Université de Montréal

Diversified Query Expansion

par
Arbi Bouchoucha

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures et postdoctorales
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en Informatique

June, 2015

Résumé

La diversification des résultats de recherche (DRR) vise à sélectionner divers documents à partir des résultats de recherche afin de couvrir autant d'intentions que possible. Dans les approches existantes, on suppose que les résultats initiaux sont suffisamment diversifiés et couvrent bien les aspects de la requête. Or, on observe souvent que les résultats initiaux n'arrivent pas à couvrir certains aspects. Dans cette thèse, nous proposons une nouvelle approche de DRR qui consiste à diversifier l'expansion de requête (DER) afin d'avoir une meilleure couverture des aspects. Les termes d'expansion sont sélectionnés à partir d'une ou de plusieurs ressource(s) suivant le principe de pertinence marginale maximale [22]. Dans notre première contribution, nous proposons une méthode pour **DER au niveau des termes** où la similarité entre les termes est mesurée superficiellement à l'aide des ressources.

Quand plusieurs ressources sont utilisées pour DER, elles ont été uniformément combinées dans la littérature, ce qui permet d'ignorer la contribution individuelle de chaque ressource par rapport à la requête. Dans la seconde contribution de cette thèse, nous proposons une nouvelle méthode de **pondération de ressources selon la requête**. Notre méthode utilise un ensemble de caractéristiques qui sont intégrées à un modèle de régression linéaire, et génère à partir de chaque ressource un nombre de termes d'expansion proportionnellement au poids de cette ressource.

Les méthodes proposées pour DER se concentrent sur l'élimination de la redondance entre les termes d'expansion sans se soucier si les termes sélectionnés couvrent effectivement les différents aspects de la requête. Pour pallier à cet inconvénient, nous introduisons dans la troisième contribution de cette thèse une nouvelle méthode pour **DER au niveau des aspects**. Notre méthode est entraînée de façon supervisée selon le principe que les termes reliés doivent correspondre au même aspect. Cette méthode permet de sélectionner des termes d'expansion à un niveau sémantique latent afin de couvrir autant que possible différents aspects de la requête. De plus, cette méthode autorise l'intégration de plusieurs ressources afin de suggérer des termes d'expansion, et supporte l'intégration de plusieurs contraintes telles que la contrainte de dispersion.

Nous évaluons nos méthodes à l'aide des données de ClueWeb09B et de trois collections de requêtes de TREC Web track et montrons l'utilité de nos approches par rapport aux méthodes existantes.

Mots clés: Diversification des résultats de recherche, expansion de requête, intégration de ressources, pondération de ressources, incorporation latente d'aspects.

Abstract

Search Result Diversification (SRD) aims to select diverse documents from the search results in order to cover as many search intents as possible. For the existing approaches, a prerequisite is that the initial retrieval results contain diverse documents and ensure a good coverage of the query aspects. In this thesis, we investigate a new approach to SRD by diversifying the query, namely diversified query expansion (DQE). Expansion terms are selected either from a single resource or from multiple resources following the Maximal Marginal Relevance principle [22]. In the first contribution, we propose a new **term-level DQE** method in which word similarity is determined at the surface (term) level based on the resources.

When different resources are used for the purpose of DQE, they are combined in a uniform way, thus totally ignoring the contribution differences among resources. In practice the usefulness of a resource greatly changes depending on the query. In the second contribution, we propose a new method of query level **resource weighting for DQE**. Our method is based on a set of features which are integrated into a linear regression model and generates for a resource a number of expansion candidates that is proportional to the weight of that resource.

Existing DQE methods focus on removing the redundancy among selected expansion terms and no attention has been paid on how well the selected expansion terms can indeed cover the query aspects. Consequently, it is not clear how we can cope with the semantic relations between terms. To overcome this drawback, our third contribution in this thesis aims to introduce a novel method for **aspect-level DQE** which relies on an explicit modeling of query aspects based on embedding. Our method (called *latent semantic aspect embedding*) is trained in a supervised manner according to the principle that related terms should correspond to the same aspects. This method allows us to select expansion terms at a latent semantic level in order to cover as much as possible the aspects of a given query. In addition, this method also incorporates several different external resources to suggest potential expansion terms, and supports several constraints, such as the sparsity constraint.

We evaluate our methods using ClueWeb09B dataset and three query sets from TREC Web tracks, and show the usefulness of our proposed approaches compared to the state-of-the-art approaches.

Keywords: Search Result Diversification; Query Expansion; Multiple Resource Integration; Resource Weighting; Latent Aspect Embedding.

Contents

List of Tables

List of Figures

List of Abbreviations

1

BOW	Bag Of Words
DSSM	Deep Structured Semantic Models
DQE	Diversified Query Expansion
eR	embedding Retrieval
eRS	embedding Retrieval with Sparsity
ERR	Expected Reciprocal Rank
ERR-IA	Expected Reciprocal Rank - Intent Aware
xMVA	explicit Mean Variance Analysis
xQuAD	explicit Query Aspect Diversification
ESA	Explicit Semantic Analysis
IR	Information Retrieval
KL	Kullback-Leibler
KNN	K-Nearest Neighbor
LDA	Latent Dirichlet Allocation
MMR	Maximal Marginal Relevance
MMRE	Maximal Marginal Relevance-based Expansion
MRV	Maximal Result Variety
MAP	Mean Average Precision
MVA	Mean Variance Analysis
MPT	Modern Portfolio Theory

List of Abbreviations

2

MSS	Multi-Search Subtopic
nQL-RW	non Query Level Resource Weighting
NTCIR	National Testbeds and Community for Information access Research project
nDCG	normalized Discriminative Cumulative Gain
NRBP	Novelty and Rank-biased Precision
ODP	Open Directory Project
Prec-IA	Precision - Intent Aware
PCA	Principal Component Analysis
PRF	Pseudo-Relevance Feedback
QE	Query Expansion
QL-RW	Query Level Resource Weighting
SRD	Search Result Diversification
SA	Spreading Activation
S-recall	Subtopic Recall
SVN	Support Vector Machines
SVR	Support Vector Regression
TREC	Text REtrieval Conference
U-RW	Uniform Resource Weighting
WT	Web Track

Je dédie cette thèse à:

Mon père ***Abderraouf***,
Ma mère ***Latifa***,
Pour tous les sacrifices qu'ils ont consentis à faire pour que je puisse être là aujourd'hui. Qu'ils puissent trouver dans ce modeste travail la récompense de tous leurs efforts.

À mon épouse ***Abir***,

À mon cher petit poussin ***Yacine***,

À ma soeur ***Inès***,

À la mémoire de ma grand-mère ***Chedlia***,

À tous ceux qui me sont chers, et qui m'ont aidé de proche ou de loin pour la réalisation de ce travail.

Remerciements

Je débute la liste de remerciements par Professeur Jian-Yun Nie qui m'a accueilli dans son équipe et m'a enseigné les bases de mes connaissances dans le domaine de recherche d'information. Je le remercie particulièrement pour la confiance qu'il m'a témoignée tout au long de cette thèse, pour son suivi, ses conseils enrichissants, et aussi son sens de perfection dans tous les détails. Il m'a permis de progresser, de prendre conscience de mes responsabilités pour parvenir à réaliser ce travail. Les mots sont faibles pour lui exprimer ma reconnaissance.

Je veux également remercier les membres du jury qui ont bien accepté d'évaluer cette thèse.

Je tiens à remercier le ministère de l'enseignement supérieur Tunisien et l'Université de Montréal pour avoir co-financé ce travail de recherche à travers plusieurs bourses d'excellence.

Je remercie ensuite Xiaohua Liu et Jing He, qui n'ont jamais épargné l'effort de m'aider, pour leurs grandes qualités humaines. Merci pour l'énorme soutien aussi bien technique que moral, leurs encouragements y sont pour beaucoup dans l'aboutissement de ce travail.

J'exprime également ma gratitude à tous mes enseignants qui ont contribué chacun dans son domaine à ma formation universitaire, sans laquelle je ne serai jamais arrivé à réaliser ce travail.

Finalement, je dis MERCI à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail modeste.

Chapter 1

Introduction

1.1 Research Context

Nowadays, Information Retrieval (IR) on the Web becomes the main mean for internet users to fulfil their information needs. Using an IR system (particularly a search engine such as Google [1], Bing [2], Baidu [3], etc.), a user can easily specify her information need through a few keywords (query) and obtain results quickly. However, most of the user queries are both short and ambiguous. Existing studies showed that the average length of user queries is around 2.3 terms per query [76]. Short queries generally mean a lot of ambiguity as to what information needs they express. This is typically the case of the ambiguous query "Java" for example, which could be interpreted as *programming language*, *island*, *coffee*, etc. Even for the case of the non-ambiguous queries (the queries in which terms have a unique interpretation), the query is still often underspecified [33] and there may be several aspects that are related to this type of queries. Consider for example, the query "C++ programming", in which the terms are not ambiguous. This query can at least be related to *books*, *discussion forums*, *online courses (tutorials)*, *software*, etc. In practice, these interpretations and aspects of queries may be used to simulate different possible user intents. Understanding user intents consists of understanding what people are seeking for with their query, what the underlying information need is, and what the final goal of search is. Note that, in this thesis, we distinguish between *interpretations* and *aspects*: despite both of them being extracted automatically, the terminology *interpretations* is generally used for ambiguous queries, while the terminology *aspects* is generally adopted for the case of non-ambiguous queries. Since we don't classify the queries into ambiguous or non-ambiguous in this thesis, and for simplicity, we will use the terminology *aspects* to refer to both interpretations and aspects.

Generally, when the user information need is clearly specified beforehand, and she has a good knowledge of the target documents, then she can efficiently formulate her query, which allows the search engine to return the most relevant and diversified documents in the top of list. This is generally

1. www.google.ca
2. www.bing.com
3. www.baidu.com

the case when users look for information on some popular topics, in which the user information need behind the original query is clearly defined. In this case, a few keywords are sufficient to fully describe the user intent. Otherwise, if the user does not have any particular document in mind or does not have any knowledge about the domain of the query, then short queries cannot clearly express the user information need. This is generally the case of the so-called informational queries [18] where the user seeks to explore particular information [87]. For instance, users searching for the ambiguous query "apple" may be interested to several intents and some users may even look for some specific information about apple, such as *the benefits of apple fruit for diabetics* or *job openings in apple store*.

The crucial problem is that the same query can be used to express very different search intents. To cope with the problem, one possible solution is to precisely determine the user intents (such as [4, 66, 68, 78, 79, 101]). However, determining the exact user intent is a difficult task due to the huge number and variety of users on the Web and their information needs (different users have very different preferences for the same query) which makes it very difficult to understand their intents. Besides, it is well known that the user generally stops at the first page from the list of result pages returned by the search engine. It is therefore important that the user finds at least one document that is relevant to her from the first result page. Otherwise, if none of the returned documents corresponds to her information need, the user satisfaction will be low and this may lead to user abandonment of search [28, 44]. Another possible solution is to diversify the search results. Search result diversification (SRD) aims to rerank the initial retrieval results in order to include documents relevant to different possible intents in the first result page, hoping that the user will find at least one document that is relevant to her information need. SRD aims to diversify the search results without the need to explicitly know the specific search intent of the user behind her query. In this thesis, we adopt the second solution which consists of diversifying search results of a given query, instead of determining the exact user intent. Several approaches have been proposed in the literature, defining different strategies to produce a diversified result list [2, 17, 42, 43, 55, 56, 104].

1.2 Specific Problem

Most approaches to SRD usually operate in two stages: a search is performed with the initial query to identify candidate documents, and these results are re-ranked by incorporating a diversity

criterion [2, 22, 105, 132]. Existing diversification methods are either *implicit* or *explicit*. Implicit SRD approaches (*e.g.*, [22, 28, 56, 125] promote dissimilar documents through the relations among documents, and aim to maximize the novelty and reduce the redundancy of the selected documents, hoping that the produced ranked list conveys both relevant and diverse information about a query. In other words, Implicit SRD approaches iteratively select the document that may bring the maximum novel amount of information compared to the documents that have been selected earlier. Consequently, a document that is redundant or is similar to at least one of the already selected documents will be penalized. Explicit SRD approaches (*e.g.*, [21, 43, 61, 104, 130]), however, explicitly extract aspects of the query, and then try to cover as much as possible these aspects based on the relation between the documents and the query subtopics. In this thesis, we define query *subtopic* or a *sub-query*, as the intents which are manually defined, such as the query subtopics that are manually identified by the TREC assessors. Since the query subtopics are not available in practice, one can automatically extract the query aspects to simulate these subtopics.

Existing studies on SRD rely heavily on the set of returned documents corresponding to the original query. They implicitly assume that the returned documents are relevant and can cover (almost) all the query intents, but these documents are not well ordered. This idea may work well if the returned search results corresponding to the original query are of good quality, *i.e.*, contain relevant documents that cover several query aspects. However, this is not always the case: initial search results are often unable to cover various search intents due to the problem of query ambiguity and dominating subtopics. For example, the results with the query "Java" will be overwhelmingly about the Java programming language (which is the dominating aspect of "java"), and the other intents (*coffee* and *island*) will likely be absent in the top search results.

The common principle used in the existing SRD approaches is to select as diverse results as possible from a given set of retrieved documents. The final ranking list is much dependent on the initial retrieval results, which may not have a good coverage of the different aspects of the query. To overcome this drawback, some existing studies on SRD attempted to expand the original query before diversifying the results. However, a traditional query expansion method, typically using pseudo-feedback documents, does not ensure that the returned results are more diverse. Indeed, when a query is expanded in a traditional way, the retrieval results with the expanded query are likely to have an even larger coverage of the dominant aspect of the query, to the detriment of less popular aspects.

To solve this problem, Vargas *et al.* [119] recently proposed a new method of pseudo-relevance feedback (PRF) for SRD. In this approach, the search results are first distinguished into different aspects and PRF is applied for each aspect separately. Compared to a unique query expansion, the aspect-dependent expansion can keep a better balance among the aspects in the final retrieval results. However, this approach is still much dependent on the retrieval results with the initial query. In the case where some aspects are not well covered in the initial retrieval results, such an aspect-dependent PRF method will be unable to cover them well. In the case of a difficult query in particular, the retrieval results are mostly irrelevant [3]. For instance, query like "appraisals" is difficult. The retrieved results of this query using a traditional model on ClueWeb09B dataset are not relevant, and based on document feedback, it is difficult to extract relevant terms for expansion. Thus, PRF will bring noise rather than useful terms into the query.

1.3 Problem Statement

In this dissertation, we distinguish between three main problems.

Part 1: Term-level DQE

Problem 1.1: As discussed in the previous section, the effectiveness of existing SRD approaches are related to the quality of initial retrieval results (which should have a good coverage of the query aspects). However, this is not always the case due to the problem of query ambiguity and dominating subtopics. Despite some attempts to expand the original query [119] using PRF, the problem is not solved since selected expansion terms are still much dependent on the retrieval results with the original query. In the case where some aspects are not well covered in the initial retrieval results, this method will be unable to cover them well.

Problem 1.2: A critical aspect of query expansion based on external resources is the coverage of the latter. An external resource should cover as much as possible all the aspects and meanings of the query terms. However, a single resource can hardly cover all the aspects for every query. For example, Wikipedia has been used as an external resource for query expansion, but Wikipedia articles do not cover all the aspects. Query logs are often used to suggest expansion terms, but there may be less frequent query aspects poorly covered by query logs. It is necessary to combine multiple resources.

Part 2: Resource weighting for DQE

Problem 2.1: When integrating multiple resources, they are combined in a uniform way. However, the usefulness of a resource can greatly change depending on the queries: one resource could be very useful for some specific query, but not useful for another one. Consequently, candidate expansion terms suggested by a resource which is useful for a given query should be preferred since they are more likely to be related to one or several aspects of the query. Similarly, candidate expansion terms that are derived from a less important resource with respect to some query should not be promoted since they are less likely to be related to the query aspects. Such expansion terms may bring much noise than useful information to the query aspects.

