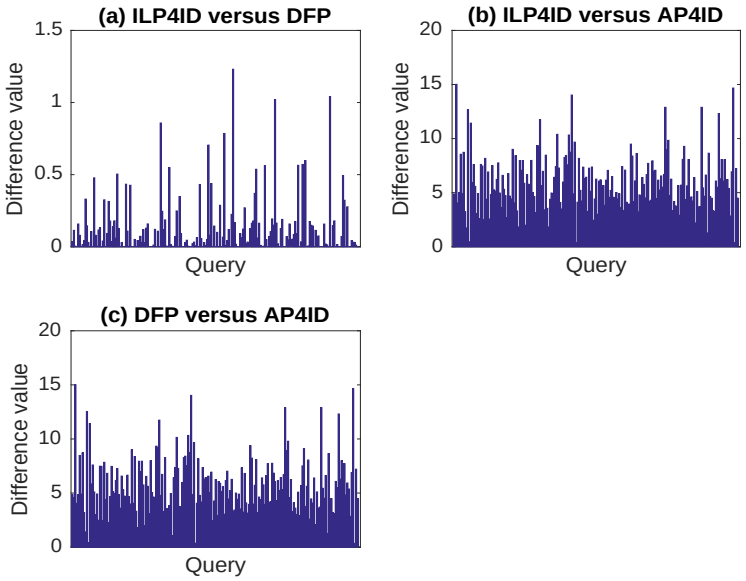


Figure 5: Optimization effectiveness comparison using top-50 documents.

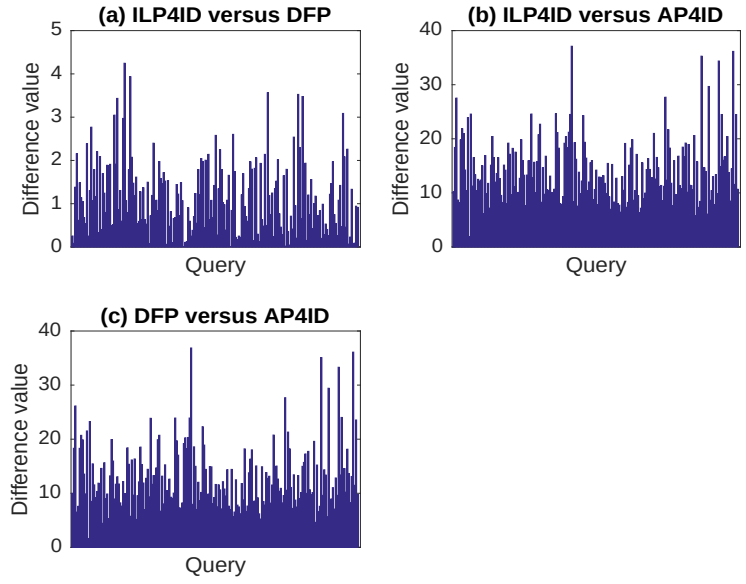


Figure 6: Optimization effectiveness comparison using top-100 documents.

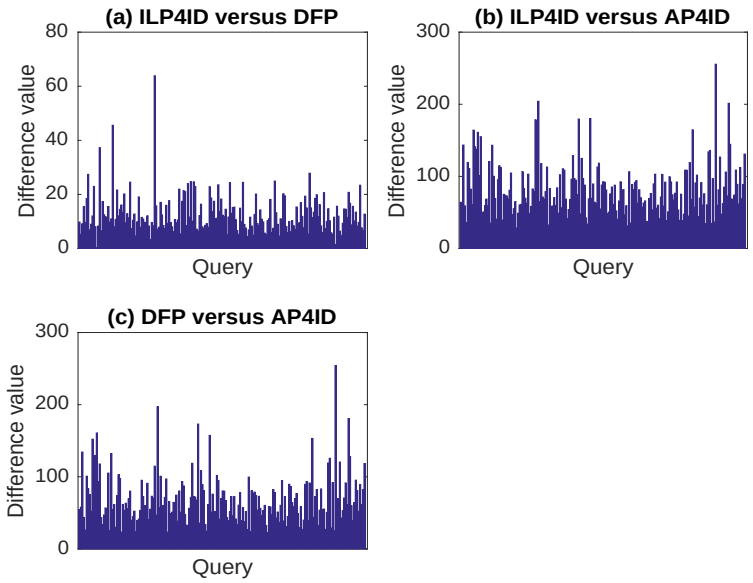


Figure 7: Optimization effectiveness comparison using top-500 documents.

the y-axis represents the difference of the representativeness (i.e., $\mathcal{D}_{ILP4ID} - \mathcal{D}_{DFP}$). Fig. 6 and Fig. 7 are obtained in a similar way.

From Fig. 5, Fig. 6 and Fig. 7, we can clearly observe that regardless of how many documents are used, $\mathcal{D}_{ILP4ID} - \mathcal{D}_{DFP} \geq 0$, $\mathcal{D}_{ILP4ID} - \mathcal{D}_{AP4ID} \geq 0$ and $\mathcal{D}_{DFP} - \mathcal{D}_{AP4ID} \geq 0$. In other words, *ILP4ID* outperforms both *AP4ID* and *DFP*, and *DFP* outperforms *AP4ID*. When the number of documents increases, so does the scale of representativeness difference values. For example, the representativeness difference values shown in Fig. 5 (a) lie in $[0, 1.5]$, while the representativeness difference values shown in Fig. 7 (a) lie in $[0, 80]$.

It is reasonable to say that the optimization effectiveness of *DFP* is comparable to *AP4ID* for tasks of using a small number of documents, since the difference values between $\mathcal{D}_{ILP4ID}$ and $\mathcal{D}_{DFP}$ are relatively small. On the contrary, for a moderately larger task the solution obtained by both *DFP* and *AP4ID*, especially *AP4ID*, significantly diverge from the optimal solution w.r.t. the objective formulation. This is because both *DFP* and *AP4ID* select exemplar documents in an approximation manner (i.e., *DFP* relies on the hill climbing algorithm, while *AP4ID* uses message propagation). In contrast, *ILP4ID* finds the exact solution based on the branch-and-bound algorithm. *ILP4ID* always returns the exact solution, while both *DFP* and *AP4ID* can not guarantee to find the optimal exemplar documents. Fig. 5, Fig. 6 and Fig. 7 essentially reveal that both *DFP* and *AP4ID* find a sub-optimal solution. Since the process of selecting exemplar documents plays a fundamental role for implicit SRD, the

effectiveness of both *DFP* and *AP4ID* is therefore greatly impacted, which is shown in terms of nERR-IA and α-nDCG in Sections 4.3.3, 4.3.4 and 4.3.5.

The dataset contains 141 faceted queries and 57 ambiguous queries. The TREC assumption [52] goes like this: For an ambiguous query that has diverse interpretations, users are assumed to be interested in only one of these interpretations. For a faceted query that reflects an underspecified subtopic of interest, the users are assumed to be interested in one subtopic, but they may still be interested in others as well. That is, heterogeneous documents providing more divergently relevant information are required for ambiguous queries.

To reveal the effect of query types on the optimization effectiveness, as an illustration, Fig. 8 shows how the difference between the representativeness by *ILP4ID* and the one by *DFP* vary with respect to faceted queries and ambiguous queries using the top-100 documents. Other comparisons are not presented due to the limited space and the fact that they show a similar trend. From Fig. 8, we can see that *DFP* performs slightly worse for faceted queries than for ambiguous queries when selecting exemplar documents.

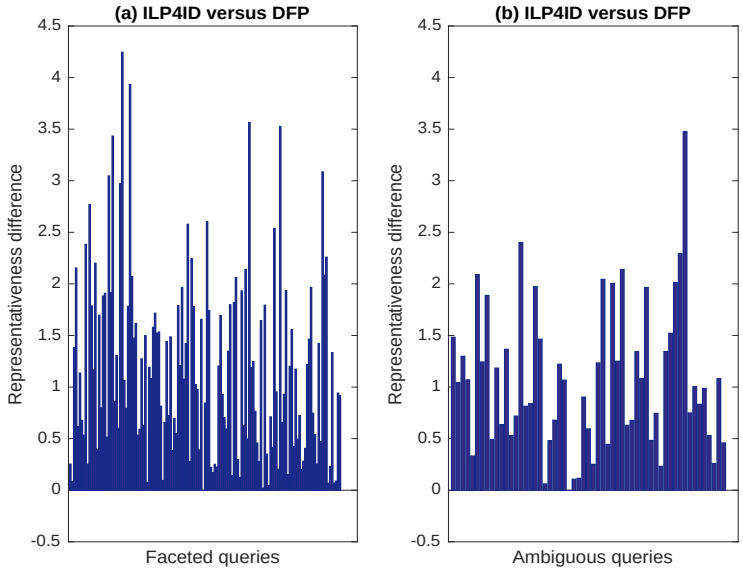
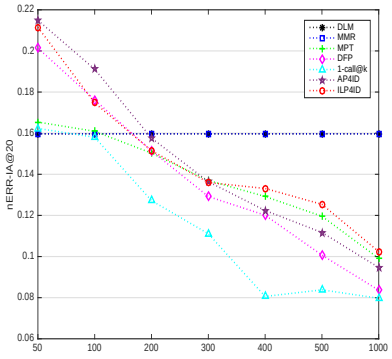


Figure 8: Optimization effectiveness comparison according to query types.

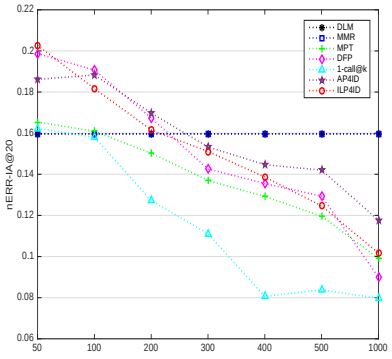
4.3.3. Implicit SRD Performance

In this section, we examine how the diversification models vary when we change the initial runs (i.e., *DLM* and *BM25*), the number of input documents (i.e., $m \in \{50, 100, 200, 300, 400, 500, 1000\}$ on the x-axis) and the ways for computing document similarity (i.e., *COS* and *JSD*).

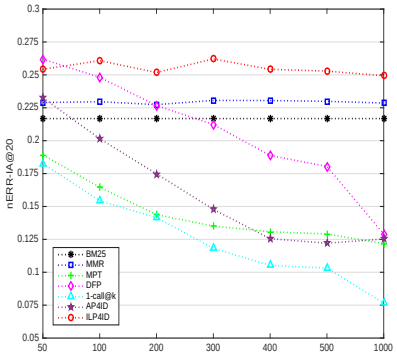
We use 10-fold cross-validation to tune the trade-off parameters, namely b for *MPT* and λ for *MMR*, *DFP*, *AP4ID* and *ILP4ID*. Particularly, we explore



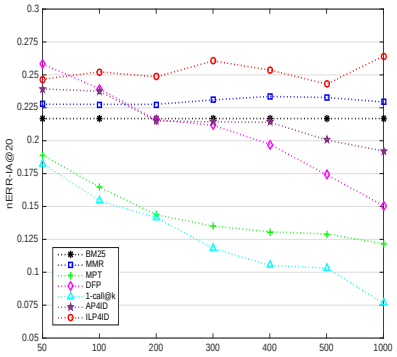
(a) Initial run: *DLM*; document similarity: COS.



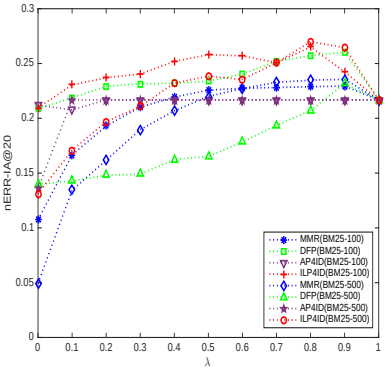
(b) Initial run: *DLM*; document similarity: JSD.



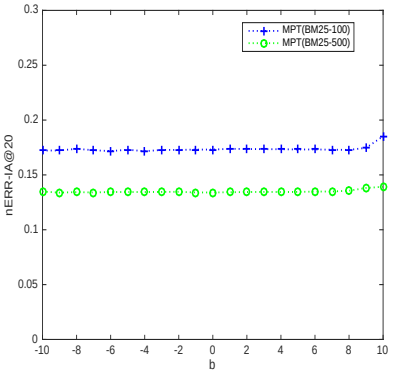
(c) Initial run: *BM25*; document similarity: COS.



(d) Initial run: *BM25*; document similarity: JSD.



(e) Comparison with changing λ.



(f) Performance variation of *MPT* w.r.t. b.

Figure 9: Cross-validation performance for implicit SRD (Figs 9(a)-9(d)), where the x-axis indicates the number of used documents. Per-λ comparison (Fig. 9(e)). Per-b performance of *MPT* (Fig. 9(f)).

the optimal results of *MMR*, *DFP*, *AP4ID* and *ILP4ID* by varying λ in the range $[0, 1]$ with a step of 0.1. We tune the b parameter of MPT with the range $[-10, 10]$, and a step of 1. The metric nERRIA@20 is used to determine the best results. The results are illustrated in Figs. 9(a)-9(d).

We note that in Figs. 9(a) and 9(b) the λ value of *MMR* that was determined via cross-validation is 1.0. Thus *MMR* fails to diversify the results (cf. Eq. 7). This is also why the performance curves of *MMR* basically overlap with those of *DLM* and *BM25*. The effect of tuning λ is detailed in Section 4.3.4.