Part 3: Aspect-level DQE

Problem 3.1: Term-level DQE methods select candidate expansion terms at the surface (word) level without considering the semantic relations between the selected terms regarding to the query. In other words, despite expansion terms being selected from different resources (which may be likely to cover different aspects of the query), it still remains unclear how these expansion terms indeed cover the aspects, with the absence of any clear and explicit representation of the query aspects. In particular, when expanding a query using a set of diversified expansion terms selected from one or several resources, we assume that an aspect of the query can simply be represented by one or several expansion terms. A potential problem is that an expansion term can appear different from the previous expansion terms, yet it describes exactly the same semantic intent. For example, once the term *library* has been selected as an expansion term for the query "Java", the term *class* could be viewed as an independent one, thus added as an additional expansion term. Yet both expansion terms are related to the same query intent - *Java programming language*. This gives rise to the problem of selecting multiple expansion terms relating to the same query aspect.

1.4 Contributions

To tackle the previously identified problems, we propose the following contributions, organized in three major parts:

Part 1: Term-level DQE

Contribution 1.1 - using external resources: An alternative approach to query expansion is to use external resources rather than the retrieval results. For example, one may use a general thesaurus such as WordNet [4] or ConceptNet [5], to expand queries. Such an approach has been explored in general IR. We first propose to dig deeper in this direction. We leverage ConceptNet for SRD, which is one of the largest common-sense knowledge base that covers semantic relationships between real-world concepts [82, 113]. It has been proven to be a useful resource that could effectively help improving search results, even for poorly performing (or difficult) queries [63, 65, 74]. When expansion terms are selected from an external resource (ConceptNet), they are less dependent on the initial results list. This may solve the problem of existing methods which mainly rely of the initial retrieval results. We also assume that most aspects for query terms exist in such a general knowledge base. By selecting different related concepts to expand the query, we can produce a more diversified query that can cover multiple aspects of the original query. As a consequence, the search results may provide a better coverage of the different aspects of the query. However, for the purpose of SRD, it is inappropriate to perform a unique expansion for the whole query. Rather, one should try to expand different aspects of the query, or to perform a diversified query expansion (DQE). Our approach is based on a similar principle to *MMR*: *Maximum Marginal Relevance* [22], which tries to select documents that are both relevant to the query and different from the documents already selected. In our case, we select expansion terms that are related to the initial query, and different from the previously selected expansion terms. We will call the approach *MMRE*: *MMR-based Expansion* (Bouchoucha *et al*. [13]). We extensively evaluate our approaches using ClueWeb09 (category B) documents' collections, and the publicly available query sets of TREC 2009, 2010 and 2011 Web tracks. Our experimental results show that our proposed DQE method significantly outperforms traditional diversification methods which rerank the initial retrieval results. This clearly shows that diversifying the expansion terms of a query may be more effective than diversifying the documents.

Contribution 1.2 - using multiple resources: To solve the problem of the lack of coverage of one resource regarding to the query aspects, we propose a unified framework to combine multiple resources for DQE. We believe that multiple resources tend to complement each other for DQE, and

4. `http://wordnet.princeton.edu`
5. `http://conceptnet5.media.mit.edu`

by integrating multiple resources, the expansion terms added can cover more intents of the query, thus increase the effectiveness of SRD. Our framework is general and can integrate any resource (Bouchoucha *et al*. [14]). Our experimental results show that combining multiple resources performs better than using any single resource for the purpose of DQE, and that multiple resources are complementary which may help to maximize the coverage of query aspects.

Part 2: Resource weighting for DQE

Contribution 2.1 - Query-dependent resource weighting: We introduce the resource weighting task to a DQE based SRD system. More precisely, we propose a linear regression model to learn the weight of a resource for each query, based on a set of features that we derive (Bouchoucha *et al*. [16]). We experimentally show the advantage of the query level resource weighting over uniform weighting and non-query level resource weighting. This leads to select more diversified expansion terms.

Part 3: Aspect-level DQE

Contribution 3.1 - modeling of latent query aspects: The missing element in the existing Term-level DQE approaches is an explicit model for the underlying *aspects* of the query, with respect to which the selected expansion terms should be diversified. By query aspects, we mean the latent semantic dimensions, similar to topic models in LDA [10], that could be used to describe different query intents. Consequently, we propose a unified and general framework for latent semantic aspect embedding which considers the semantic relationship between expansion terms and their capability to cover uncovered aspects in order to create latent semantic aspects to represent the potential intents of a query. Our approach is based on embedding to automatically learn the possible aspects of a query. A noticeable difference from previous approaches such as LDA is that in our case the latent aspects are learnt to reflect some known semantic relations between terms (*e.g.*, through existing linguistic resources such as ConceptNet [113] or WordNet [91]), rather than merely to generate the documents. For example, for query "java", if *programming* and *algorithm* are known to be semantically related (similar), then we would like to create aspects such that these terms can be mapped into the same aspect(s), while *indonesia* will be mapped into a different aspect since it is semantically related neither to *programming* nor to *algorithm* (it corresponds to another aspect of Java which is *tourism*). In so doing, created aspects can naturally encode our knowledge about semantic relations between terms.

Another way to look at our approach is to consider the relations between terms found in different resources as constraints when the latent aspects are generated - Similar terms should correspond to the same aspects. Such constraints are natural: Without an explicit definition of aspects a priori (which is a difficult task in itself), the best way to define aspects is to rely on the known relations between terms. Besides, according to our investigation, an expansion term usually covers only a few aspects of the query. This inspires us to consider a sparsity constraint, and directly integrate it in our method when modeling query aspects. In Section 5.3.2 and Section 5.5.3, we explain in more detail the reason of using the sparsity constraint in our model and its effectiveness on the overall performance of our approach.

Using the same dataset (ClueWeb09-category B documents' collections), and the same query sets (those of TREC 2009, 2010 and 2011 Web tracks), we experimentally show the advantage of aspect modeling compared to the term-level DQE and to existing state-of-the-art diversification methods: our aspect-level DQE method significantly contributes in improving the effectiveness of SRD. We also show that sparsity constraint plays an important role in further improving the diversity of the search results.

1.5 Roadmap

The remainder of this dissertation is organized as follows: Chapter 2 first describes our experimental and evaluation methodologies and then reviews related work about search result diversification, diversified query expansion, the utilization of resources in IR and explains the connection between our work and related embedding works. In Chapter 3, we first describe in detail our DQE method for one resource namely ConceptNet (Contribution 1.1), and then extend it to be used for multiple resources (Contribution 1.2). Chapter 4 is dedicated to describe our query level resource weighting for DQE (Contribution 2.1). Chapter 5 introduces a novel method for latent semantic aspect embedding that explicitly models query aspects by presenting each expansion term as an aspect vector in the space of the query and allows integrating multiple resources (Contribution 3.1). Finally, in Chapter 6, we present the conclusions of this dissertation and outline some directions for future research. Note that these chapters mainly correspond to articles that have been published as part of this thesis. We also introduced some minor changes to these publications in order to provide further details and examples.

Chapter 2

Background and Related Work

We will focus in the first part of this chapter on describing test collections and sources of information that we used along this thesis to evaluate our methods and compare them with existing approaches. Then, we conduct a literature review of previous studies that are related to this dissertation. As this study aims to improve the state-of-the-art SRD approaches, we will review the major existing diversification methods. Due to the multitude of the proposed methods, we classify them into two categories, according to how they diversify the results: implicit SRD and explicit SRD, and into three strategies: coverage-based SRD, novelty-based SRD and hybrid SRD, according to which criteria is used to diversify search results. Thereafter, we will describe some recent methods which diversify the expansion terms of the query instead of diversifying the search results. Since integrating multiple resources belongs to our main interests in this thesis, we will also review some studies which attempt to use different resources and combine them in order to solve common problems in SRD (and also in general IR). Finally, the last part of this chapter will be dedicated to describe some approaches about embedding and their connection with our proposed methods on aspect embedding for DQE.

2.1 Evaluation Methods

The domain of Information Retrieval is built in the culture of hypothesis's validation through experimentation. The foci of these experimentations are the concept of relevance and evaluation methods. While different chapters in this thesis have different experimental setups, in this section, we describe the test collections, sources of information and evaluation metrics used in our experiments, which are common to all the chapters of this thesis.

2.1.1 Document Collections and Topics

Our experiments are conducted on the ClueWeb09 (category B) dataset [1]. We indexed these document collections using Indri / Lemur [2], which is an open-source IR system. Statistics of the index and the document collections are reported in Table 2.I.

Size (uncompressed)	1.5 TB
Number of English documents	50,220,423
Number of English documents judged relevant	14,842
Average number of relevant documents per query	99
Number of unique terms	87,262,399
Total number of terms	40,417,947,339
Average documents' length (in number of words)	805
Size of the index	586 Go

Table 2.I: Statistics about the index and the document collections.

It is worth noting that ClueWeb09 is till now the second largest Web collection which is available to the IR researchers (ClueWeb12 [3] is the largest one till now). The whole ClueWeb collections (category A and category B) involve more than one billion Web pages, written in ten different languages, half of which are in English. These documents were collected in January and February 2009. In this study, we only consider the category B which is available for us.

For the topics (*i.e.*, test queries), we use those of TREC. TREC (Text REtrieval Conference) [4] is one of the major evaluation campaigns. The first edition of TREC was held in 1992. TREC organizers provide a corpus of documents, a set of queries or topics which correspond to information needs, and their relevance judgements which map each topic to one or multiple documents assumed to be relevant for that topic. These data, known as *test collection*, are commonly used by the IR researchers to evaluate their methods and compare them with existing ones. Each year, several participating groups submit their system's results and work on different search tracks, such as Web track, Microblog track, Medical track, etc.

In this work, we use the 148 test queries from TREC 2009 [34], 2010 [31] and 2011 [32] Web tracks, henceforth refereed to as WT09, WT10 and WT11, respectively. Statistics about these query

1. http://www.lemurproject.org/clueweb09.php
2. http://www.lemurproject.org/indri
3. http://www.lemurproject.org/clueweb12.php
4. http://trec.nist.gov/

sets are reported in Table 2.II. We exclude queries 95 and 100 since no relevance judgements are available for them. For these 148 queries, TREC assessors also provide the corresponding relevance judgements enabling the evaluation of adhoc and diversity search approaches. For each topic, TREC assessors identify from 2 to 8 subtopics. Figure 2.1 illustrates an example of a topic from WT09 ("cell phones"), along with its 8 manual subtopics.

Year	*Number of Queries*	*TREC Query Numbers*	*Average Number of Query Subtopics*	*Average Query Length (Nb. of non-stopword terms)*
2009	50	1 - 50	4.9	1.9
2010	48	51 - 99	4.4	1.6
2011	50	101 - 150	3.4	3.0

Table 2.II: Statistics for query sets being used.

In our experiments, the *query* field of a topic is used as the original query. Each topic has a *description* field which provides a brief summary for the general information need behind the query. Each topic is also categorized as either *ambiguous* or *faceted*. Ambiguous queries (*e.g.*, "java") have multiple distinct interpretations (*e.g.*, 'language', 'island', 'coffee'), while faceted queries (*e.g.*, "cell phones") are under-specified ones with different aspects covered by subtopics (*e.g.*, 'prepaid cell phones', 'phone companies', 'unlocked phones'). In turn, each subtopic is categorized as being either informational (*inf*) or navigational (*nav*), as judged by TREC assessors. In the former, the user is seeking for some information related to the query, while in the latter, the user is seeking a specific URL [34]. Note that in this study, we treat all the queries in the same way and we don't explicitly distinguish the query type (ambiguous/faceted), nor the type of their subtopics (informational/navigational). In the future, we will consider these issues. The consideration of these factors in result diversification is an interesting aspect to be investigated in the future.

2.1.2 Resources

In this dissertation, we consider four typical sources of information, which are available to us:
(1) The last version of ConceptNet [5] which is actually the largest commonsense knowledge base;
(2) The English Wikipedia dumps [6] of July 8th, 2013;

5. http://conceptnet5.media.mit.edu
6. http://stats.wikimedia.org/EN/Sitemap.htm

```
<topic number="34" type="faceted">
  <query> cell phones </query>
  <description>
  Find information about cell phones and cellular service providers.
  </description>
  <subtopic number="1" type="inf">
    What free phones are available from different vendors?
  </subtopic>
  <subtopic number="2" type="nav">
    Go to AT&T's cell phones page.
  </subtopic>
  <subtopic number="3" type="nav">
    Go to Verizon's page that lists phones for sale.
  </subtopic>
  <subtopic number="4" type="inf">
    Find information on prepaid cell phones.  What companies offer them?
    What kind of phones are available?
  </subtopic>
  <subtopic number="5" type="nav">
    Go to Nokia's home page.
  </subtopic>
  <subtopic number="6" type="inf">
    What cell phone companies offer Motorola phones?
  </subtopic>
  <subtopic number="7" type="nav">
    Go to Sprint's page that lists phones for sale.
  </subtopic>
  <subtopic number="8" type="inf">
    Where can I find information on buying unlocked phones?
  </subtopic>
</topic>
```

Figure 2.1: Example of a WT09 topic ("cell phones") along with its manual subtopics.

(3) The log data of Microsoft Live Search 2006 [1];
(4) The top 50 results returned for the original query, which correspond to Web documents originating from ClueWeb09-B document collections.

In our study, we use these resources to automatically extract candidate expansion terms for the purpose of better diversifying search results. In the remainder of this section, we will briefly describe the first three resources (please refer to Section 2.1.1 for a description of the document collections being used).

*** ConceptNet**

People have the ability of common-sense reasoning, which is the ability to understand and reason about things. However, computers lack this competence. ConceptNet was designed to encode common-sense relations for computers.

ConceptNet was first designed as a project in the MIT (Massachusetts Institute of Technology) Media Lab. It was built through the collaboration of over 14000 authors, who brought their expertise and knowledge in several domains such as, computer science, mathematics, physics, art, sports, etc. Hence, the (semantic) relations involved in ConceptNet reflect well the understanding of human beings in different areas.

Currently, ConceptNet 5 includes more than 1.6 million assertions and it is linked with 20 different semantic relations such as, *isA*, *UsedFor*, *CapableOf*, *PartOf*, *LocationOf*, etc [113]. The nodes used in ConceptNet represent semi-structured natural language fragments and correspond to real world concepts. Figure 2.2 below presents a fragment of the graph of ConceptNet. As opposed to WordNet, ConceptNet is not limited to some "basic" relations such as synonyms, hyponyms, hypernyms, but extends them to more complex and interesting semantic relations such as causal, spatial and functional assertions. The network-based structure of ConceptNet opens up possibilities for making complex and multi-step inferences. For example, from Figure 2.2, it follows that the concepts "*house*" and "*chair*" are connected via the following chain of inferences: "*in house*" $\rightarrow$ "*kitchen table*" $\rightarrow$ "*chair*". Table 2.III reports some statistics about ConceptNet [113] that we use in our study.

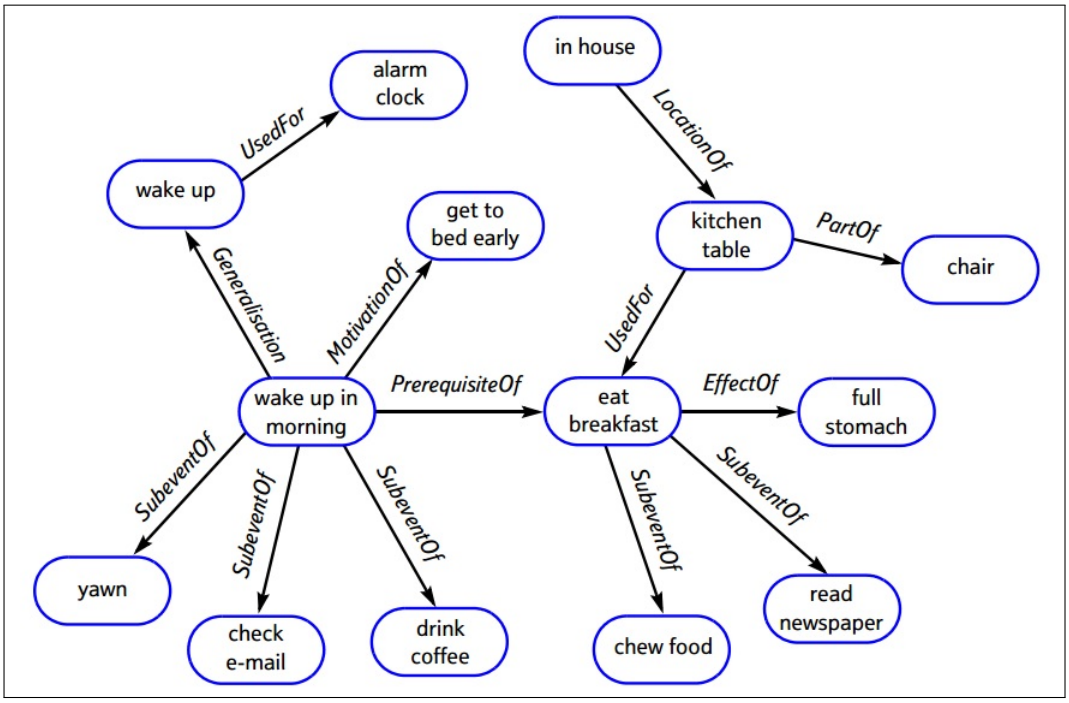

Figure 2.2: Fragment of the graph of ConceptNet (adapted from [82]).

Size (uncompressed)	2.9 GB
Number of assertions (predicates)	8.7 M
Number of nodes (concepts)	3.9 M
Number of different semantic relations	20

Table 2.III: Statistics about ConceptNet.

*** Wikipedia**

Wikipedia [7] is an online valuable source of information which contains a large number of articles (pages) in different languages. In our study, we only consider the English Wikipedia pages. The Wikipedia articles are manually built by humans who share their knowledge and expertise in different domains. Each Wikipedia page contains an amount of information related to a specific topic (or domain). Different Wikipedia pages could also be connected using anchor texts and redirection pages. In our study, we exploit the rich content of Wikipedia pages in order to extract candidate expansion terms for the purpose of SRD. Table 2.IV reports some statistics about the Wikipedia dumps of July

7. `https://en.wikipedia.org`

8th, 2013 that we use in our study.

Size (uncompressed)	50 GB
Total number of articles (for all the languages)	28.1 M
Number of English articles	4.3 M
Number of English words	170 M
Number of English outlinks	6.5 M

Table 2.IV: Statistics about the Wikipedia dumps.

*** Query logs**

Query log is a valuable resource that describes the search behavior of a user. It can be used to predict how people interact with the search system. Due to the important amount of data that it contains (including query reformulations and URL clicks, stored in several user sessions), query logs could be exploited for several tasks in IR, such as query expansion, text retrieval, image retrieval, etc.

In our study, we use the MSN 2006 query logs which spans over one month (starting from May 1st) and contains a large number of queries which were submitted by US users to MSN search [8]. Most of these queries are in English. The log data is split into two files: the first file contains about 15 million queries with their corresponding user sessions, and the second file contains about 8.9 million queries with their corresponding clicks. Figure 2.3 shows an example of a subset from the first file of the query logs in which one can clearly distinguish for each query, the *Time stamp*, the *Query string*, the *Query ID*, an anonymous *Session ID*, and the *Result Count* which corresponds to the number of returned results for that query. Figure 2.4 shows an example of a subset from the second file of the query logs in which one can clearly distinguish for each query, the *Query ID*, the *Query string*, the *Time stamp*, the *clicked URL*, and the *Position* which corresponds to the rank of that URL in the result page. Since we want to exploit the information available in both two files, in our experiments, we combine these two files together according to the query string (and ignore case). In our study, we exploit the query terms (or query reformulations), the user sessions and the clicked URLs in order to extract candidate expansion terms for the purpose of SRD. Finally, we report in Table 2.V some statistics about the search log data that we use in our study.

8. In the MSN 2006 query logs, adult queries were extracted separately. In our experiments, we did not use these adult queries.

Time	Query	QueryID	SessionID	ResultCount
.				
.				
.				
2006-05-01 00:17:34	George Clooney	b038b48a2cd14190	031b5a17967b4cc6	10
2006-05-01 00:17:34	dell computers	c10c6dee59fc4ee4	1acfb383faff4e5d	16
2006-05-01 00:17:38	memorials statues	d2e9e15ab24643c1	12433941f97d4762	15
2006-05-01 00:17:38	richard boyd playwright	2f9374e740b64ca1	1a9757ea2c79400b	11
2006-05-01 00:17:40	CROSS DRESS	65e6f810f8ed4ef9	0b0e1f55ccb24497	18
2006-05-01 00:17:40	msn ratero	40fabfcafdfd45f9	222fa358fb4c495a	10
2006-05-01 00:17:41	costa rico in central america	0cc0d438839c4c57	1475d9f640df49ee	15
2006-05-01 00:17:41	honda financial	6ef7eca1cd3243d5	03e9e5c27d114176	15
2006-05-01 00:17:42	brick landscaping flower bed	f7b10c29c7a246a5	172135cf7cee4da7	16
2006-05-01 00:17:43	hollister hills events	0196d0ba2a764505	11a699b9433f47ab	10
2006-05-01 00:17:43	yahoo	57a18dbb6ede4409	09acbf51bf594147	10
.				
.				
.				

Figure 2.3: A subset of queries with their corresponding sessions from the 2006 MSN log data.

QueryID	Query	Time	URL	Position
.				
.				
.				
000278f5dd7e4806	san diego zoo	2006-05-11 18:12:43	http://www.sandiegozoo.org/	1
00027b7d50d449a3	clipart	2006-05-12 11:21:57	http://www.clipartslide.com/	8
00027c585e464767	GOOGLE	2006-05-15 06:42:58	http://www.google.com/	1
0002802de46a4972	house spiders	2006-05-17 12:11:46	http://ohioline.osu.edu/hyg-fact/2000/2060.html	1
0002802de46a4972	house spiders	2006-05-17 12:12:38	http://www.zoo.org/educate/fact_sheets/spiders/giant.html	2
0002802de46a4972	house spiders	2006-05-17 12:12:59	http://www.west-ext.com/house_spider.html	3
0002803f2c874bf4	BABY NAMES	2006-05-10 11:59:51	http://www.babyzone.com/babynames/	5
0002807a15834f8c	weather in panama	2006-05-19 12:42:16	http://weather.yahoo.com/regional/PMXX.html	9
00028178bc294b6a	camput	2006-05-24 14:19:37	http://www.camput.org/	1
.				
.				
.				

Figure 2.4: A subset of queries with their corresponding clicked URLs from the 2006 MSN log data.

Number of queries for the first file	14,921,285
Number of queries for the second file	8,832,457
Number of unique queries	6,623,635
Number of user clicks	12,251,067
Number of clicks per query	1.387
Number of user sessions	7,470,915

Table 2.V: Statistics about the search log data.

2.1.3 Evaluation Metrics

The main purpose of IR is to fulfil the user information need which could be expressed by a query. A good search engine is the one that selects the maximum of relevant documents and the minimum of non-relevant ones. Several methods could be used to evaluate the quality of a search engine. While some evaluation methods focus on user studies to understand her behavior in front of the search engine, other methods instead rely on a set of evaluation metrics to quantify the quality of a search engine [71]. In this dissertation, we use the latter method to evaluate our approaches by computing metrics on our system. Such pre-defined metrics compare the retrieval results obtained by a search engine with the relevance judgements which are already provided. TREC assessors select the top k documents returned by the participants' systems, and use a *pooling* method to manually choose from these k documents a sample which will be used as relevance judgements for the IR community. In our experiments, we use the relevance judgements and the evaluation metrics provided by TREC assessors in order to evaluate our approaches and compare them with other existing ones. Since our purpose is twofold: to improve the diversity of search results, and also to improve their relevance , we will present, in the remainder of this section, an overview of the (official) relevance and diversity metrics used at TREC.

* Relevance Metrics

Several metrics have been proposed in the literature in order to evaluate the ability of search engines to retrieve relevant documents and rank them in the top of the results list, for the purpose of better satisfying the user information need underlying her query. Given a query, a document is viewed to be relevant for that query if it contains any amount of information that may satisfy the information need of the user who submitted that query. The relevance of a document could be defined as its usefulness for the query, or its relation and correspondence to the query, or maybe the degree of surprise that it could bring to the user (novel information), etc [9]. In this section, we review three relevance metrics which are official in the adhoc task of TREC Web track.

MAP (Mean Average Precision) [5]: MAP@k is defined as the arithmetic mean average precision

9. These definitions are from the course of Information Retrieval (IFT 6255) of Professor Jian-Yun Nie which could be found at this link: `http://www.iro.umontreal.ca/~nie/IFT6255`.

over a set of topics T, as follows:

$$MAP@k = \frac{1}{|T|} \sum_{Q \in T} AveP(Q)@k \tag{2.1}$$

where $AveP(Q)@k$ is the average precision at rank k of the retrieved results of query Q, which is defined as follows:

$$AveP@k = \frac{1}{|Rel|} \sum_{i=1}^{k} relevant(i) \cdot P@i \tag{2.2}$$

where $relevant(i) = 1$ if the document at rank i is relevant, and 0 if not; $|Rel|$ is the total number of relevant documents found at the first k returned documents; and $P@i$ is simply the precision score of the first i returned documents (*i.e.*, the number of documents that are relevant in the top i returned results divided by the number of retrieved documents from the set of i returned results).

nDCG (normalized Discriminative Cumulative Gain) [5]: This measure is used to evaluate the usefulness or the *gain* of a document based on its position (or rank) in the result list. This gain is accumulated from the top to the bottom of the results' list and may be reduced, or *discounted*, at lower ranks [44]. The DCG is the total gain accumulated at a particular rank k, and defined as follows:

$$DCG@k = relevant(1) + \sum_{i=2}^{k} \frac{relevant(i)}{log_2(i)} \tag{2.3}$$

where $relevant(i)$ is the relevance level of the document retrieved at position i. In Formula 2.3, $log_2(i)$ is the discount or reduction factor that is applied to the gain.

In general, the result sets as well the number of topics used to test the effectiveness of a search engine, may vary in size. Hence, it is important to *normalize* Formula 2.3 so that the performance of several systems could be fairly compared. This leads to the following normalized DCG metric:

$$nDCG@k = \frac{DCG@k}{IDCG@k} \tag{2.4}$$

where nDCG@k is the normalized DCG score at rank k, and IDCG@k is the *ideal* DCG scores at the same rank.

ERR (Expected Reciprocal Rank) [26]: ERR is a cascade based metric which estimates the probability that the user stops at the rank k. ERR@k is defined as follows:

$$ERR@k = \sum_{i=1}^{k} \frac{1}{i} \prod_{j=1}^{i-1} (1 - p_j) \cdot p_i \tag{2.5}$$

where p_i is the probability of the i^{th} document being relevant to the query. In Formula 2.5, the product $\prod_{j=1}^{i-1}(1-p_j)$ denotes the probability that none of the documents ranked higher than the i^{th} document is relevant.

*** Diversity Metrics**

Several metrics have been proposed in the recent years in order to evaluate the diversification effectiveness of search engines. A good diversification system is the one that satisfies multiple information needs (or user intents) underlying a query that is submitted to that system by different users, or by the same user in different contexts. In the context of search result diversification, a query is represented by a set of *subtopics* or *aspects* (which generally correspond to user intents). The relevance of a document with respect to a query is judged separately for each subtopic, and is estimated by the ability of that document to cover different subtopics of the same query. In this section, we review five diversity metrics which are official in the diversity task of TREC Web track.

α-nDCG (α-normalized Discriminative Cumulative Gain) [33]: α-nDCG@k is computed as follows:

$$\alpha - nDCG@k = \frac{\alpha - DCG@k}{\alpha - DCG'@k} \tag{2.6}$$

where α-DCG'@k is a normalization factor corresponding to the maximal value of α-DCG@k that gives the ideal document ranking. α-DCG@k is computed as follows:

$$\alpha - DCG@k = \sum_{j=1}^{k} \frac{\sum_{s \in S(Q))} rel(d_j, s))(1-\alpha)^{\sum_{i=1}^{j-1} rel(d_i, s))})}{log_2(1+j)} \tag{2.7}$$

In Formula 2.7, the parameter α ($\alpha \in [0,1]$) represents the user satisfaction factor for the set of documents that have been already browsed by the user. This parameter (α) is generally fixed to 0.5.

For instance, suppose that the user has found a relevant document at the first position. In that case, the user is satisfied for some aspect s of Q. Therefore, a high score (close to 1) to the parameter α will be assigned. Once the user has found her information needed, less importance will be given to the following documents (starting from the second position). Otherwise, if the user hasn't fulfil her information need, she will continue browsing the result list until she founds a document which is relevant for her. In such a case, a small value will be assigned to the parameter α, which means that a higher importance will be attributed to the next coming documents in the retrieved results' list.

The relevance feedback (RF) and pseudo-relevance feedback (PRF) are the most used techniques to evaluate the user satisfaction. In Formula 2.7, Q is a query; $S(Q)$ is the set of subtopics underlying Q; and d_i (resp. d_j) is the document ranked at the i^{th} (resp. j^{th}) position. $rel(d,s)$ is a function that evaluates the relevance of a document d with respect to a given subtopic s. Note also that α-nDCG considers the set of already (k-1) selected documents when evaluating a document at position k. This means that the metric takes into account the dependency between the returned documents. Finally, note that $(1-\alpha)^{\sum_{i=1}^{j-1} rel(d_i,s)}$ penalises the coverage of already covered aspects of the query and α controls the amount of penalization.

ERR-IA (Expected Reciprocal Rank - Intent Aware) [27]: ERR-IA(Q, D) for a given query Q and over a set of returned documents D with respect to Q is defined as follows:

$$ERR-IA@k = \sum_{s \in S(Q)}^{k} p(s|Q) \cdot ERR(s,D) \tag{2.8}$$

where $ERR(s,D)$ is computed *separately* for each subtopic s of Q using Formula 2.5; and $p(s|Q)$ denotes the importance of subtopic s regarding to the query Q (the more popular the subtopic s for Q, the higher is $p(s|Q)$). Of course, we assume our knowledge is complete, *i.e.*, $\sum_{s \in S(Q)} p(s|Q) = 1$ where $S(Q)$ is the set of possible subtopics for Q.

NRBP (Novelty- and Rank-Biased Precision) [30]: NRBP is an extension of the RBP (Rank-Biased Precision) metric [92]. The basic intuition that NRBP uses is that, the user has some specific intent and is generally interested in one particular aspect (or nugget) of the query, at least at that time. For instance, following Clarke *et al.* [30], we mention the example of query on "Windows": "If a user is

interested in buying windows for house, we might guess that they are not interested in the Windows operating system, at least at that instant". NRBP is defined as follows:

$$NRBP = \frac{1-(1-\alpha)^{\beta}}{N} \cdot \sum_{k=1}^{\infty} \beta^{k-1} \cdot \sum_{i=1}^{N} J(d_k, i)(1-\alpha)^{C(k,i)} \tag{2.9}$$

Here, d_k denotes the k^{th} document; N is the (possible) number of nuggets (or aspects) of a given query; $J(d,i) = 1$ if document d is relevant to the i^{th} aspect (or nugget) of the query , and $J(d,i) = 0$ if it is not; $C(k,i)$ is the number of documents at cut-off k that have been judged to be relevant to the i^{th} aspect of the query; parameter $\beta \in [0,1]$ is used to model the patience level of the user [10]; and parameter $\alpha \in [0,1]$ refers to the user declining interest. Finally, similar to α-nDCG [33], $(1-\alpha)^{C(k,i)}$ penalises the coverage of already covered aspects of the query and α controls the amount of penalization.

S-recall (Subtopics - recall) [125]: S-recall@k measures the percentage of the subtopics covered by the top k ranked results.

$$S-recall@k = \frac{|\bigcup_{i=1}^{k} subtopics(d_i)|}{n_Q} \tag{2.10}$$

where n_Q is the possible number of subtopics for a given query Q, and $subtopics(d)$ is the set of subtopics to which document d is relevant.