At first glance, Figs. 9(a) and 9(b) based on *DLM* reveal that all the diversification models except *MMR* exhibit high effectiveness when using the smaller number of documents (top-50 documents). We also see that *DFP*, *AP4ID* and *ILP4ID* which belong to the cluster-based diversification paradigm are more effective than other formulations, such as MPT and 1-*call*@k when smaller numer of retrieved documents are used. This observation is consistent with the previous reports [13]. However, when we increase the initial number of retrieved documents, *MPT*, *DFP*, 1-*call*@k, *AP4ID* and *ILP4ID* consistently show decreased performance. In particular, when the number of used documents is quite large, these models can not even improve over the naive-baseline results with DLM. The plausible reason is that more *noisy documents* are included in larger document sets. This is actually supported by Fig. 4 which shows that relatively more non-relevant documents are included if we increase the retrieved documents threshold.

A closer look at Figs. 9(a) and 9(b) reveals that the ways of computing document similarity also affects the performance of *DFP*, *AP4ID* and *ILP4ID*, where the performance of MPT and 1-*call*@k can be used as a static reference since they do not rely on either *COS* or *JSD*. Note also that DFP occasionally achieves better results than *ILP4ID*, e.g., using top-100/200 documents in Fig. 9(b). This may result from the second ranking procedure after the k exemplar documents have been selected. However, *AP4ID* and *ILP4ID* outperform the baseline methods in most cases.

When changing the initial run, i.e., using a better one such as $BM25$, Figs. 9(c)-9(d) demonstrate that the diversification models present substantially different performance. Specifically, all the models except *AP4ID* tend to show better performance than the one based on the initial run with DLM. *MPT*, *DFP* and 1-*call*@k are characterized by the decreased performance when we increase the number of retrieved documents. However, *MMR* and *ILP4ID* always demonstrate a positive diversification performance that does not degrade when increasing the number of documents. *ILP4ID* outperforms the other models in most reference comparisons.

Now we investigate the possible reasons for the above findings. Even though 1-*call*@k does not require to fine-tune the trade-off parameter λ, the experimental results show that 1-*call*@k is not as competitive as the methods like MPT, DFP and $ILP4ID$. The most possible explanation is that the top-m documents are directly used to train a latent subtopic model. As Fig. 4 shows, a large portion of documents are non-relevant, thus this method greatly suffers from the noisy information. Another awkward factor that may affect 1-*call*@k is

that the topic number of the subtopic model has to be fine-tuned, otherwise the representation of each document as a subtopic vector would not be sufficiently precise.

Both *MMR* and *MPT* rely on the best first strategy, the advantage of which is that it is simple and computationally efficient (cf. Fig. 11). However, at a particular round, the document with the maximum gain via a specific heuristic criterion (i.e., Eq.7 of *MMR* and Eq.8 of *MPT*) may incur *error propagation*. For example, a long and relevant document may also include some noisy information. Once noisy information is included in the algorithm process, the diversity score of a document measured with respect to the previously selected documents would not be correct. This largely explains why both *MMR* and *MPT* under-perform *DFP* and *ILP4ID* that globally select documents.

DFP can alleviate the aforesaid problem (i.e., error propagation) based on the swapping process as it iteratively refines S by swapping a document in S with another unselected document whenever the current solution can be improved. However, *DFP* is based on the hill climbing algorithm. A potential problem is that hill climbing may not necessarily find the global maximum, but may instead converge to a local maximum. In contrast, *ILP4ID* casts the process of selecting exemplar documents as an ILP problem. Moreover, *ILP4ID* appeals to the strategy of bound-and-branch to get the exact solution. Thanks to this, *ILP4ID* is able to simultaneously consider all the candidate documents and to globally identify the optimal subset. The potential issue of error propagation is then avoided, making *ILP4ID* more robust to the noisy documents and letting it outperform the other models. Different from *ILP4ID*, *AP4ID* relies on the approximate strategy of message passing. Though *AP4ID* relatively under-performs *ILP4ID*, it achieves better efficiency (shown in Fig. 11).

To summarize, *DFP*, *AP4ID* and *ILP4ID* which belong to the cluster-based diversification paradigm are more effective than *MMR*, *MPT* and 1-*call*@k. This echoes the findings in the previous work on cluster-based IR [13, 36, 37]. Benefiting from the advantage of obtaining the optimal solution, *ILP4ID* substantially outperforms the baseline methods in most reference comparisons. Furthermore, *for implicit SRD, the factors like different initial runs, the number of input documents, the ways of computing document similarity and the optimization strategies of solving the objective formulation greatly affect the performance of a specific model.*

4.3.4. Effects of Trade-off Parameters

To clearly show the effect of the trade-off parameters λ and b for balancing relevance and diversity, we investigate how *MMR*, *MPT*, *DFP*, *AP4ID* and *ILP4ID* vary per-λ or per-b. Specifically, the top-100, 500 documents of the initial run with $BM25$ are used. All the 198 queries are tested. λ is set in the range $[0, 1]$ with a step of 0.1, and b is set in the range $[-10, 10]$ with a step of 1. In particular, for *MMR*, DFP, *AP4ID* and $ILP4ID$, $\lambda \in (0, 1)$ implies that the ranking process relies on both the relevance part and diversity part. The closer λ is to 1, the less effect the diversity component has. With $\lambda = 1$, *MMR*, *DFP*, *AP4ID* and *ILP4ID* simply rely only on the relevance of

documents, hence, they have the same performance as the initial run. With $\lambda = 0$, the performance of a model merely depends on the ability of selecting the representative documents. Regarding the effect of b on MPT (cf. Eq. 8), a positive b indicates that MPT performs a risk-aversion ranking, namely an unreliably-estimated document (with high variance) should be ranked at lower positions. The smaller b is, the less risk-averse the ranking.

In terms of ERR-IA@20, Fig. 9(e) shows how *MMR*, *DFP*, *AP4ID* and *ILP4ID* vary with changing λ, and Fig. 9(f) demonstrates how MPT varies per-b.

From Fig. 9(e), we see that tuning λ has a large effect on the performance of all models except *AP4ID*. This indicates that λ needs to be fine-tuned to achieve an optimal performance. The performance of MPT is slightly enhanced when b is close to 10 when looking at Fig. 9(f). When b is set using smaller values, the effect is not quite obvious. Moreover, a closer look at Figs. 9(e)-9(f) reveals that *ILP4ID* outperforms the baseline methods across most λ settings (and b for MPT), even though different numbers of documents of the initial run are used. This again clearly attests the effect of the deployed optimization strategy for solving the objective implicit SRD formulation.

4.3.5. Effectiveness w.r.t. Query Types

We now investigate the effectiveness of the different methods with respect to query types (cf. Section 4.3.2), either *faceted* or *ambiguous*. The comparison is conducted based on the initial retrieval with *BM25* by using the top-100, 300 and 1,000 documents, separately. In particular, Table 2 shows the results in terms of $nERR\text{-}IA@20$ and $\alpha\text{-}nDCG@20$ obtained for *MMR*, *MPT*, *DFP*, 1-*call*@k, *AP4ID* and *ILP4ID* on faceted and ambiguous queries, respectively. Table 3 shows the results in terms of P-IA@20 and Strec@20 obtained for *MMR*, *MPT*, *DFP*, 1-*call*@k, *AP4ID* and *ILP4ID* on faceted and ambiguous queries, respectively.

At first glance, Table 2 reveals that all models perform worse in terms of both $nERR\text{-}IA@20$ and $\alpha\text{-}nDCG@20$ on ambiguous queries than they do on faceted queries. This reveals that it is relatively harder to select diverse relevant documents for ambiguous queries. To further explore the possible reasons, we examined the distribution of relevant documents based on the ground-truth files. For each query type, we computed the average number of relevant documents and the average number of relevant documents that are relevant to at least 2 subtopics (termed *multi-relevant documents*). For faceted queries, these numbers are 112.42 and 47.27 whereas for ambiguous queries they are 109.35 and 19.6, respectively. These results, especially the average number of multi-relevant documents, demonstrate that it is relatively easy to retrieve some relevant documents to satisfy the subtopics of faceted queries, thus higher $nERR\text{-}IA@20$ and $\alpha\text{-}nDCG@20$ scores are observed in Table 2. This also reveals the intrinsic difference between faceted queries and ambiguous queries from the perspective of the characteristics of their relevant documents.

A joint look at Tables 2 and 3 reveals that: Except the case of using top-100 documents, *ILP4ID* outperforms other models in terms of P-IA@20 for both

Table 2: Performance of different models w.r.t. 141 faceted queries and 57 ambiguous queries in terms of $nERR\text{-}IA$@20 and $\alpha\text{-}nDCG$@20. The best result of each setting is indicted in bold. The superscript † indicates statistically significant difference when compared to the best result based on the Wilcoxon signed-rank test with $p < 0.05$.

Model	Type	$nERR\text{-}IA$@20			$\alpha\text{-}nDCG$@20		
		top-100	top-300	top-1000	top-100	top-300	top-1000
BM25	Faceted	0.2515	0.2515	0.2515	0.316	0.316†	0.316†
	Ambiguous	0.131†	0.131†	0.131†	0.1852†	0.1852†	0.1852†
MMR	Faceted	0.2622	0.269	0.2659	0.3294	0.337	0.3337
	Ambiguous	0.1421	0.137†	0.1389†	0.2009	0.2005†	0.1981†
MPT	Faceted	0.1898†	0.1578†	0.151†	0.2302†	0.1704†	0.1496†
	Ambiguous	0.1024†	0.0789†	0.0492†	0.1448†	0.1081†	0.0532†
DFP	Faceted	0.2666†	0.2264†	0.1679†	0.3321	0.3007†	0.2383†
	Ambiguous	0.1726	0.1756†	0.1079†	0.2254	0.2238	0.1601†
1-$call@k$	Faceted	0.1779†	0.1287†	0.0847†	0.2326†	0.1755†	0.1133†
	Ambiguous	0.0959†	0.0922†	0.0565†	0.1482†	0.1343†	0.0877†
$AP4ID$	Faceted	0.2544†	0.2325†	0.2028†	0.3275†	0.317†	0.2968†
	Ambiguous	**0.1961**	0.1692†	0.1658†	**0.2388**	0.2205†	0.2231†
$ILP4ID$	Faceted	**0.2832**	**0.2804**	**0.2914**	**0.3455**	**0.349**	**0.358**
	Ambiguous	0.176	**0.2116**	**0.1971**	0.2194	**0.2492**	**0.2423**

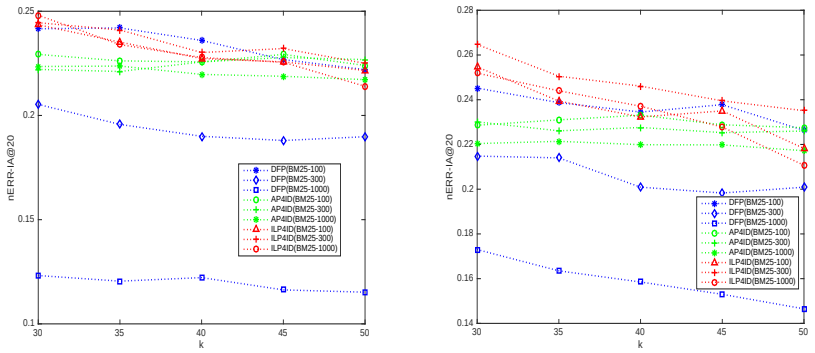
Table 3: Performance of different models w.r.t. 141 faceted queries and 57 ambiguous queries in terms of $P\text{-}IA$@20 and $Strec$@20. The best result of each setting is indicted in bold. The superscript † indicates statistically significant difference when compared to the best result based on the Wilcoxon signed-rank test with $p < 0.05$.

Model	Type	$P\text{-}IA$@20			$Strec$@20		
		top-100	top-300	top-1000	top-100	top-300	top-1000
BM25	Faceted	**0.1313**	0.1313	0.1313	0.5682†	0.5682†	0.5682†
	Ambiguous	0.0768	0.0768†	0.0768†	0.3863†	0.3863†	0.3863†
MMR	Faceted	0.1271	0.1239	0.1144	0.5961†	0.6113†	0.6083†
	Ambiguous	0.0764†	0.073	0.0632	0.4281†	0.4374†	0.4389†
MPT	Faceted	0.0615†	0.0293	0.0195	0.4455†	0.3163†	0.258†
	Ambiguous	0.042†	0.0254†	0.0076†	0.3322	0.2713†	0.1102†
DFP	Faceted	0.1177†	0.1043	0.0724	0.5926†	0.5973†	0.5511
	Ambiguous	0.0724	0.0596	0.0376†	**0.4523**	**0.4497**	0.4135
1-$call@k$	Faceted	0.0789†	0.0498	0.0271	0.474†	0.406†	0.2752†
	Ambiguous	0.0454†	0.0373†	0.0217†	0.3865†	0.343†	0.2509†
$AP4ID$	Faceted	0.1254	0.1275	0.1248	**0.6105**	**0.6286**	**0.6352**
	Ambiguous	0.0751	0.0697	0.0729†	0.4421†	0.4222†	**0.4585**
$ILP4ID$	Faceted	0.1309†	**0.1379**	**0.1392**	0.5955†	0.6156†	0.6229†
	Ambiguous	**0.0771**	**0.082**	**0.0808**	0.407†	0.4322†	0.4409

ambiguous queries and faceted queries. However, except the cases of using top-100 and top-300 documents, *AP4ID* shows a better performance than other models in terms of Strec@20, especially for faceted queries. In view of the fact that Strec@20 is not position-sensitive (discussed in Section 4.1), theses differences indicate that *ILP4ID* is more effective in ranking relevant documents at top positions.