Prec-IA (Precision - Intent Aware) [2]: Prec-IA@k is defined using Formula 2.11.

$$Prec-IA@k = \sum_{c \in C(Q)} p(c|Q) \cdot Prec(Q|c)@k \tag{2.11}$$

where $C(Q)$ denotes the set of categories to which Q belongs to; $p(c|Q)$ is the probability of query Q belonging to the category c; and $Prec(Q|c)@k$ is the (standard) precision score of the top k ranked documents regarding to the category c.

10. Once the user has browsed the first document in the results' list, the probability of moving to browse the second document in the same results' list is β, and $(1-\beta)$ otherwise.

2.2 Search Result Diversification (SRD)

In this section, we review the existing methods on SRD, and present some applications of SRD in general IR.

2.2.1 The SRD Problem

SRD tries to select relevant but diversified results among the top results. It is known to be *NP-hard* [2, 23, 56, 59]. For instance, in [46], the authors conduct a theoretical study of the diversification problem and show that existing approaches on SRD are very complex. It can be seen as an optimization problem whose purpose is to determine an order (or a ranking) of documents, so as to cover as much as possible the different query aspects. SRD aims to identify relevant information under the uncertainty posed by query ambiguity. Its effectiveness is dependent on both the relevance of returned documents, and their ability to fulfill multiple user intents, with respect to the user query. SRD could be seen as a generalization of the standard ranking problem [57, 100], where the challenges to be met are:

C_1. Satisfy multiple information needs behind the user query.

C_2. Avoid redundancy in the ranking.

The first challenge (C_1) is due to the query *ambiguity* problem, meaning that a query can have several aspects (or interpretations). It is nevertheless not clear which aspect the user is concerned with. The second challenge (C_2) is due to the fact that, once a document d satisfying the user information need has been observed, another document $d^{'}$ that satisfies the same user information need as d, is seen to be no longer useful (or redundant) for the user. It was shown that the relevance of a document in a ranking should be estimated *dependently* of the relevance of the documents ranked above it [125]. In other words, a good search engine must consider the relevance of a document in light of the other retrieved documents. It is hence important to remove such redundant document ($d^{'}$) from the ranking list.

Query ambiguity can be tackled by ensuring a high *coverage* of the possible information needs underlying the query, and document redundancy can be tackled by ensuring a high *novelty* for the set of returned documents [33]. By maximizing coverage and minimising redundancy with respect to the aspects underlying a query, SRD can effectively meet these two above challenges (C_1 and C_2)

[33]. Note that a high coverage does not necessarily imply a high novelty, and vice versa. Indeed, covering all the user needs with respect to a query does not guarantee that the selected documents are not redundant. Conversely, a ranking with a maximum of novelty does not guarantee that the returned set of results cover (almost) all the query aspects.

Several studies have been proposed in the literature. While some studies have used the search diversification principle in some specific domains, others have rather attempted to propose new methods to effectively diversify search results in general. According to that, we begin this section by reviewing the existing diversification methods which are state-of-the-art, then we present some studies that attempt to use SRD in several practical applications in IR.

2.2.2 Existing Methods in SRD

Based on the discussion in Section 2.2.1, we first classify the existing SRD methods into three strategies depending on the criteria used to diversify search results: *coverage-based SRD*, *novelty-based SRD*, and *hybrid SRD* which combines both coverage and novelty. We can also classify the methods into two categories, depending on how they represent the query aspects: *implicit SRD* approaches and *explicit SRD* approaches. While implicit SRD promotes dissimilar documents through the relations among documents to produce ranked lists that convey both relevant and diverse information about a query, explicit SRD attempts to cover as much as possible the different aspects (or subtopics) of the query (which are either manually defined or automatically extracted) based on the relation between the documents and the query subtopics [103, 104]. Based on that, we propose to organize these approaches according to these two complementary dimensions: *diversification strategy* and *aspect representation*. Table 2.VI summarizes the most significant approaches in SRD, according to this organization. In the remainder of this section, we review existing diversification approaches according to these two dimensions.

Implicit SRD Approaches

**** Novelty-based Methods:***

As highlighted in Table 2.VI, the majority of implicit SRD approaches adopt a strategy based on novelty. The importance of novelty has been demonstrated in several studies. For instance, Xu and

Diversification Strategy	*Aspect Representation*	
	Implicit	*Explicit*
Novelty (or *Non-Redundancy*)	Carbonell and Goldstein (1998) [22] Zhai *et al.* (2003) [125] Zhai and Lafferty (2006) [128] Chen and Karger (2006) [28] Zhu *et al.* (2007) [132] Wang and Zhu (2009) [120] Gollapudi and Sharma (2009) [56] Rafiei *et al.* (2010) [98] Gil-Costa *et al.* (2011) [36] Gil-Costa *et al.* (2013) [54]	Demidova *et al.* (2010) [45] Dou *et al.* (2011) [51] Santos *et al.* (2012) [103]
Coverage	Carterette and Chandar (2009) [24] He *et al.* (2011) [60]	Radlinski and Dumais (2006) [96] Radlinski *et al.* (2008) [97] Capannini *et al.* (2011) [21] Zheng *et al.* (2011) [130] Santos *et al.* (2012) [103] Dang and Croft (2012) [43] Dang and Croft (2013) [42]
Hybrid	Yue and Joachims (2008) [124] Raman *et al.* (2012) [99]	Agrawal *et al.* (2009) [2] Zheng *et al.* (2010) [131] Santos *et al.* (2010) [104] Santos *et al.* (2010) [105] Santos *et al.* (2010) [106] Liang *et al.* (2014) [81]

Table 2.VI: Existing SRD approaches, organized into two complementary dimensions: aspect representation and diversification strategy.

Yin [123], Clarke *et al.* [33] and Gollapudi and Sharma [56] show that it is important that current search engines take into account the novelty criterion.

Carbonell and Goldstein [22] propose a method called *Maximum Marginal Relevance* (*MMR*), which is the first implicit SRD method based on novelty. *MMR* is an early representative method of implicit SRD and it is one of the most popular approaches in document diversification. *MMR* aims to balance the relevance and the diversity of a ranked list, by selecting documents that maximize relevance and reduce redundancy with respect to higher ranked documents. The following formula is

used to select a document at each round:

$$MMR(D_i) = \lambda \cdot rel(D_i, Q) - (1 - \lambda) \cdot \max_{D_j \in S} sim(D_i, D_j) \tag{2.12}$$

where Q denotes a query, D_i is a candidate document from a collection, and S is the set of documents already selected so far. The parameter λ controls the trade-off between relevance and novelty (*i.e.*, non-redundancy). $rel(.,.)$ and $sim(.,.)$ are two functions that determine respectively the relevance score of the candidate document to the query and its similarity to a previously selected document. In each step, *MMR* selects the document with the highest *MMR* score. In [22], *MMR* was applied for text retrieval and summarization.

Several studies extend *MMR* [103, 128] and apply it in different domains [26, 37]. Zhai and Lafferty [128] and Zhai *et al.* [125] propose another version of *MMR*, called *MMR loss function*, within a risk minimization framework in language modeling [127]. The authors model IR as a decision problem. The user preferences are seen as a loss function, and document retrieval becomes a problem of risk minimization (please refer to the thesis of ChengXiang Zhai [126] for more details). For each new query, the user judges the relevance of the returned documents by stating her feedback. Whenever a document is considered to be irrelevant regarding to the user intent, a loss is added to the retrieval model. Via the definition of *MMR loss function*, the authors demonstrate that the risk of irrelevance decreases when the document is selected such that it is both relevant and non redundant to the documents already selected. This function is applied to automatically mining subtopics for a given query. In addition, both Zhai and Lafferty [128] and Zhai *et al.* [125] observe this optimization problem from a risk minimization view, and they don't consider whether the selected documents can cover the different aspects of the query. It is important to consider this aspect to better understand the user intent behind her query.

Chen and Karger [28] apply the novelty principle for the problem of query abandonment [44]. They consider the case of ambiguous queries for which the probability of abandonment is generally high. Their approach selects the document more likely to introduce novel information, compared to the set of documents already selected. Based on that, they introduce a sequential document selection algorithm to optimize an objective function aiming to maximize the chance of finding at least one relevant document for all the users. They demonstrated that the probability of abandonment decreases

significantly, which reflects that the user satisfaction is increased. However, their approach is not realistic, since the user intents are different, and it is rare to find one document that can satisfy together *all* users.

Zhu *et al.* [132] use random walks on an absorbing Markov chain to prevent redundant items from receiving a high rank by tuning ranked items into absorbing states. The absorbing states decrease the importance of items that are similar to them, thereby promoting items that are dissimilar to them.

Wang and Zhu [120] introduce a new diversification approach called *MVA* (*Mean Variance Analysis*), which is inspired by the modern portfolio theory (MPT) [11] in finance. *MVA* is similar to *MMR*, in the sense that both of them consider a trade-off between relevance and non-redundancy. However, unlike *MMR* which evaluates the redundancy in terms of similarity between documents, *MVA* defines the redundancy by observing how the relevance score of a document is correlated with those of the other documents. Indeed, the authors consider both the average and the variance of the relevance scores of the returned documents. Given a portfolio of a limited number of places (n), the idea consists of iteratively selecting a set of n documents ensuring the maximization of a gain (mean) that corresponds to a high relevance of the whole set of n documents, while minimizing the risk (variance) by reducing the redundancy of this set of documents. In each iteration, the selected document (d) is the one that maximizes the following objective function:

$$\mu_d - b \cdot w_i \cdot \sigma_d^2 - 2 \cdot b \cdot \sigma_d \cdot \sum_{d_j \in D_Q} w_j \cdot \sigma_{d_j} \cdot \rho_{d,d_j} \tag{2.13}$$

where μ_d and σ_d^2 are respectively the mean and the variance of the relevance estimates associated with document d, and the summation component estimates the redundancy of d in light of the whole set of returned documents (D_Q) with respect to an original query Q. Here, ρ_{d,d_j} refers to the well-known Pearson correlation [12] of the relevance estimates of the two documents d and d_j, w_i is a weight in [0,1] corresponding to the discount of the document at the i^{th} ranking position (the more the document is top ranked, the less the discount is, which promotes the documents ranked in the top of the list). In Formula 2.13, the parameter b is used to control the trade-off between relevance, variance and redundancy. A very similar approach was also proposed by Rafiei *et al.* [98]. Later, Santos *et*

11. http://en.wikipedia.org/wiki/Modern_portfolio_theory
12. http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient

al. [103] propose *xMVA* (*explicit Mean Variance Analysis*), which is an extension of *MVA* [120] based on how well the documents satisfy the explicitly represented query aspects. The optimization frameworks proposed in Rafiei *et al.* [98] and in Wang *et al.* [120] are based on risk minimization which purpose is to minimize the redundancy of documents being selected. However, these studies do not take into account the coverage criterion which is important for the purpose of SRD.

Gollapudi and Sharma [56] characterize the problem of result diversification within an axiomatic framework. They develop a set of axioms that a diversification system is expected to satisfy, and show that there is no diversification function that can satisfy all these axioms simultaneously. Therefore, they introduce a set of redundancy functions to characterize the proposed axioms. Finally, they conduct a large-scale evaluation based on data derived from Wikipedia disambiguation pages.

One drawback of most SRD methods based on novelty is that they attempt to compare several documents in order to promote novelty. The number of comparisons between documents quickly increases when the number of documents increases, which makes these approaches expensive in practice ($O(n^2)$ document pair comparisons where n is the number of returned documents). Gil-Costa *et al.* [36, 54] propose to use several techniques to partition the initial ranking of a query into zones, such that each zone groups together similar documents. Using this method, they were able to drastically reduce the number of comparisons required to promote novelty.

**** Coverage-based Methods:***

While several approaches on SRD adopt a strategy based on novelty (or non-redundancy), other studies instead attempt to use a strategy based on coverage. Carterette and Chandar [24] formalize the SRD problem as an optimization problem. They propose a probabilistic approach for maximizing the coverage of multiple query aspects. These aspects are generated by constructing either relevance models [77] or topic models [10] from the top retrieved documents of the query. Afterwards, they select the highest scored documents for each aspect and then combine them using a round-robin fashion. Despite the usefulness of the proposed framework, the authors don't consider whether the selected documents are non-redundant. This criterion is important to be considered for the purpose of SRD.

More recently, a similar approach was introduced by He *et al.* [60] who proposes to partition the set of documents initially retrieved into non-overlapping clusters. This partitioning is based on topic models [10] in which each cluster covers one possible topic (aspect) of the original query Q. For each

cluster c, they assign a score $p(c|Q)$ based on the cluster likelihood of generating the query Q. Hence, the coverage-based diversification approach consists of selecting the most relevant documents from the high scored clusters. The selection strategy is based on the utilization of a weighted round-robin technique which performed the best.

**** Hybrid Methods:***

While some existing implicit SRD approaches use a novelty-based criteria and others use a coverage-based criteria, some existing approaches on SRD attempt to combine both novelty and coverage aiming to take advantage of both criteria. Yue and Joachims [124] propose a method for learning a function to diversify the search results, taking into account several features. Using *SVM* (*Support Vector Machines*), this function predicts diverse subsets of documents, so that each subset corresponds to one aspect of the query. Through experiments on a TREC collection, the authors demonstrate the effectiveness of their function (having a linear complexity) and show that it outperforms other existing methods that do not use machine learning techniques. However, the proposed method separates the concept of *diversity* from the concept of *relevance*, which is not realistic, since at the same time that we seek for documents covering several aspects, we also should promote the most relevant ones. Moreover, their approach assumes that the query aspects are known a priori, which is not the case in practice.

A similar approach was introduced by Raman *et al.* [99] who propose a new machine learning framework aiming to minimize redundancy and maximize coverage of the set of returned documents, with respect to an original query. Instead of using expensive training data, their algorithm learns within an online setting from implicit feedback (in the form of preferences between rankings). Such algorithm was shown to be more effective than other supervised learning algorithms in term of optimizing the trade-off between relevance and diversity.

Explicit SRD Approaches

As highlighted in Table 2.VI, the majority of explicit SRD approaches adopt a strategy based on coverage since such approaches require that the aspects of the query are known (which are often manually determined).

**** Coverage-based Methods:***

A multitude of explicit SRD approaches based on coverage were proposed in the literature. The work of Radlinski and Dumais [96] was the first that depicts the direction towards explicit SRD approaches. The authors propose to find for each query Q other queries that are related to Q, from the search log data within a 30 minutes time window. Such a time window is often used to segment query logs into sessions, each for a unique information need. The more query reformulations are executed, the higher the diversity is. In Radlinski and Dumais [96], reformulated queries are defined as queries that are found in the search logs having at least one common word with the original query Q, and that the user submitted at least two queries related to Q within a minute of each other. The authors develop three methods for generating the set of related queries from which *MRV* (*Maximum Result Variety*) is the most powerful one. In each iteration, *MRV* selects a query according to the following formula:

$$argmax_{Q_j}(\lambda \cdot p_{ij} - (1-\lambda) \cdot max_{Q_k \in R(Q_i)} p^*_{jk}) \tag{2.14}$$

where $R(Q_i)$ is the set of queries that are related to a test query Q_i, p_{ij} is the empirical probability that Q_i was followed by Q_j in the log, and $p^*_{ij} = p^*_{ji} = \sqrt{p_{ij} \cdot p_{ji}}$ is the related symmetric measure between the two queries Q_i and Q_j. Formula 2.14 uses a very similar principle to that adopted by *MMR* [22]: it greedily selects queries that are frequent reformulations (using p_{ij}) but different from other queries that have already been selected (using p^*_{ij}). Finally, the parameter λ in Formula 2.14 is used to control the trade-off between these two components.