Noteworthy, $nERR\text{-}IA@20$ and $\alpha\text{-}nDCG@20$ are the main metrics for evaluating diversification models. A closer look at Table 2 shows that *AP4ID* achieves the best performance for ambiguous queries when using the top-100 documents in terms of both $nERR\text{-}IA@20$ and $\alpha\text{-}nDCG@20$. However, with the increase of used documents, *AP4ID* shows decreased performance compared with other methods (e.g., *ILP4ID*). Benefiting from the robustness of the adopted optimization way (i.e., bound-and-branch), *ILP4ID* outperforms the other methods in most reference comparisons for both ambiguous queries and faceted queries, especially when more documents are used, being many of the improvements statistically significant.

4.4. Effects of Tuning k



(a) Initial run: *BM25*; document similarity: COS.

(b) Initial run: *BM25*; document similarity: JSD.

Figure 10: Performance variation according to the setting of k, where the x-axis indicates the value of k and the y-axis indicates the performance in terms of $nERR\text{-}IA@20$.

To clearly investigate the effect of the pre-defined cluster number k on the diversification performance, we examine how *DFP*, *AP4ID* and *ILP4ID* vary when we change the k value. Specifically, the top-100, 300, 1000 documents of the initial run with $BM25$ are used. The trade-off parameter λ is set to be 0.8, under which *DFP*, *AP4ID* and *ILP4ID* achieve high performance according to Fig. 9(e). In terms of ERR-IA@20, Fig. 10(a) and Fig. 10(b) illustrate how *DFP*, *AP4ID* and *ILP4ID* vary with different values of k.

From Fig. 10(a), we can see an overall trend that both *DFP* and *ILP4ID* perform worse when we increase the value of k, while *AP4ID* shows relatively stable performance. Meanwhile, special cases can also be observed. For example, when using the top-300 documents, *ILP4ID* under the setting of $k = 45$ shows

better performance than the result under the setting of $k = 40$. When using the top-1000 documents, *DFP* under the setting of $k = 40$ shows slightly better performance than the result under the setting of 35. Furthermore, the aforesaid overall trend is clearer in Fig. 10(b) when JSD is adopted to compute document similarity. In view of the fact that the performance is evaluated in terms of ERR-IA@20 (i.e., only the top-20 ranked results are considered), one possible reason is that larger values of k have an impact on the effectiveness of both *DFP* and *ILP4ID*. However, the impact on the effectiveness of *AP4ID* is relatively small. Therefore, when setting the cluster number k, values that are much larger than the metric's cut-off value are not recommended for both *DFP* and *ILP4ID*.

Compared with both *AP4ID* and *ILP4ID*, we see from both Fig. 10(a) and Fig. 10(b) that the performance of *DFP* is greatly impacted when more documents are used (e.g., top-1000 documents). A plausible reason is that *DFP* may suffer from the skewness problem. Specifically, given the definition of *DFP* by Eq. 9, there are k numbers (each number is in $[0, 1]$) in the relevance part Eq. 10, and m-k numbers (each number is in $[0, 1]$) in the diversity part Eq. 11. The skewness problem between the relevant part and the diversity part exists especially when $m \gg k$. As a result, the trade-off parameter λ might fail to balance the relevant part and the diversity part.

4.5. Efficiency

Common formulations of search result diversification (say, *MPT*, *DFP* and *ILP4ID*) are NP-hard (cf. [58, 17] for detailed analysis), thus approximate methods are generally adopted to find the solution. Although solving arbitrary ILPs is also an NP-hard problem, various efficient branch-and-bound algorithms have been developed. In fact, modern ILP solvers (e.g., GLPK, CPLEX and Gurobi) can find the optimal solution for moderately large optimization problems in a reasonable amount of time.

In our study, we have also evaluated the overhead of *MMR*, *MPT*, *DFP*, 1-*call*@k, *AP4ID* and *ILP4ID* by measuring the average run-time per query when generating the diversified results. All the experiments are conducted using Java (JRE 1.8.0_31) on an IMac (Intel Core i7, 4GHz, 32 GB of RAM). Based on the initial run by $BM25$, Fig. 11 plots the run-time of each model (i.e., y-axis) versus the number of input documents (i.e., x-axis).

From Fig. 11, we see that although both *MMR* and *MPT* rank documents sequentially, *MPT* requires less time when dealing with a small number of documents (say less than 400 documents). However, when the amount of documents increases, *MPT* requires more time than *MMR* and *DFP*. The main overhead is incurred by the calculation of relevance variance based on term occurrences (the time complexity is $\mathcal{O}(m^2 \cdot |W|)$, where W denotes the number of unique terms within the top-m documents). Although the formulation of *DFP* is similar to *AP4ID* and *ILP4ID*, *ILP4ID* has a higher computational cost. This is not surprising given the deployment of a branch-and-bound algorithm in order to obtain the optimal solution. Moreover, 1-*call*@k is the most computationally expensive. In fact, the time overhead is mostly caused by training the LDA subtopic model. We note that these results should be considered as indicative

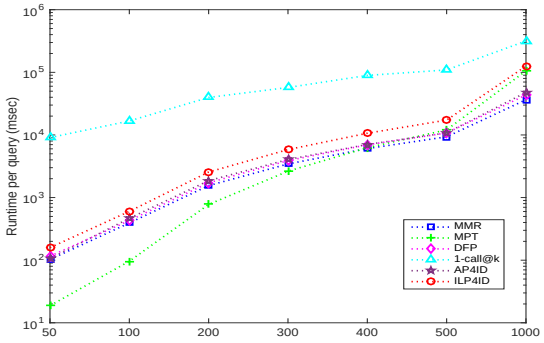


Figure 11: Average runtime per-query (msec).

only as it is possible to optimize the codes of each method, which is beyond the scope of this paper. For example, the highly-efficient algorithm [59] can be used for topic modeling which is used by 1-*call*@k. For Integer Linear Programming oriented *ILP4ID*, distributed algorithms[5] [38] or learning-to-branch methods [60, 61] are possible directions to enhance the efficiency. Moreover, today's commercial products can solve sparse problems with thousands of variables and constraints in a second, making this a realistic and promising approach in many real world problems.

5. Summary of The Key Findings

Our key findings in this paper can be summarized as follows:

- The proposed ILP formulation for implicit SRD is effective. In particular, the experimental results show that *ILP4ID* achieves substantially improved performance when compared to the state-of-the-art baseline methods.
- Given the same objective formulation (e.g., Eqs. 12 - 18) for implicit SRD, the adopted optimization strategy significantly impacts the final performance. Specifically, benefiting from the ILP formulation, *ILP4ID* which relies on the strategy of bound-and-branch is able to get the exact solution when selecting the exemplar documents. The approximate method *AP4ID* yields higher efficiency in return but inferior performance.
- By thoroughly investigating the effects of the tunable *parameters*, such as different initial runs, the number of input documents, query types, the ways of computing document similarity and the pre-defined cluster

[5] http://www.gurobi.com/products/distributed-optimization

number, we found that these parameters have significant effects on the final performance.

- The cluster hypothesis [30] can be utilized as a general paradigm for analyzing the typical unsupervised models for implicit SRD, including both the prior models [41, 10, 13, 51] and our proposed methods (i.e., *ILP4ID* and *AP4ID*).

Overall, it is reasonable to say that the cluster-based paradigm for implicit SRD is effective for generating diversified results, whereas it also provides flexibility when designing a particular model. In particular, for a specific objective formulation, both the optimization strategy and the tunable parameters should be well designed and fine-tuned. Otherwise, the diversification models would be significantly impacted.

6. Conclusions And Future Work

In this paper, we propose a novel ILP formulation to solve the problem of implicit SRD. The key idea is to formulate implicit SRD as a process of selecting and ranking k exemplar documents from the top-m documents of an initial retrieval. In particular, two different approaches *ILP4ID* and *AP4ID* are proposed to solve the objective ILP formulation.

To justify the effectiveness and efficiency of the proposed approaches, a series of experiments are conducted based on four benchmark collections. The experimental results demonstrate that: Given the ILP formulation of implicit SRD, *ILP4ID* is able to obtain the optimal solution, and leads to substantially improved performance when compared to the state-of-the-art baseline methods. As a complementary way of solving the proposed ILP formulation, *AP4ID* that works via message passing is proposed as an approximate method. Although AP4ID outperforms ILP4ID and other baseline methods only for some cases (cf. Section 4.3.3), it sheds light on devising more efficient algorithms for solving ILP formulation of implicit SRD and is less computationally expensive than *ILP4ID*. Since problems analogous to implicit SRD arise in a variety of applications, e.g., recommender systems [62, 63], we believe that our approaches provide a new perspective for addressing problems of this kind.

The proposed approaches can be further improved in several aspects. For example, the optimal cluster number k essentially differs from query to query [35]. Dynamically determining the value of k on both *ILP4ID* and *AP4ID* is then worthy to be investigated in the future. Moreover, in view of the success achieved by the state-of-the-art deep learning algorithms for document representation [64, 65], we also plan to study how to incorporate the algorithms of this kind in order to explore the internal correlations among documents within the same cluster and the external correlations among exemplar documents.

7. Acknowledgments

Sincere thanks to reviewers for helping us improve this study. This research has been supported by JSPS KAKENHI Grant Number JP17K12784.

References

[1] S. Kim, J. Lee, Subtopic mining using simple patterns and hierarchical structure of subtopic candidates from web documents, Information Processing & Management 51 (6) (2015) 773–785.

[2] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to rank using gradient descent, in: Proceedings of the 22nd ICML, 2005, pp. 89–96.

[3] Z. Cao, T. Qin, T. Liu, M. Tsai, H. Li, Learning to rank: from pairwise approach to listwise approach, in: Proceedings of the 24th ICML, 2007, pp. 129–136.

[4] L. Xia, J. Xu, Y. Lan, J. Guo, X. Cheng, Learning maximal marginal relevance model via directly optimizing diversity evaluation measures, in: Proceedings of the 38th SIGIR, 2015, pp. 113–122.

[5] F. Radlinski, R. Kleinberg, T. Joachims, Learning diverse rankings with multi-armed bandits, in: Proceedings of the 25th ICML, 2008, pp. 784–791.

[6] L. Xia, J. Xu, Y. Lan, J. Guo, X. Cheng, Modeling document novelty with neural tensor network for search result diversification, in: Proceedings of the 39th SIGIR, 2016, pp. 395–404.

[7] Y. Yue, T. Joachims, Predicting diverse subsets using structural SVMs, in: Proceedings of the 25th ICML, 2008, pp. 1224–1231.

[8] L. Xia, J. Xu, Y. Lan, J. Guo, W. Zeng, X. Cheng, Adapting markov decision process for search result diversification, in: Proceedings of the 40th SIGIR, 2017, pp. 535–544.

[9] Z. Jiang, J. Wen, Z. Dou, W. X. Zhao, J. Nie, M. Yue, Learning to diversify search results via subtopic attention, in: Proceedings of the 40th SIGIR, 2017, pp. 545–554.

[10] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st SIGIR, 1998, pp. 335–336.

[11] R. L. Santos, C. Macdonald, I. Ounis, Exploiting query reformulations for web search result diversification, in: Proceedings of the 19th WWW, 2010, pp. 881–890.

[12] V. Dang, W. B. Croft, Diversity by proportionality: an election-based approach to search result diversification, in: Proceedings of the 35th SIGIR, 2012, pp. 65–74.

[13] G. Zuccon, L. Azzopardi, D. Zhang, J. Wang, Top-k retrieval using facility location analysis, in: Proceedings of the 34th ECIR, 2012, pp. 305–316.

[14] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to information retrieval, ACM Transactions on Information Systems 22 (2) (2004) 179–214.

[15] H. Yu, J. Adam, R. Blanco, H. Joho, J. Jose, L. Chen, F. Yuan, A concise integer linear programming formulation for implicit search result diversification, in: Proceedings of the 10th WSDM, 2017, pp. 191–200.

[16] O. Kurland, The cluster hypothesis in information retrieval, in: SIGIR2013 tutorial, 2013.

[17] R. L. T. Santos, C. Macdonald, I. Ounis, Search result diversification, Foundations and Trends in Information Retrieval 9 (1) (2015) 1–90.

[18] B. J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.

[19] M. J. Garnett, E. J. Edelman, S. J. Heidorn, et al., Systematic identification of genomic markers of drug sensitivity in cancer cells, Nature (2012) 570–575doi:10.1038/nature11005.

[20] Y. Wang, L. Chen, K-MEAP: multiple exemplars affinity propagation with specified K clusters, IEEE Transactions on Neural Networks and Learning Systems PP (99) (2015) 1–13.

[21] J. Xiao, J. Wang, P. Tan, L. Quan, Joint affinity propagation for multiple view segmentation, in: Proceedings of the 11th ICCV, 2007, pp. 1–7.

[22] I. E. Givoni, B. J. Frey, A binary variable model for affinity propagation, Neural Computation 21 (6) (2009) 1589–1600.

[23] H. Yu, F. Ren, Search result diversification via filling up multiple knapsacks, in: Proceedings of the 23rd CIKM, 2014, pp. 609–618.

[24] V. Dang, W. B. Croft, Term level search result diversification, in: Proceedings of the 36th SIGIR, 2013, pp. 603–612.

[25] S. Hu, Z. Dou, X. Wang, T. Sakai, J. Wen, Search result diversification based on hierarchical intents, in: Proceedings of the 24th CIKM, 2015, pp. 63–72.

[26] S. Liang, Z. Ren, M. de Rijke, Fusion helps diversification, in: Proceedings of the 37th SIGIR, 2014, pp. 303–312.