Later, Radlinski *et al.* [97] reformulate the learning to rank problem, by considering the dependency between documents. They propose a function that learns to diversify the ranking of documents based on user clicking behaviour. They define an online learning approach that aims to maximize the coverage of clicks for a given query. Their approach is based on the assumption that users with different intents would click on different documents for the same query. They experimentally showed that such approach can maximize the probability of relevance while reducing the probability of query abandonment.

Capannini *et al.* [21] postulate that ambiguous queries need to be diversified more than other existing ones. Since the intent behind this kind of queries is usually not clearly defined, they propose to clarify the user information need by mining several queries from search log data, with a more

specific representation of the user intent [10] than the original query. In each iteration, the proposed algorithm (*OptSelect*) attempts to select the document that covers some identified aspects underlying the query. Finally, based on a series of experiments on a TREC Web Track collection, they show that *OptSelect* outperforms two existing diversification frameworks (*IASelect* [2] and *xQuAD* [105]) in term of scalability and response time.

Zheng *et al.* [130] claim that the majority of the proposed SRD functions based on coverage do not really cover the different query aspects because such functions are not sub-modular [13]. To break this limitation, the authors define a set of strategies that lead to derive five sub-modular coverage functions. The following proposed function (namely *SQR*) is shown to produce the best performance in terms of subtopics coverage:

$$SQR(s,d,D) = \lambda \cdot rel(Q,d) + (1-\lambda) \cdot coverage(Q,d,s,D) \tag{2.15}$$

where

$$coverage(Q,d,s,D) = \sum_{s \in S(Q)} (weight(s,w) \cdot cov(s,d) \cdot (2 - 2 \cdot \sum_{d' \in D} cov(s,d') - cov(s,d))) \tag{2.16}$$

Here, D denotes the document collection, $S(Q)$ is the set of possible aspects of the query Q, λ is a parameter that controls the trade-off between relevance and diversity (*i.e.* coverage), $rel(Q,d)$ is the relevance score of the document d with respect to the query Q, $weight(s,Q)$ is a function that measures the importance of the aspect s with respect to Q, and $cov(s,d)$ is another function that measures the degree of coverage of the aspect s with respect to the document d. Formulas 2.15 and 2.16 encourage to cover the aspects that have not been covered for the query, by promoting the selection of the documents which are likely to cover the missing aspects. However, the quality of this work is very dependent on the way in which the aspects are extracted. In other words, using two different methods to mining query aspects may yield to different results. Finally, the authors assigned uniform weights to all the query aspects ($\frac{1}{|S(Q)|}$) by assuming that all the query aspects have the same importance. This is not true since the users could be interested to one particular aspect more than other ones. Despite these limitations, we consider that this work [130] is one of the significant

13. http://en.wikipedia.org/wiki/Submodular_set_function

contributions in the state-of-the-art SRD approaches based on coverage.

The work of Santos *et al.* [103] is one of the most significant studies that belongs to explicit SRD approaches based on both coverage and novelty. In this work, the authors reported the results of a series of experiments to assess the role of the novelty in search diversification. They claimed that "[... existing diversification approaches based solely on novelty cannot consistently improve over a standard, non-diversified baseline ranking ...]". This surprising result downgrades the importance of novelty as a method of diversification. They observed that "[... the objectives of search result diversification are two-fold: (1) to maximize the number of query aspects covered in the ranking, and (2) to avoid excessive redundancy among the covered aspects]". Based on that, the authors attempted to *combine* both novelty and coverage to take advantage from each method. They conclude that novelty significantly contributes to improve the relevance and diversity of the documents when it is combined with coverage. Particularly, they empirically demonstrated that novelty "[... plays a role at breaking the tie between similarly diverse results]".

Dang and Croft [43] use the official subtopics manually identified by TREC assessors and suggestions provided by a commercial search engine as aspect representations, and propose a two-stage diversification framework called *PM-2* which, for each position in the result ranking list, first determines the aspect that best maintains the overall *proportionality* of the aspects covered and then selects the best document on that aspect. Their method selects documents in a greedy fashion using the Sainte-Laigue principle [14]. Hence, *PM-2* "[... is a probabilistic adaptation of the Sainte-Lague method for assigning seats to members of competing political parties such that the number of seats for each party is proportional to the votes they receive]". The following formula is used:

$$d^* = argmax_{d_j \in R}(\lambda \cdot qt[i^*] \cdot p(d_j|t_{i^*}) + (1-\lambda) \cdot \sum_{i \neq i^*} qt[i] \cdot p(d_j|t_i)) \tag{2.17}$$

Here, d_j denotes the j^{th} document from R, the set of documents that are relevant to the original query; t_i is the i^{th} subtopic (or aspect) which is related to the original query; and $p(d_j|t_i)$ is the probability that document d_j being relevant to query topic t_i. Parameter λ (which is tuned using two-fold cross validation on TREC 2009 and 2010 Web track query sets) controls the trade-off between the relevance to the aspect t_{i^*} and the relevance to more query aspects. In Formula 2.17, $qt[i]$ denotes the *quotient*

14. http://en.wikipedia.org/wiki/Sainte-Lague_method

score of the i^{th} document which corresponds to the number of votes that the i^{th} document has received (w_i) and the number of seats it has taken (s_i). $qt[i]$ is computed as follows:

$$qt[i] = \frac{w_i}{2 \cdot s_i + 1} \tag{2.18}$$

**** Novelty-based Methods:***

Demidova *et al.* [45] propose *DivQ*, a new framework for balancing the relevance and the novelty over structured databases. Instead of diversifying the set of returned documents with respect to a given query, *DivQ* attempts to diversify a ranked list of query interpretations. They first introduce a new probabilistic query disambiguation model in order to extract different interpretations of a query keyword, using several databases. Then, they propose a diversification schema for generating the k most relevant and diverse (*i.e.* non-redundant) query interpretations. Finally, they conduct an evaluation using two-real world databases, and they demonstrate that by using *DivQ*, the novelty of keyword search results over structured data can be substantially improved.

Dou *et al.* [51] argue that search results should be diversified in a multi-dimensional way, since queries are usually ambiguous at different levels and dimensions. Consequently, they propose a multi-dimensional SRD framework that exploits four data sources, including anchor texts, query logs, search result clusters and Web sites in order to mine query subtopics on multiple dimensions. Such subtopics are used to diversify documents by considering both their relevance to their novelty, following the *MMR* principle [22]. The authors evaluate their approach on TREC query sets in the context of diversity task and show the effectiveness of their method. In particular, they experimentally demonstrate that combining different resources yields to better improvement in terms of user intents' coverage. In this thesis, we also combine multiple resources which may help to improve the diversity of search results by maximizing the coverage of query aspects.

**** Hybrid Methods:***

While some proposed methods on explicit SRD approaches are based either on novelty or on coverage, other existing works rather combine both principles for better performance. Agrawal *et al.* [2] were interested in the problem of diversifying the search results for the case of ambiguous queries.

They show that, in general, diversification is an NP-hard problem. They propose a new approach for SRD which aims to select non-redundant documents that cover as much as possible the different query aspects. The idea is to diversify a document ranking list in light of a taxonomy of query intents. Given an ambiguous query, the first step is to determine a hierarchical taxonomy of the query in order to disambiguate it. For example, "Java" is an ambiguous query which could be interpreted to at least *programming language*, *coffee*, *dance*, and *island*. Under each of these interpretations, one can specify multiple aspects (*e.g. books*, *forums*, *source code*, for *programming language*). Query intents are represented by different categories from the ODP (Open Directory Project) [15]. Given the classification of both query and documents, the next step consists of *matching* the taxonomy of the tested query and each returned document. The more the two taxonomies are well matched, the more the corresponding document is considered to be relevant to that query. Based on that, the authors propose an intent-aware selection (*IA-Select*) algorithm and show that *IA-Select* can improve the ranking of the most relevant documents in the top results, while also promoting the diversity of the results.

Relatedly, Zheng *et al.* [131] propose to exploit a hierarchical classification of the concepts in order to mine query subtopics and infer their relations. Based on that, they propose a method for better diversifying search results, which breaks the limitation of existing SRD methods assuming that query subtopics are independent to each other.

Several existing SRD approaches are unable to ensure an effective coverage of the different query aspects. To solve this problem and better diversify the search results, Santos *et al.* [104, 105] transform the diversification problem to a query reformulation task. They introduce a new probabilistic framework called *xQuAD* (*explicit Query Aspect Diversification*), which can *explicitly* model the different query aspects. For this, several resources have been exploited, including Wikipedia to disambiguate the query and three major Web search engines to automatically extract the sub-queries. Diversity is estimated based on how relevant the document is to multiple aspects and by considering the relative importance of each aspect. A document is re-ranked depending on how it can cover the uncovered aspects. More precisely, starting with an initial document ranking, *xQuAD* aims to iteratively choose, for a given query Q, the document d having the highest score according to this formula:

$$(1-\lambda)\cdot p(d|Q)+\lambda\cdot p(d,\overline{S}|Q) \tag{2.19}$$

15. `http://www.dmoz.org`

where S is the set of documents already selected, $p(d|Q)$ is the probability of observing d given Q, and $p(d,\overline{S}|Q)$ is the probability of observing d but not the documents already selected in S. The parameter λ is used to control the trade-off between relevance and diversity. The diversification quality of *xQuAD* depends on both the relevance of each document with respect to the selected sub-queries, and the importance of each sub-query (subtopic). This latter is determined by estimating the size of the set of returned documents regarding to each subtopic: the higher the number of returned documents for the subtopic, the more important the corresponding subtopic. Instead of comparing each document with respect to each other (which is expensive in terms of complexity), the authors estimate the relevance of a document by its ability to cover multiple aspects of the query. This is one advantage of *xQuAD* compared to other existing SRD frameworks based on document-document similarity. They experimentally show that *xQuAD* outperforms several existing SRD approaches in terms of diversity. In this dissertation, we also compare our diversification methods with *xQuAD*.

Relatedly, in Santos *et al.* [106], the same authors observe that "... not all queries are equally ambiguous, and hence different queries could benefit from different diversification strategies". Therefore, their proposed approach aims to *selectively* diversify the Web search results by tailoring a diversification strategy to the ambiguity level of different queries. More precisely, given an unseen query, the authors use *xQuAD* [104, 105] and learn the trade-off between relevance and non-redundancy, based on optimal trade-offs observed for similar training queries. Santos *et al.* [106] use *KNN* algorithm to find the query neighbourhood based on a set of 953 features. These features are categorized into five groups: query concept identification, query type detection, query performance prediction, query log mining, and query topic classification. As a result, their approach effectively determines *when* to diversify the results for an unseen query, and also by how much.

More recently, Liang *et al.* [81] propose a new perspective of the diversification problem: Instead of re-ranking the set of initial retrieval results (as most of the state-of-the-art SRD approaches do), the authors propose to cast the diversification problem as a data fusion problem which consists of combining diversified ranked lists and inferring latent topics of the query from that merged list. At the end, the authors conclude that fusion helps diversification.

2.2.3 Applications of SRD in IR

The application of the SRD principle in different domains in order to solve practical problems stimulated a vast amount of research. It was first applied for system recommendation [133] based on the user profile and preferences. User profile is considered as a query and the goal is to return a number of queries to cover all interests. Ziegler *at al.* [133] show that trying to have a high coverage greatly improves the user satisfaction. However, one drawback of this work is that it does not consider the quality of a recommendation. This criterion is important since not all the recommendations are equally important for the user.

El-Arini *et al.* [52] propose an application of coverage-based SRD in the blogosphere. A set of messages is chosen so as to cover (almost) all the published news, which gives a complete summary to the user about the daily events. However, this approach does not distinguish the importance (or popularity) of a message.

SRD was applied on the question-answering problem. For instance, Clarke *et al.* [33] combine the novelty and the coverage principles when selecting the most relevant answers with respect to a question, where the question corresponds to a user query and the answer is the set of returned documents. The authors conclude that the user satisfaction increases if the returned documents are not redundant and cover different user intents. Haritsa [58] has also attempted to solve the same problem within a diversification metaphor, but using a machine learning technique. He was inspired from the KNN (K-Nearest Neighbor) algorithm to select similar answers with respect to a given question (query).

SRD was also applied to solve the problem of query abandonment [11, 28, 44] in the case of ambiguous queries where the probability of abandonment is generally high. The document selection criterion is based on its novelty compared to the other documents. The authors in [11, 28, 44] demonstrate that, by using the novelty principle, the probability of query abandonment decreases dramatically. However, the Bookstein's approach [11] used in these studies usually requires the explicit user feedback after each returned document, which makes this approach less practical because users are not willing to provide relevance feedback. The work of Chen and Karger [28] overcomes this drawback by proposing a function that returns a (relevant) document for all the users, which does not require any user feedback. The proposed function aims to maximize the probability of finding one

relevant document by assuming the non-relevance of the previously selected ones. However, this is not realistic because the intents differ from one user to another, and it is rare to find a document able to satisfy *all* the users simultaneously.

The application of SRD also includes query suggestion, such as the work of Strohmaier *et al.* [115] who introduce a new method seeking to better diversify query suggestions to match the user intents. In fact, query suggestion becomes a technique most commonly used by current search engines. It helps the user, seeking for the information, to reformulate her query so as to maximize her chance to find relevant documents with respect to her query [5, 6, 72, 86]. By using query logs (more precisely user clicks), they demonstrate in [115] that SRD can generate *intentional query suggestions*, which makes the user intents more explicit. Their system outperforms the Yahoo! Suggestion system. However, document relevance was not considered in their work: it merely tries to diversify query suggestions, without considering their relevance. The work of Ma *et al.* [86] overcomes this drawback and proposes a trade-off between relevance and diversity. Once the suggestions were collected from the search log data, they will be ranked using *Markov Random Walk*, based on their novelty. Nevertheless, it is arguable whether their method works for *rare queries*, in which case the corresponding search log data is generally poor. Recently, the work of Song *et al.* [111] mitigates this problem by proposing a more general framework for query suggestion, also inspired by the SRD principle, and that addresses the case of rare queries.

Other applications of SRD in several domains include image research [109] in order to maximize the coverage of query aspects, filtering systems [129] aiming to classify novel and redundant documents, text summarize by novelty [22] or by coverage [83] in which the idea is to mine, from a text, a set of phrases or sentences providing a complete and coherent summary of this text. These different applications of SRD in several domains highlight the importance of result diversification and its ability to solve different problems.

2.3 Diversified Query Expansion

All the diversification methods that we described before are applied at the document-level, *i.e.*, attempt to diversify the initial retrieval results. Instead of diversifying the results' list, a few recent methods diversify the query, by selecting candidate expansion terms that may cover the query aspects.

Vargas *et al.* [119] observe that the initial retrieval results from which documents are selected could be improved through query expansion. They adapt *xQuAD* [104, 105] to select diverse terms extracted from documents related to different query aspects in order to expand the query on different subtopics. The subtopics are extracted based on clusters of returned documents that group documents sharing the same aspect underlying the query. This could ensure a better balance between aspects in the final retrieval results, helping solve the problem of dominating subtopics. This work is very close to ours: both try to diversify the expansion of a query. However, an important difference is that in the method of Vargas *et al.* [119], expansion terms are extracted only from the retrieval results of the initial query, which may suffer from poor coverage of the different aspects of the query. If an aspect was not covered within the initial query, such aspect will never be covered. This may especially occur when we consider difficult queries [3] where the set of documents feedback brings a lot of noise, rather than useful information. Our approach does not rely solely on the set of returned documents with respect to the query; instead, we believe that considering an external resource and/or combining different resources could potentially bring better improvements.

Dang and Croft [42] extend *PM-2* [43] (which greedily determines the aspect that best maintains the overall proportionality of the aspects covered and then selects the best document on that aspect) by incorporating query expansion using topic terms extracted with an algorithm for document summarization from feedback documents, hoping that the expanded query can cover more query aspects. They show that there is no need to explicitly determine the whole query subtopics (which is a difficult task in itself), and that single expansion terms could be enough to represent these subtopics. In this dissertation, we will compare our diversification methods with *PM-2* based on QE, simply because it has been demonstrated to be effective on the ClueWeb collection, which we also use to conduct our experiments.