[27] A. M. Ozdemiray, I. S. Altingovde, Explicit search result diversification using score and rank aggregation methods, Journal of the American Society for Information Science and Technology 66 (6) (2015) 1212–1228.

[28] S. Liang, E. Yilmaz, H. Shen, M. D. Rijke, W. B. Croft, Search result diversification in short text streams, ACM Transactions on Information Systems 36 (1) (2017) 8:1–8:35.

[29] Y. Zhu, Y. Lan, J. Guo, X. Cheng, S. Niu, Learning for search result diversification, in: Proceedings of the 37th SIGIR, 2014, pp. 293–302.

[30] C. J. V. Rijsbergen, Information Retrieval, 2nd Edition, 1979.

[31] X. Liu, W. B. Croft, Cluster-based retrieval using language models, in: Proceedings of the 27th SIGIR, 2004, pp. 186–193.

[32] O. Kurland, L. Lee, Clusters, language models, and ad hoc information retrieval, ACM Transactions on Information Systems 27 (3) (2009) 13:1–13:39.

[33] X. Liu, W. B. Croft, Evaluating text representations for retrieval of the best group of documents, in: Proceedings of the 30th ECIR, 2008, pp. 454–462.

[34] O. Kurland, Re-ranking search results using language models of query-specific clusters, Journal of Information Retrieval 12 (4) (2009) 437–460.

[35] O. Levi, F. Raiber, O. Kurland, I. Guy, Selective cluster-based document retrieval, in: Proceedings of the 25th CIKM, 2016, pp. 1473–1482.

[36] J. He, E. Meij, M. de Rijke, Result diversification based on Query-specific cluster ranking, JASIST 62 (3) (2011) 550–571.

[37] F. Raiber, O. Kurland, Ranking document clusters using markov random fields, in: Proceedings of the 36th SIGIR, 2013, pp. 333–342.

[38] K. D. Naini, I. S. Altingovde, W. Siberski, Scalable and efficient web search result diversification, ACM Transactions on the Web 10 (3) (2016) 15:1–15:30.

[39] S. Guo, S. Sanner, Probabilistic latent maximal marginal relevance, in: Proceedings of the 33rd SIGIR, 2010, pp. 833–834.

[40] S. Sanner, S. Guo, T. Graepel, S. Kharazmi, S. Karimi, Diverse retrieval via greedy optimization of expected 1-call@k in a latent subtopic relevance model, in: Proceedings of the 20th CIKM, 2011, pp. 1977–1980.

[41] J. Wang, J. Zhu, Portfolio theory of information retrieval, in: Proceedings of the 32nd SIGIR, 2009, pp. 115–122.

[42] S. Gollapudi, A. Sharma, An axiomatic approach for result diversification, in: Proceedings of the 18th WWW, 2009, pp. 381–390.

[43] D. Roth, W. Yih, Integer linear programming inference for conditional random fields, in: Proceedings of the 22nd ICML, 2005, pp. 736–743.

[44] A. F. T. Martins, N. A. Smith, E. P. Xing, Concise integer linear programming formulations for dependency parsing, in: Proceedings of the 47th ACL, 2009, pp. 342–350.

[45] K. Woodsend, M. Lapata, Multiple aspect summarization using integer linear programming, in: EMNLP-CoNLL2012, 2012, pp. 233–243.

[46] R. Agrawal, S. Gollapudi, A. Halverson, S. Ieong, Diversifying search results, in: Proceedings of the 2nd WSDM, 2009, pp. 5–14.

[47] R. E. Gomory, Solving linear programming problems in integers, in: Proceedings of Symposia in Applied Mathematics, Vol. 10, 1960, pp. 211–215.

[48] A. H. Land, A. G. Doig, An automatic method of solving discrete programming problems, Econometrica 28 (3) (1960) 497–520.

[49] N. Lazic, Message passing algorithms for facility location problems, Ph.D. thesis, University of Toronto (2011).

[50] D. Dueck, Affinity propagation: clustering data by passing messages, Ph.D. thesis, University of Toronto (2009).

[51] G. Zuccon, L. Azzopardi, Using the quantum probability ranking principle to rank interdependent documents, in: Proceedings of the 32nd ECIR, 2010, pp. 357–369.

[52] C. L. Clarke, N. Craswell, I. Soboroff, Overview of the TREC 2009 web track, in: TREC, 2009.

[53] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, I. MacKinnon, Novelty and diversity in information retrieval evaluation, in: Proceedings of the 31st SIGIR, 2008, pp. 659–666.

[54] C. X. Zhai, W. W. Cohen, J. Lafferty, Beyond independent relevance: methods and evaluation metrics for subtopic retrieval, in: Proceedings of the 26th SIGIR, 2003, pp. 10–17.

[55] T. Sakai, R. Song, Evaluating diversified search results using per-intent graded relevance, in: Proceedings of the 34th SIGIR, 2011, pp. 1043–1052.

[56] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at trec-3, in: Proceedings of TREC, 1994.

[57] J. H. Lee, Analyses of multiple evidence combination, in: Proceedings of the 20th SIGIR, 1997, pp. 267–276.

[58] T. Deng, W. Fan, On the complexity of query result diversification, ACM Transactions on Database Systems 39 (15) (2014) 15:1–15:46.

[59] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T. Liu, W. Ma, LightLDA: big topic models on modest computer clusters, in: Proceedings of the 24th WWW, 2015, pp. 1351–1361.

[60] E. B. Khalil, P. L. Bodic, L. Song, G. Nemhauser, B. Dilkina, Learning to branch in mixed integer programming, in: AAAI Conference on Artificial Intelligence, 2016, pp. 724–731.

[61] H. He, H. Daumé, III, J. Eisner, Learning to search in branch-and-bound algorithms, in: Proceedings of NIPS 27, 2014, pp. 3293–3301.

[62] M. d. Gemmis, P. Lops, G. Semeraro, C. Musto, An investigation on the serendipity problem in recommender systems, Information Processing & Management 51 (5) (2015) 695–717.

[63] W. Chen, F. Cai, H. Chen, M. de Rijke, Personalized query suggestion diversification, in: Proceedings of the 40th SIGIR, 2017, pp. 817–820.

[64] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature (521) (2015) 436–444. doi:10.1038/nature14539.