He *et al.* [61] propose the *Multi-Search Subtopics* (*MSS*) framework which combines click logs, anchor text and Web n-grams to generate related terms for QE, for the purpose of improving the diversity of search results. These terms are organized into a graph on which random walks are performed to compute the similarities between suggested terms, which are used to estimate the similarity between subtopics extracted from different heterogeneous resources. Note that, their approach selects expansion terms according to their similarity to the query terms, and does not consider the possible redundancy among expansion terms, as we do in this thesis. Since this approach is similar to the DQE

method that we propose in this dissertation (both two approaches use QE in the context of SRD where expansion terms are selected from multiple heterogeneous resources), we also compare our method with that of He *et al.* [61].

These DQE methods, despite their novelty, have been shown to be effective and provide promising results over existing state-of-the-art SRD methods which diversify the initial retrieval results. In fact, most of the existing SRD methods rerank the initial results' list with respect to an original (short) query, which generally consists of few terms (2 or 3 words). However, a few words could not be enough to fully describe all the user intents (query subtopics). This may explain why initial retrieval results is enable to cover all the query subtopics, thus negatively reflected to the quality of the SRD methods.

Several studies show that query expansion may improve the quality of the retrieval results. However, when query expansion is performed for the purpose of SRD, it has a distinctive feature from general query expansion: the goal is not only to cover more relevant documents, but also to cover more diversified documents. Therefore, the diversity of the expansion terms should be explicitly taken into account. This *enforces* that some aspects have the chance of being covered since the first retrieval results, which may help solve the problem of dominating subtopics.

In this dissertation, we propose to go further in this direction. We believe that DQE may replace the (standard) SRD methods which are applied at the document level, and consequently, we introduce a new method for DQE which greedily selects, for each query, a diversified set of expansion terms which are good representative of the query aspects. In our study, we exploit query aspects but without the need of manually determining them in the form of subtopics, as most of explicit SRD approaches do. Besides, in order to ensure that the selected expansion terms have a good coverage of the query aspects, and that are not limited to the initial retrieval results, we use different resources (including ConceptNet, Wikipedia, query logs and feedback documents) from which we extract our candidate expansion terms.

2.4 Using External Resources in IR

The utilization of external resources has attracted much attention by IR researchers. During the last two years, TREC organizers have introduced a new track called*Federated Web Search* [16] aiming to querying multiple search engines (*i.e.*, resources) simultaneously and combine their results into one single list. The track includes three tasks: Resource selection, results merging and vertical selection. The results of groups participating to this track clearly show the advantages of integrating multiple resources which may help improving the quality of retrieval results. This strongly motivates us to use multiple resources and combine them for the purpose of SRD.

The idea of exploiting different resources has been successfully applied in different fields in information retrieval (*e.g.*, to collect good expansion terms for QE also known as query reformulation). While some approaches rely on a single resource (*e.g.*, ConceptNet [65, 74], query logs [39, 40], PRF [20, 85, 122], Wikipedia [80], anchor text [41], to name just a few), other methods rather combine multiple resources (*e.g.*, [8, 47, 48, 51, 61]).

For instance, some studies attempted to leverage ConceptNet for different tasks, such as word-sense disambiguation [28] and image retrieval [63, 116–118]. In particular, ConceptNet has been exploited in QE. Hsu *et al.* [64, 65] compared the effectiveness of ConceptNet and WordNet [17] for QE using Spreading Activation and existing machine learning techniques. They conclude that WordNet can select highly discriminative terms while ConceptNet ensures a higher diversity. This result shows that ConceptNet could be appropriate for diversifying search results, which motivate us to use this resource for the purpose of better diversifying the results. More recently, Kotov and Zhai [74] proposed methods that leverage ConceptNet for QE, and demonstrate that ConceptNet is an effective resource to improve search results when pseudo-relevance feedback becomes ineffective, which is usually the case for difficult queries. The authors showed the richness of ConceptNet as a common-sense knowledge base, compared to other lexico-semantic resources such as WordNet and Wikipedia. It is then possible to infer complex information between the concepts from ConceptNet in order to select good terms for expansion. The authors proposed several heuristics and learning-based methods to automatically select effective terms from ConceptNet for expansion. However, no previous research tried to diversify expansion terms using ConceptNet as we propose in this dissertation.

16. `http://trec.nist.gov/data/federated.html`
17. `http://wordnet.princeton.edu`

Instead of using a single resource, one can benefit from the combination of several resources together motivated by the fact that expansion terms selected from a single resource may not be enough to ensure a good coverage of the query topics and that combining multiple resources may yield to a better coverage. For instance, Diaz and Metzler [48] present a mixture of relevance models, in which they found that combining multiple external resources improves the relevance of the results. Bendersky *et al.* [8] collect expansion terms (concepts) from news-wire and Web corpora. These resources are then used to compute the importance (weight) of each concept, and to perform PRF. They show that combining multiple resources is usually more effective than considering any single resource, and that such combination yields improved diversity of search results. Recently, Deveaud *et al.* [47] observe that the more we use several resources, the more likely we can improve the topical representation of the user information need.

All these studies suggest the utilization of multiple resources when possible. Different from these studies, we take into account diversity. For SRD, He *et al.* [61] select candidate expansion terms from several heterogeneous external resources (namely Web n-grams, anchor text and click logs). Selected expansion terms may correspond to different query subtopics. They experimentally show that by combining these resources, better topic models are formed, and such combination may alleviate the lack of coverage. Dou *et al.* [51] also propose to combine multiple resources including anchor texts, query logs, search result clusters and Web sites to mine query subtopics on multiple dimensions. Such subtopics are used to better diversify the search results. They show that combining multiple resources is beneficial compared to the use of any single resource, and that these resources are complementary in the sense that they provide a better coverage of the user intents. Hong and Si [62] use different external sources in the context of Federated Web Search, and combine them to better diversify the document ranking (a better coverage of query aspects). The authors show the effectiveness of their proposed methods by conducting extensive experiments on the federated search testbed of the ClueWeb dataset.

Compared to these studies, our work has three significant differences. First, in our study, these resources are used to directly generate diversified candidate expansion terms. Second, *MMR* principle is used to remove the redundancy of selected expansion terms and also to cover as many aspects as possible of the query. In particular, we will show in chapter 3 of this dissertation that integrating multiple resources can improve the diversity of search results and the coverage of the query aspects. During our participation to the NTCIR IMine task, we combined five different resources (we consider

feedback documents, Wikipedia, ConceptNet, query logs and query suggestions provided from Bing, Google and Yahoo! search engines) and observe that the more resources we consider, the more aspects of the query we can cover. A third significant difference, is that in the previous studies, all the resources are weighted uniformly. To the best of our knowledge, no previous study has proposed to properly weight different resources for the purpose of SRD, as we propose in chapter 4 of this dissertation. More precisely, we introduce a new query-dependent resource weighting method for the purpose of DQE, and we show experimentally that such a proper weighting can lead to significant gains in retrieval effectiveness.

2.5 Embedding

Although several approaches have been proposed to diversify the expansion terms of a query (such as [61, 119]), no explicit representation of query aspects has been used. Therefore, term dissimilarity is measured at the surface level *i.e.*, using a word-based representation. This gives rise to the problem of selecting multiple expansion terms relating to the same query aspect - two terms may be considered different at the surface level, yet they are related to the same query aspect. For example, for a query on "Java", the word *program* and *algorithm* are different, but are related to the same aspect of *programming language*.

To ensure a good coverage of the query aspects, one should adopt an explicit SRD method. As stated before in section 2.2.2, explicit SRD methods first automatically extract the query aspects and then diversify search results according to these aspects. However, the quality of these methods is dependent on that of the extracted aspects: the better aspects of the query we extract, the better we can diversify search results. Ideally, these methods perform well by assuming that the query aspects are already available and one can use the manually identified query subtopics. However, such manual query subtopics are generally not provided in practice.

Our study is an extension to these studies by extracting a set of aspects for a query. However, unlike existing explicit SRD methods, in this dissertation, we utilize the query aspects in the context of DQE but without assuming that they already exist. There has been studies on extracting query aspects from feedback documents (such as the work of Vargas *et al.* [119]). However, to our knowledge, no study has used query aspects for DQE, which is what our dissertation concerns. More precisely, in

our work, we try to determine the latent aspects underlying a query. This is related to the work on word embedding - an abstract representation created to represent latent semantics. With embedding, any object (*e.g.*, a term, an aspect) can be mapped to a vector in the embedding space, thus has a latent semantic representation. In our case, each expansion term will be mapped to a vector in the aspect embedding space, in which each dimension is assumed to relate to a query aspect. Term dissimilarity is then measured at the aspect level rather than the surface term level, which may help to determine deeper and semantic relations between the expansion terms. In the example above, *program* and *algorithm* will be considered similar with respect to the aspect they cover, while *coffee* will be different from them. An explicit representation of query aspects may have an important advantage for DQE: the expansion would be able to better cover all the aspects of the query. Our work represents a further development in DQE based first on the term level and then on an explicit representation of query aspects at the latent semantic level.

At first glance, our work is similar to that of He *et al.* [61] which combine click logs, anchor text and Web n-grams to generate related terms for QE, for the purpose of improving the diversity of search results: both define a global similarity function for expansion terms from multiple heterogeneous resources. However, He *et al.* [61] estimates term similarity directly at term level, without defining aspects as we do in this thesis. As we will show in our experiments, a DQE approach using aspects leads to better search results than without using aspects.

In our study, we use the idea of embedding in order to determine the latent aspects underlying a query, based on the expansion terms that have been selected for that query. It is worth noting that the idea of embedding has been successfully exploited for a wide range of tasks. For example, Koren *et al.* [73] use matrix factorization technologies to map users and movies to the same vector space, and win the Netflix Prize competition. Their proposed model provides personalized recommendations for each individual user and movie based on the user preferences and other demographic data. Huang *et al.* [67] exploit a multi-layer neural network to learn vector representations for the document using click-through data. The authors use a standard BOW (bag-of-words) representation of both the query and the document and match each raw term vector to its Latent semantic vector space. Their proposed framework namely, DSSM (Deep Structured Semantic models) is reported to give superior IR performance compared to other latent semantic models for the Web document ranking task. Mikolov *et al.* [90] use embedding to efficiently learn high-quality distributed vector representations, aiming to

capture a large number of precise syntactic and semantic word relationships, such as phrases. Word embeddings are learnt from free text, using *one-to-one* relationships between entities of different types, such as *capital of* relation between countries and cities. In Mikolov *et al.* [90], the proposed objective function drives the model to learn similar embedding vectors for semantically related words (*e.g.*, synonym words tend to appear in similar contexts). In the same context, Mikolov *et al.* [89] learn vectors representations of words from huge data sets (their model is trained using 1.6 billion words) in order to preserve some syntactic and semantic regularities and show that using word vector embeddings leads to promising results in practice, such as in machine translation tasks.

The embedding function can be learnt based on deep neural networks [67], probabilistic topical models [10, 61], matrix factorization [73, 121], quantum computing [112], trace norm regularization [84], etc.

Recently, our work [84] was the first attempt towards using embedding in the context of diversified query expansion. We introduced a new method called *compact aspect embedding* which is an instance of DQE, and consists of three steps. Given a query, we first generate expansion terms using an external resource, namely query logs. Then, we map expansion terms into a low-rank vector space by solving the following optimization problem:

$$\begin{aligned} &\min \frac{1}{2}||\boldsymbol{E}^T\boldsymbol{E}-\boldsymbol{S}||_F^2+\eta||\boldsymbol{E}||_* \\ &\text{subject to: } ||\vec{e}||_F^2=1, \forall e\in E. \end{aligned} \tag{2.20}$$

where q denotes an original query; E, the expansion terms related to q; $e \in E$, an expansion term; $\vec{e}$, the column vector corresponding to expansion term e; $\boldsymbol{E}$, the matrix with each column representing an expansion term vector, which also denotes the vector space to be to learnt; $||\cdot||_F$, the Frobenius-norm of a matrix (respectively a vector), defined as the sum of the absolute squares of all elements of the matrix (respectively a vector); $||\cdot||_*$, the trace norm of a matrix, defined as the sum of the singular values of the matrix; $\boldsymbol{E}^T$, the transpose of matrix $\boldsymbol{E}$; $\boldsymbol{S}=(s_{ij})$, the similarity matrix, where s_{ij} denotes the similarity between two expansion terms e_i and e_j.

With the learnt vector space, we select an eigenvector (aspect vector) for each non-zero eigenvalue to represent an aspect of the query in the vector space. Accordingly, we use the absolute value of the eigenvalue (aspect weight) to model the associations strength of the corresponding aspect with

the query. To ensure that the expansion terms selected are relevant to the query and cover all the aspects of the query, we also design the following greedy selection strategy: we first order the aspect vectors in descent order by their weights. Afterwards, for each aspect vector, several expansion terms are selected which may cover this aspect while not being redundant with already selected expansion terms. In addition, we make the number of selected expansion terms for an aspect proportional to the weight of the aspect. We have extensively evaluated our compact aspect embedding approach on TREC diversification data sets, and show that it significantly outperforms the state-of-the-art SRD approaches and that the explicit modeling of query aspects brings significant gains.

This work [84] is very similar to the method that we present in chapter 5: Both use embedding in the context of diversified query expansion to explicitly learn the aspects of a query. However, there are two main differences between these two methods. The method that we describe in chapter 5 is more flexible and is regulated by multiple resources, each of which is weighted during the process of learning the query aspects which may help suggesting expansion terms from a better quality. However, the method presented in [84] is defined for one single resource, thus making the query's aspects coverage limited to that of the resource. A second major difference between these two methods is in the way used for learning aspects: In [84], we exploit trace norm regularization to learn a low rank vector space for the query, with each eigenvector of the learnt vector space representing an aspect, and the absolute value of its corresponding eigenvalue representing the association strength of that aspect to the query. In chapter 5, we use an embedding function that maps query expansion terms to aspect vectors for a given query. The embedding function is discriminatively trained so that two expansion terms are pushed close in the aspect vector space if they are similar according to some resource. We also formulate the learning procedure as an optimization problem similar to matrix factorization [73]. In addition, observing that an expansion term is usually related to one or a few query aspects, we also use the sparsity constraint in our model. Since a query often has a limited number of different aspects, the learnt aspect vector often has only a few dimensions, making our embedding computationally efficient. This is different from most of the embedding studies, which often requires a large number of dimensions to capture the great variances among a large number of objects *E.g.*, [67] uses 30,621 dimension vectors to represent a vocabulary of 500,000 words. In chapter 5, we will compare our aspect-level DQE method with our compact aspect embedding method that we have already introduced in [84], and we will experimentally show the usefulness of our latent

aspect DQE method compared to the compact aspect embedding method.

Finally, it is worth noting that some existing studies on SRD show the usefulness of weighting the query aspects in explicit SRD. For instance, Santos *et al.* [105] estimate the sub-query importance in order to promote aspects of interest to the user, and show that weighting query aspects improves both relevance and diversity of search results. In the same context, Ozdemiray and Altingovde [94] use post-retrieval query performance predictors to estimate aspects' weights based on the retrieval effectiveness on the document set. They experimentally show that weighting query aspects improves the state-of-the-art SRD approaches. In our work, selected expansion terms are also weighted according to their relevance to the original query and also their novelty compared to the expansion terms already selected, for the same query.

2.6 Conclusion

In this chapter, we described test collection and information sources which we use along this thesis to evaluate our approaches and compare them with existing methods. Thereafter, we reviewed the studies which are related to our work. In particular, we first described diversification methods which we categorized on either explicit or implicit, according on how they represent the query aspects, and on coverage-based SRD, novelty-based SRD and hybrid SRD according to which criteria is used to diversify the search results. Afterwards, we reviewed recent methods which diversify the expansion terms of the query instead of diversifying the retrieval results. Then, we described some studies which use different external resources to solve common problems in IR. Finally, since our work is also closely related to embedding and to aspect representation, we also reviewed some approaches about embedding, and clarify their connection with our work.

The following three chapters will describe our work addressing different problems in DQE. In Chapter 3, we describe an approach to DQE using external resources. The main content of the chapter corresponds to the following two published papers (with some modifications): [13, 14]. In Chapter 4, we tackle the problem of resource weighting. The chapter corresponds to the following paper: [16]. In Chapter 5, we describe our approach based on aspect embedding. The content appeared in the following paper: [84].

Chapter 3

Diversified Query Expansion using External Resources

3.1 Introduction

In its basic setting, Search result diversification (SRD) aims to select diverse documents from the initial search results. A prerequisite is that the set of retrieved results corresponding to the original query contains diverse documents, which is not always the case for different queries. The final ranking list is much dependent on the initial retrieval results, which should have a good coverage of the different aspects of the query. Despite some attempts [119] to use query expansion (QE) and pseudo-relevance feedback (PRF), these methods are limited because they are still much dependent on the retrieval results with the initial query. In the case where some aspects are not well covered in the initial retrieval results, this method will be unable to cover them well. For a difficult query in particular, the retrieval results are mostly irrelevant[3]. PRF will bring more noise rather than useful terms into the query.

In this chapter, we first propose a new approach to SRD by diversifying the query (Bouchoucha *et al*. [13]). To ensure that QE will be less dependent on the initial retrieval results, expansion terms are selected from an external resource, namely ConceptNet, which is presently the largest commonsense knowledge base. In particular, we perform a diversified query expansion (DQE) following a similar principle to MMR (Maximal Marginal Relevance) [22].

It is worth noting that when query expansion is performed for the purpose of SRD, it has a distinctive feature from general query expansion: the goal is not only to cover more relevant documents, but also to cover more diversified documents. Therefore, the diversity of the expansion terms should be explicitly taken into account as we do in this chapter. DQE represents recent efforts in explicit SRD, with the goal of directly generating a set of diversified expansion terms.

Since the coverage of the query aspects is limited by that of the resource, we propose in the second part of this chapter, the use of multiple resources (in addition to ConceptNet, we consider query logs, Wikipedia and document feedback), thus yielding to a more general and effective framework for diversified query expansion (Bouchoucha *et al*. [14]).

3.2 DQE using a Single Resource: ConceptNet

In this section, we first briefly present ConceptNet to explain our motivation of using this resource. Afterwards, we motivate our proposed approach by an example in TREC, and then present our method in detail.

3.2.1 Motivation Example

To analyze the behaviour of standard QE techniques in term of diversity, let us consider the query #8 from the TREC 2009 Web track [34]: Q = "*appraisals*". This query is ambiguous and has four different subtopics identified by TREC organizers [1] (see Table 3.I).

Subtopic	*Description*
1	What companies can give an appraisal of my home's value?
2	I'm looking for companies that appraise jewelry.
3	Find examples of employee performance appraisals.
4	I'm looking for web sites that do antique appraisals.

Table 3.I: List of the TREC subtopics for the query Q = "*appraisals*".

Q is a difficult query because only a few relevant documents can be retrieved using a traditional model (MAP = 0.0058 with KL retrieval method on ClueWeb09B dataset). Based on document feedback, it is difficult to extract relevant terms for expansion.

Alternatively, one can think to use an external resource, from which extracting (good) candidate expansion terms. ConceptNet is known to be a good resource and the (semantic) relations between concepts it contains reflect well the understanding of human beings in different areas. Please refer to Section 2.1.2 which briefly describes ConceptNet and its advantage. In our study, we leverage ConceptNet in order to make similar complex inferences to identify the effective expansion terms that are *broadly* related to a given query. ConceptNet could be useful when the initially retrieved results are of poor quality and, consequently, cannot be used as a source of (good) expansion terms.

Spreading Activation (denoted *SA* hereafter) [64, 65] has been shown as an effective QE method with ConceptNet. The traditional QE identifies a set of expansion terms that are the most related to the original query terms (or have the highest activation scores). More specifically, we first construct

1. `http://trec.nist.gov/data/web/09/wt09.topics.full.xml`

a graph containing the nodes that are (semantically) related to the query's terms. The activation score ($ActivS(i)$) of a node i in the graph is calculated using Equation3.1 as follows [64]:

$$ActivS(i) = C_{dd} \cdot \sum_{j \in Neighbor(i)} (ActivS(j) \cdot W(i,j)) \tag{3.1}$$

where $C_{dd} \leq 1$ is a constant called *distance discount* or *decay factor* (following Hsu *et al.* [64], we set $C_{dd} = 0.5$ in our experiments), $Neighbor(i)$ represents the nodes connected to node i, $ActivS(j)$ is the activation score of node j and $W(i,j)$ is the weight of the link from node i to node j. To compute $W(i,j)$, we follow the work of Kotov and Zhai [74] who design an empirical procedure to calculate the weights between the concepts (*i.e.* nodes) in the graph of ConceptNet. At the first step, each node has an initial activation score (which is experimentally set to 1.0). Table 3.II shows the top 10 expansion terms determined in this way.

We denote by Q_1 the resulting expanded query. We manually tag each expansion term with their corresponding subtopic numbers listed in Table 3.I. The character "-" means that the corresponding expansion term does not correspond to any specific subtopic of Q, or may correspond to all possible subtopics of Q. For example, both expansion terms *jewelry* and *diamond* correspond to the second subtopic of Q, but expansion terms *money* or *expert* do not correspond to any subtopic of Q, as defined by TREC assessors.

	appraisals	*appraise*	*worth*	*estimate*	*expert*	*money*
	-	-	-	-	-	-
Q_1	*jewelry*	*examine*	*evaluation*	*diamond*		
	2	-	-	2		

Table 3.II: List of the expansion terms produced for the query Q using *SA*, and their corresponding subtopic numbers.

From Table 3.II, we observe that the expansion terms only correspond to one aspect (aspect 2) and they do not promote the diversity of search results. This result can be explained by the fact that the query is expanded *globally* in a unique way, leading to the expansion of the dominant aspect (meaning) of the query. Using such an expanded query, one may expect that the search results are not much diversified. In the next section, we propose a new method that aims to select diverse expansion terms ensuring a good coverage of the different query aspects.

3.2.2 Diversifying Expansion Terms using ConceptNet

**** Principle:***

Diversifying query expansion has a very similar goal to result diversification. On the one hand, we want the expansion terms to be closely related to the initial query. On the other hand, we want the expansion terms to be diverse, or non-redundant. A similar approach to MMR can naturally be used.

MMR (Maximal Marginal Relevance) [22] is a method of SRD trying to select documents that are dissimilar from the ones already selected. The following formula is used:

$$MMR(D_i) = \lambda \cdot rel(D_i, Q) - (1 - \lambda) \cdot \max_{D_j \in S} sim(D_i, D_j) \tag{3.2}$$

where Q denotes a query, D_i is a candidate document from a collection, and S is the set of documents already selected so far. The parameter λ controls the trade-off between relevance and novelty (*i.e.*, non-redundancy) which is often set at 0.5. $rel(.,.)$ and $sim(.,.)$ are two functions that determine respectively the relevance score of the candidate document to the query and its similarity to a selected document. In each step, *MMR* selects the document with the highest *MMR* score.

We adapt the *MMR* principle for selecting expansion terms from an external resource, that is ConceptNet in our case. Our method is called *MMRE* (*MMR*-based Expansion).

**** The MMRE method:***

Given a query $Q = q_1 q_2 ... q_n$ formed by n terms (after removing stopwords), we iteratively select the top N expansion terms having the highest *MMRE* scores. The *MMRE* score is computed as follows:

$$MMRE(e_i, Q) = \lambda \cdot sim(e_i, Q) - (1 - \lambda) \cdot \max_{e_j \in S} sim(e_i, e_j) \tag{3.3}$$

where e_i is a candidate expansion term or a concept in ConceptNet (*i.e.* a node) from ψ: the set of concepts that are related to Q, S is the set of terms already selected, and Q is the query under consideration. $sim(e_i, Q)$ determines the similarity between e_i and Q, and $sim(e_i, e_j)$ determines the similarity between two expansion terms e_i and e_j.

We use the following *Jaccard* similarity function $sim(e_i, e_j)$:

$$sim(e_i, e_j) = \frac{|N_{e_i} \cap N_{e_j}|}{|N_{e_i} \cup N_{e_j}|} \tag{3.4}$$

where N_{e_i} (resp. N_{e_j}) is the set of nodes from the graph of ConceptNet that are related to the node of the concept e_i (resp. e_j). In other words, we consider related nodes those that are connected together in the graph of ConceptNet either directly or through other nodes. For example, in Figure 2.2, we consider that the two nodes *wake up in morning* and *eat breakfast* directly related since they are directly connected in the graph of ConceptNet based on the relation *PrerequisiteOf*. Nodes *wake up in morning* and *full stomach* are indirectly related since they are connected through an intermediate node, that is *eat breakfast*. The more common node e_i and e_j share, the more they are considered to be (semantically) similar.

$sim(e_i, Q)$ in Equation 3.3 (where Q is considered as a bag-of-words) could be defined in a similar way by replacing N_{e_i} in the above formula by N_Q, which is the set of nodes that are simultaneously connected to all terms in Q. However, it is often the case that no node in ConceptNet is connected to *all* the terms in Q. We therefore define a modified $sim(e_i, Q)$ that considers the proportion of the terms in Q that are related to nodes in ConceptNet as follows:

$$sim(e_i, Q) = max_q\{\frac{|N_{e_i} \cap N_q|}{|N_{e_i} \cup N_q|} \cdot \frac{|q|}{|Q|}\} \tag{3.5}$$

where q is a subset of Q and $|q|$ is its size.
The idea is to allow a term (or a concept) e_i to *match* part of the query Q, but its similarity is proportional to the number of terms in Q it matches. Our algorithm (see Figure 3.1) uses any of the subsets of terms in Q as a possible candidate q.

Notice that ConceptNet contains a weight between each pair of nodes that reflects the *strength* of relationship between them. These weights are between -1 and 1. As mentioned in line 6 of the *MMRE* algorithm, we only keep the concepts having positive weights, since they correspond to true assertions.

Another parameter that we use in the algorithm of *MMRE* is the radius (ρ), which refers to the depth (*i.e.* the number of edges) that we consider for the construction of the graph. $\rho = 1$ means that we only consider the directly connected nodes, $\rho = 2$ means that we consider nodes related through two edges, etc. In section 3.5, we will test *MMRE* with $\rho = 1$, $\rho = 2$ and $\rho = 3$.

MMRE (Q, n, r, λ, N)

1. Let E_i be the set of possible subsets of i terms (or concepts) of Q from n.
2. Initialize $\phi \leftarrow \emptyset$, $S \leftarrow \emptyset$
3. **while** ($|S| \leq N$)
4. **for** i **from** 1 **to** n **do**
5. **for each** subset q **from** E_i **do**
6. $\psi \leftarrow \emptyset$
7. Find, from ConceptNet, the terms that are connected to the terms of q in a radius ρ, and only keep the terms with positive weights. Add these terms to ψ.
8. **for each** term e **from** ψ **do**
9. $MMRE(e,Q) = \lambda \cdot sim(e,Q) - (1-\lambda) \cdot \max_{e' \in S} sim(e,e')$
10. **end for**
11. **end for**
12. **end for**
13. $e^* = argmax_{e'} MMRE(e',Q)$
14. $S = S \cup \{e^*\}$
15. **end while**
16. Return S.

Figure 3.1: The *MMRE* algorithm.

The result of applying *MMRE* (with $\rho = 2$ and $\lambda = 0.6$) to the example query Q = "*appraisals*" given earlier in section 3.2.1, can be found in Table 3.III. We denote by Q_2 the resulting expanded query. From Table 3.III, we can observe that *MMRE* performs well for the selection of expansion terms related to more query aspects than a traditional expansion approach, despite the fact that some subtopic (the subtopic 3) is still missing from the top 10 selected expansion terms. One can expect that the retrieval results with this expanded query is more diversified than with the one using traditional query expansion.

	appraisals	*value*	*antique*	*appraise*	*jewelry*	*company*
	-	-	4	-	2	1, 2
Q_2	*home*	*evaluation*	*buy*	*web*		
	1	-	-	4		

Table 3.III: List of the expansion terms produced for the query Q using *MMRE*, and their corresponding subtopic numbers.

3.3 Integrating Multiple Resources for DQE

In the previous section, we introduced a new method (*MMRE*) which selects diversified expansion terms from a single external resource, namely ConceptNet. The coverage is thus limited to that of the resource. To alleviate this issue, we propose in this section, to extend *MMRE* for multiple resources, thus yielding to a more general DQE framework. More specifically, given an original query, our framework first automatically generates a list of diversified expansion terms from each resource, and then combines the retrieved documents for all the expanded queries following the Maximal Marginal Relevance principle [22]. In this section, we first motivate our proposed framework and then present the method in detail.

3.3.1 Motivation

As we have seen in the previous example, one single resource (*e.g.* ConceptNet, documents feedback) usually cannot ensure a high coverage of the query aspects, and for different queries. Our approach described in this section is largely motivated by the following observation: there are a large number of queries for which ConceptNet cannot yield good performance but some other resources can suggest good terms. For example, "*defender*", the #20 query from the TREC 2009 Web track [34], is such an example. This query is ambiguous and has six different subtopics [2], as described in Table 3.IV.

In our experiments, traditional IR models for this query return no relevant documents. In other words, none of the retrieval results of this query is relevant according to the relevance judgements that are provided by TREC assessors. Therefore Pseudo-Relevance Feedback does not help. ConceptNet returns results covering subtopic 2, 3 and 6, while Wikipedia and query logs provide documents covering subtopic 1, 2, 3, 4 and 1, 2, 4, 5, respectively. By integrating all these sources, we obtain a list of documents covering all the subtopics. This example motivates the use of multiple resources for query expansion.

2. `http://trec.nist.gov/data/web/09/wt09.topics.full.xml`

Subtopic	*Description*
1	I'm looking for the homepage of Windows Defender, an anti-spyware program
2	Find information on the Land Rover Defender sport-utility vehicle.
3	I want to go to the homepage for Defender Marine Supplies.
4	I'm looking for information on Defender, an arcade game by Williams. Is it possible to play it online?
5	I'd like to find user reports about Windows Defender, particularly problems with the software.
6	Take me to the homepage for the Chicago Defender newspaper.

Table 3.IV: List of the TREC subtopics for the query "*defender*".

3.3.2 Proposed Framework

Our proposed framework consists of two layers. In the first layer, we generate for each original query, a diversified set of expansion terms using each resource. In the second layer, we apply a diversified document result fusing. In the remainder of this section, we describe in detail each layer.

*** First Layer:**

The first layer integrates a set of resources, denoted by R, to generate diversified queries as we already explained in Section 3.2.2. Given an original query Q, it iteratively generates a good expansion term e^* for each resource $r \in R$, which is both similar to the initial query Q and dissimilar to the expansion terms already selected:

$$e^* = argmax_{e \in E_{r,Q}}(\lambda_r \cdot sim_r(e,Q) - (1-\lambda_r) \cdot max_{e_i \in S_{r,Q}} sim_r(e,e_i)) \tag{3.6}$$

Here, $E_{r,Q}$ and $S_{r,Q}$ represent the set of candidate expansion terms and the set of selected terms for resource r, respectively; the parameter λ_r (in [0,1]) controls the trade-off between relevance and redundancy of the selected term; $sim_r(e,e_i)$ returns the similarity score of two candidate expansion terms e and e_i for resource r; $sim_r(e,Q)$ is the similarity score between expansion term e and the original query Q, based on resource r which is computed using Formula 3.7, where q is a subset of Q

and $|q|$ denotes the number of words of q.

$$sim_r(e,Q) = max_{q \in Q} sim_r(e,q) \cdot \frac{|q|}{|Q|} \tag{3.7}$$

Once expansion term e^* is selected, it is removed from $E_{r,Q}$ and appended to $S_{r,Q}$. With the parameter λ_r, initial term candidates $E_{r,Q}$, and the term pair similarity function $sim_r(e,e_i)$, which depend on the particular resource, Formula 3.6 becomes a generalized version of Maximal Marginal Relevance-based Expansion (*MMRE*) that we proposed before in Section 3.2.2, and by instantiating λ_r, $E_{r,Q}$ and $sim_r(e,e_i)$, our framework can integrate any resource.
Now, we first describe how expansion terms are generated from each resource, then, we explain how the similarity between a pair of expansion terms is computed across different resources.