[65] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31st ICML, 2014, pp. 1188–1196.

$$\sigma_{ij} = (1-\lambda) \cdot k \cdot s(d_i, d_j) \tag{A.5}$$

$$\begin{aligned}\beta_{ij}(x_{ij}) &= \max_{t_1} ... \max_{t_{j-1}} \max_{t_{j+1}} ... \max_{t_n} \{\mathcal{H}_i(x_{i1}, ..., x_{im}) + \sum_{j':j' \neq j} \tau_{ij'}(t_{j'})\} \\ &= \begin{cases} x_{ij} = 1 & \sum\limits_{j':j' \neq j} \tau_{ij'}(0) \\ x_{ij} = 0 & \max\limits_{j':j' \neq j} \{\tau_{ij'}(1) + \sum\limits_{j'':j'' \notin \{j,j'\}} \tau_{ij''}(0)\} \end{cases}\end{aligned} \tag{A.6}$$

where $\{t_{j'} \in \{0,1\}\}$ denotes the possible states of all neighboring variable nodes $x_{i1}, ..., x_{i(j-1)}, x_{i(j+1)}, ..., x_{im}$.

$$\beta_{ij} = \beta_{ij}(1) - \beta_{ij}(0) = -\max_{j':j' \neq j} \tau_{ij'} = \begin{cases} i = j & -\max\limits_{j':j' \neq i} \{\tau_{ij'}\} \\ i \neq j & -\max\{\tau_{ii}, \max\limits_{j':j' \notin \{i,j\}} \tau_{ij'}\} \end{cases} \tag{A.7}$$

$$\alpha_{ij}(x_{ij}) = \max_{t_1} ... \max_{t_{i-1}} \max_{t_{i+1}} ... \max_{t_n} \{\mathcal{F}_j(x_{1j}, ..., x_{mj}) + \sum_{i':i' \neq i} \rho_{i'j}\{t_{i'}\}\} \tag{A.8}$$

$$\alpha_{ij}(1) = \begin{cases} i = j & \sum\limits_{i':i' \neq i} \max\{\rho_{i'j}(0), \rho_{i'j}(1)\} \\ i \neq j & \rho_{jj}(1) + \sum\limits_{i':i' \notin \{i,j\}} \max\{\rho_{i'j}(0), \rho_{i'j}(1)\} \end{cases} \tag{A.9}$$

$$\alpha_{ij}(0) = \begin{cases} i = j & \sum\limits_{i':i' \neq i} \rho_{i'j}(0) \\ i \neq j & \max\{\sum\limits_{i':i' \neq i} \rho_{i'j}(0), \rho_{jj}(1) + \sum\limits_{i':i' \notin \{i,j\}} \max\{\rho_{i'j}(0), \rho_{i'j}(1)\}\} \end{cases} \tag{A.10}$$

$$\alpha_{ij} = \alpha_{ij}(1) - \alpha_{ij}(0) = \begin{cases} i = j & \sum\limits_{i':i' \neq i} \max\{0, \rho_{i'j}\} \\ i \neq j & \min\{0, \rho_{jj} + \sum\limits_{i':i' \notin \{i,j\}} \max\{0, \rho_{i'j}\}\} \end{cases} \tag{A.11}$$

$$\begin{aligned}\eta_i(x_{ii}) &= \max_{z_i, x_{ii}, z_{i-1}} \{\mathcal{G}_i(z_i, x_{ii}, z_{i-1}) + b_i(z_i) + a_{i-1}(z_{i-1})\} \\ &= \begin{cases} \max_{z_i}\{b_i(z_i) + a_{i-1}(z_i - 1)\} & if\, x_{ii} = 1\, and\, z_{i-1} = z_i - 1 \\ \max_{z_i}\{b_i(z_i) + a_{i-1}(z_i)\} & if\, x_{ii} = 0\, and\, z_{i-1} = z_i \end{cases}\end{aligned} \tag{A.12}$$

$$\eta_i = \eta_i(1) - \eta_i(0) = \max_{z_i}\{b_i(z_i) + a_{i-1}(z_i - 1)\} - \max_{z_i}\{b_i(z_i) + a_{i-1}(z_i)\} \quad \text{(A.13)}$$

For the message a, we have

$$\begin{aligned} a_1(z_1) &= \max_{z_0, y_1, z_1}\{\mathcal{G}_1(z_1, x_{11}, z_0) + c_{i-1}(z_0) + \varepsilon_1\} = \max_{z_1, s.t.\ z_1 = x_{11} + z_0}\{a_0(z_0) + \varepsilon_1\} \\ &= \begin{cases} a_0(z_1 - 1) + \varepsilon_1(1) & if\ x_{11} = 1\ and\ z_0 = z_1 - 1 \\ a_0(z_1) + \varepsilon_1(0) & if\ x_{11} = 0\ and\ z_0 = z_1 \end{cases} \end{aligned} \quad \text{(A.14)}$$

then taking the constant part as $\varepsilon_1(0)$, we get

$$a_1(z_1) = \max\{a_0(z_1 - 1) + \varepsilon_1,\ a_0(z_1)\} \quad \text{(A.15)}$$

$$a_j(z_j) = \max\{a_{j-1}(z_j - 1) + \alpha_{jj} + \beta_{jj} + r_j,\ a_{j-1}(z_j)\} \quad \text{(A.16)}$$

$$b_{j-1}(z_{j-1}) = \max\{b_j(z_{j-1}), b_j(z_{j-1} + 1) + \alpha_{jj} + \beta_{jj} + r_j\} \quad \text{(A.17)}$$

To summarize, the message update equations are:

$$\tau_{ij} = \begin{cases} i = j & \eta_i + \alpha_{ii} + \gamma_i \\ i \neq j & \sigma_{ij} + \alpha_{ij} \end{cases} \quad \text{(A.18)}$$

$$\rho_{ij} = \begin{cases} i = j & \eta_i + \beta_{ii} + \gamma_i \\ i \neq j & \sigma_{ij} + \beta_{ij} \end{cases} \quad \text{(A.19)}$$

$$\beta_{ij} = \begin{cases} i = j & -\max_{j' : j' \neq i}\{\tau_{ij'}\} \\ i \neq j & -\max\{\tau_{ii}, \max_{j' : j' \notin \{i,j\}} \tau_{ij'}\} \end{cases} \quad \text{(A.20)}$$

$$\alpha_{ij} = \begin{cases} i = j & \sum_{i' : i' \neq i} \max\{0, \rho_{i'j}\} \\ i \neq j & \min\{0, \rho_{jj} + \sum_{i' : i' \notin \{i,j\}} \max\{0, \rho_{i'j}\}\} \end{cases} \quad \text{(A.21)}$$

$$\eta_i = \max_{z_i}\{b_i(z_i) + a_{i-1}(z_i - 1)\} - \max_{z_i}\{b_i(z_i) + a_{i-1}(z_i)\} \quad \text{(A.22)}$$

$$a_j(z_j) = \max\{a_{j-1}(z_j - 1) + \alpha_{jj} + \beta_{jj} + r_j,\ a_{j-1}(z_j)\} \quad \text{(A.23)}$$

$$b_{j-1}(z_{j-1}) = \max\{b_j(z_{j-1}), b_j(z_{j-1} + 1) + \alpha_{jj} + \beta_{jj} + r_j\} \quad \text{(A.24)}$$

Finally, the update equation in Section 3.2 can be derived by substituting β_{ij} and τ_{ij} correspondingly.