Given a resource r and a query Q, we assume there exists a corresponding function $gen_r(Q)$ to produce a set of candidate expansion terms. In this work, we investigate four typical resources available to us: ConceptNet, Wikipedia, query logs, and pseudo feedback documents, hereafter denoted by C, W, L and F, respectively. The implementation of $gen_r(Q)$ often depends on the resource.

For ConceptNet ($r = C$), we use the same approach that we already described in Section 3.2.2, by choosing the concepts that are connected to the terms of Q (we test for different values of the radius ρ). We define $gen_C(Q)$ as the set of terms (nodes) in the graph of ConceptNet that match the query terms or a part of the query terms.

For Wikipedia ($r = W$), the candidate expansion terms are the terms in the anchor texts (outlinks) and the category names of the Wikipedia pages that match Q (or any part of Q). In cases where no Wikipedia pages match Q (or any part of Q), we use Explicit Semantic Analysis (ESA) [53] to collect semantically related Wikipedia pages, on which we perform the extraction. ESA assumes that each Wikipedia article represents a distinct semantic unit. Two terms are considered to be similar if they correspond to similar Wikipedia articles.

For query logs ($r = L$), expansion terms are extracted from the queries that share the same click-through data with Q, and the reformulated queries of Q that appear in a user session within a time window of 30 minutes, as suggested by Radlinski and Dumais [96].

Finally, for feedback documents ($r = F$), we consider top 50 returned results as relevant documents, and select terms that co-occur often with the query terms (within text windows of size 15).

For the remainder of this section, we use e_i and e_j to design two expansion terms that are determined by using resource r.

Firstly, for ConceptNet, given e_i and e_j, $sim_C(e_i, e_j)$ is computed using the same Equation 3.4 described before:

$$sim_C(e_i, e_j) = \frac{|gen_C(e_i) \cap gen_C(e_j)|}{|gen_C(e_i) \cup gen_C(e_j)|} \tag{3.8}$$

These similarity functions are defined in different ways on other resources, but following a similar principle: a graph is constructed for a given query in which two terms are connected if they are related in Wikipedia (*i.e.* they share at least one anchor text or one category), co-occur in the same search session in query logs, or appear in a feedback document for the query. The similarity between terms is estimated in a similar way to Formula 3.8, *i.e.*, by computing the Jaccard coefficient. We now provide the details about these similarity functions regarding to each resource.

For Wikipedia, to compute the similarity between two expansion terms, we first run ESA [53] to obtain a set of semantically related words for each expansion term with each related word being represented as a vector. In other words, given an expansion term e, we collect the Wikipedia pages (*i.e.* vectors) in which term e appears. Then we apply Formula 3.9:

$$sim_W(e_i, e_j) = \frac{1}{|W_i||W_j|} \sum_{w_i \in W_i, w_j \in W_j} sim(w_i, w_j) \tag{3.9}$$

where W_i (resp. W_j) is the set of semantically related words of e_i (resp. e_j), and $sim(w_i, w_j)$ is the cosine similarity between vectors w_i and w_j.

The log data that we consider in this work contains several useful information, such as the user sessions (each session is identified by an ID), the time-stamp that the user has spend in her session, the query string, the number of results on results page, as well as the click-through data (URLs). For query logs, the similarity between expansion terms e_i and e_j is proportional to the number of queries in the logs that include both e_i and e_j:

$$sim_L(e_i, e_j) = \frac{|Q_i \cap Q_j|}{|Q_i \cup Q_j|} \tag{3.10}$$

where $Q_i = \{e | e \in gen_L(Q), e_i \in Q\}$ (resp. $Q_j = \{e | e \in gen_L(Q), e_j \in Q\}$) is the subset of $gen_L(Q)$ that includes e_i (resp. e_j).

Finally, for feedback documents, the similarity between two expansion terms is calculated based on their co-occurrences in a text window across all the feedback documents:

$$sim_F(e_i, e_j) = \frac{|D_i \cap D_j|}{|D_i \cup D_j|} \tag{3.11}$$

where D_i (resp. D_j) is the set of text windows (of size 15) containing e_i (resp. e_j).
Notice that all the above similarity measures are normalized in similar ways. Therefore, they can be combined in a straightforward way.

Finally, we select from each resource a fixed number of candidate expansion terms that are relevant to the query Q. This yields to a single output list for the expanded query $gen_r(Q)$ generated from each resource r. Each of the expanded queries are used to retrieve a set of documents. These form several sets of diversified results. In the next layer, these results will be merged into a single set.

*** Second Layer:**

The second layer of our framework generates diversified search results from the retrieval lists obtained with different expanded queries. We use the MMR principle [22] to iteratively select d^* from the document candidates. Formula 3.12 defines this process:

$$d^* = argmax_{d \in DC_Q}(\lambda \cdot rel(d,Q) - (1-\lambda) \cdot max_{d_i \in DS_Q} sim(d,d_i)) \tag{3.12}$$

where DC_Q denotes the document candidates, which is initialized as D_Q; DS_Q denotes the set of selected documents, which is empty at the very beginning; λ is the parameter that controls the trade-off between relevance and diversity (which is set at 0.5 as it is generally the case with MMR); $rel(d,Q)$ measures the similarity between document d and query Q (which will described below); $sim(d,d_i)$ denotes the similarity between two documents (for simplicity, we use the cosine similarity in our experiments). The selected document d^* is then removed from DC_Q to DS_Q.

One core element of the second layer is $rel(d,Q)$, which is defined using Formula 3.13, where $rel(D_{r,Q},d)$ and $rank(D_{r,Q},d)$ are the normalized relevance score and the rank of document d in $D_{r,Q}$, respectively. For $d \notin D_{r,Q}$, we set $\frac{1}{rank(D_{r,Q},d)} = 0$. For the normalization of the relevance score, we use the exp function, *i.e.*, $x \leftarrow \frac{\exp x}{\sum_{x'} \exp x'}$. This formula captures our intuition that the more a document

is ranked on top and with high relevance score in different candidate lists, the more relevant it is to the query. The formula can also be seen as a relevance score normalized by the rank of the document, which plays a role of decaying factor.

$$rel(d,Q) = \sum_{r \in R} \frac{rel(D_{r,Q}, d)}{rank(D_{r,Q}, d)} \tag{3.13}$$

3.4 Experimental Setup and Datasets

In this section, we describe the setup for the experiments conducted in Section 3.5. These experiments aim to answer the following three research questions:

1. Is our proposed DQE approach effective at improving search results in terms of both relevance and diversity, compared to the state-of-the-art approaches?
2. What is the impact of integrating multiple resources compared to the use of a single resource?
3. What is the sensitivity of our proposed framework to the choice of some parameters?

The first two research questions will be addressed in Section 3.5.1 in which we run extensive experiments on TREC diversification data to evaluate our approach and compare it to other existing methods. Section 3.5.3 will be mainly dedicated to answer our third question. We will also provide in Section 3.5.2 an illustrative example to show the impact of combining multiple resources instead to using a single one.

3.4.1 Document Collections, Resources and Topics

Please refer to Section 2.1.1 and Section 2.1.2 for a full description of the document collection, topics and resources that we use in our experiments.

3.4.2 Evaluation Metrics

We consider several standard measures as performance metrics. For the relevance-based metrics, we use nDCG (normalized Discounted Cumulative Gain) [5] and ERR (Expected Reciprocal Rank) [26]. The former computes the *gain* of a document based on its position (or rank) in the result list. We

also use MAP (Mean Average Precision) [5] for adhoc performance. These are the standard measures used in IR. For the diversity-based metrics, we use α-nDCG [33] (in our experiments, α=0.5), ERR-IA [26], and NRBP [30] which reward novelty and penalize redundancy at each position from the list of ranked documents, based on how much the information contained in the document at some rank is already seen by the user from the set of documents returned at earlier ranks. We also use Prec-IA [2] which measures the precision across all subtopics of the query, and S-recall [125] which computes the ratio of covered subtopics in the search results. Please refer to Section 2.1.3 in which we explain how these metrics are defined. All of these metrics are computed on the top 20 documents retrieved by each model. Notice that these metrics have been widely used in the official evaluation of the diversity task at TREC.

Finally, for the test statistical significance, we use two-tailed t-test (p-value < 0.05) when comparing two systems, and we use the Tukey's honest significance test when comparing three systems and more. In fact, it has been demonstrated [25, 102] that if one would compare three systems or more, using pairwise tests may be jumping to wrong conclusions due to the family-wise errors [3] and Tukey's test could be appropriate to handle the family-wise errors. To run our significance statistical tests for both t-test and Tukey's test, we use *R* software [4] together with ANOVA [5].

3.4.3 Baselines and Diversification Frameworks

We compare our DQE method with the following systems:

- *BL*, the basic retrieval system, which is built with Indri and is based on a query generative language model with Dirichlet smoothing (μ=2000), Krovetz stemmer [75], and stopwords removal using the standard INQUERY stopword list;

- *SA*, the Spreading Activation framework [64, 65] to generate a query expansion based on ConceptNet.

- *MMR*, the basic search results re-ranking [22] which trade-offs relevance to non-redundancy at the document level;

- *xQuAD*, a probabilistic framework for search result diversification, which explicitly models a query

3. https://en.wikipedia.org/wiki/Familywise_error_rate
4. https://cran.r-project.org
5. http://www.gardenersown.co.uk/education/lectures/r/anova.htm

as a set of sub-queries [105].
It is worth noting that *MMR* and *xQuAD* are well known state-of-the-art SRD approaches, which show competitive results over the state-of-the-art diversification.

Hereafter, we denote by $MMRE_r$ our Maximal Marginal Relevance based Expansion method proposed in this chapter, which uses resource r to select candidate expansion terms. To further study the effectiveness of all the core components of our system, we build a reference system: *Comb*. *Comb* is the model which combines different resources. Given a query Q, *Comb* combines different sets of retrieved documents, each with an expanded query using $MMRE_r$ with resource r, as already described in Section 3.3.2. We choose to compare with this method in order to answer our second research question, in which we want to assess the impact of using multiple resources compared to a single one.

3.4.4 Parameter Settings

Our model and our considered baselines and diversification frameworks come with a number of parameters. All the parameters are determined using 3-fold cross validation. We use in turn each of the query sets from WT09, WT10 and WT11 for test while the other two sets for training. During this procedure, we optimise for α-nDCG@20 [33]. Each of the methods $MMRE_r$ (with resource r), *MMR* and *xQuAD* has one parameter (λ for *MMR* and *xQuAD* and λ_r for $MMRE_r$) to be tuned. We consider values of λ and λ_r in the range of [0.1, 1] with an increment of 0.1.

For *SA* (see Equation 3.1), we set the decay factor $C_{dd} = 0.5$ and we initialize the activation score of each node in the graph of ConceptNet to 1, following Hsu *et al.* [64].

The remaining free parameters that should be tuned are the following: K, the number of expansion terms that we consider for each query and from each resource; and *wind*, which is the window size that we used to select candidate expansion terms that co-occur with the query terms from the feedback documents. We vary these two parameters (K and *wind*) in the range of {5, 10, 15, ..., 40}.

Finally, it is worth noting that each selected expansion term is weighted based on its score calculated by our method $MMRE_r$, by using *#weight* operator in Indri. Using Indri, we retrieve the set of documents corresponding to the new expanded query.

3.5 Experimental Results

The goal of this section is to answer our three research questions.

3.5.1 Evaluation of MMRE

*** Impact of DQE**

In order to study the role of DQE compared to traditional (standard) QE, we choose to compare $MMRE_C$ with *SA* since both approaches use the same resource, that is ConceptNet, to select candidate expansion terms to expand an original query. Besides, to study the role of DQE compared to standard SRD, we also compare $MMRE_C$ with *MMR* since both approaches use similar principle (which trade-off relevance to non-redundancy). We test $MMRE_C$ with different values of radius ρ: $\rho = 1$, $\rho = 2$ and $\rho = 3$. Recall that parameter ρ in Algorithm 3.1 refers to the depth (*i.e.* the number of edges) that we consider in the graph of ConceptNet when selecting candidate expansion terms. Table 3.V reports our results for the query sets.

First, from these results, we can clearly observe that the best performance of diversified results were obtained using $MMRE_C$ on the three query sets (with $\rho = 1$ for WT09, and $\rho = 2$ for WT10 and WT11). The difference on ρ could be explained by the fact that the topics of WT10 and WT11 are known to be harder than the topics of WT09 (based on the MAP values). Hence, for WT10 and WT11, we need to traverse the graph of ConceptNet deeper to extract *good* terms for expansion. However, for WT09, a depth of 1 is sufficient to gather meaningful terms that can cover the different query subtopics. Note that the value $\rho = 3$ leads to a decrease of the performance. This result was expected because whenever we go farther in the graph of ConceptNet, the expansion is likely to bring in more noisy terms.

Second, by observing the results using *SA*, we can see that standard QE based on ConceptNet statistically improves adhoc retrieval performance compared to *BL*, but does not improve a lot in terms of diversity. On the other hand, the use of $MMRE_C$ yields to a significant improvement not only in relevance, but also in diversity, over the three query sets. This result can be explained as follows: Despite that both $MMRE_C$ and *SA* use the same external resource (*i.e.*, ConceptNet) to collect expansion terms, the latter selects terms that are *globally* relevant to the query, while the former selects diverse terms that are related to different query aspects (in addition to be relevant to

Query sets	*Model*	*MAP*	*nDCG*	*α-nDCG*	*ERR-IA*	*S-recall*
WT09	*BL*	0.161§	0.240§	0.188§	0.097	0.367§
	SA	0.176-§	0.258*§	0.203§	0.109§	0.391-§
	MMR	0.166§	0.246§	0.191§	0.103§	0.377§
	$MMRE_C$ ($\rho = 1$)	**0.195***+-§	**0.293***-§	**0.269***+-§	**0.140***-§	**0.482***+-§‡
	$MMRE_C$ ($\rho = 2$)	0.182*-§	0.272*§	0.244+-§	0.121*§	0.427*§
	$MMRE_C$ ($\rho = 3$)	0.092	0.124	0.109	0.058	0.199
WT10	*BL*	0.103§	0.115§	0.198§	0.110	0.442§
	SA	0.116-§	0.139*-§	0.235*§	0.122§	0.480§
	MMR	0.106§	0.119§	0.209*§	0.111	0.459§
	$MMRE_C$ ($\rho = 1$)	0.128*-§	0.162*-§	0.267*+-§	0.138§	0.556*+§
	$MMRE_C$ ($\rho = 2$)	**0.146***+-§	**0.196***+-§	**0.293***+-§	**0.165***+§	**0.664***+-§♮
	$MMRE_C$ ($\rho = 3$)	0.059	0.067	0.115	0.077	0.282
WT11	*BL*	0.093	0.155§	0.380§	0.272§	0.700§
	SA	0.115*-§	0.232*-§	0.405§	0.284§	0.786-§
	MMR	0.096	0.159§	0.382§	0.269§	0.714§
	$MMRE_C$ ($\rho = 1$)	0.142*+§	0.291*§	0.481*+§	0.340*+§	0.945*+§
	$MMRE_C$ ($\rho = 2$)	**0.155***+-§	**0.320***-§	**0.552***+-§♮	**0.397***+-§	**0.975***+-§
	$MMRE_C$ ($\rho = 3$)	0.047	0.091	0.153	0.115	0.331

Table 3.V: Comparison between DQE and standard QE. *, +, -, ♮, ‡ and § means significant improvement over *BL*, *SA*, *MMR*, $MMRE_C$ ($\rho = 1$), $MMRE_C$ ($\rho = 2$) and $MMRE_C$ ($\rho = 3$), respectively ($p<0.05$ in Tukey's test).

the original query). Therefore, the diversity of the retrieval results with the former is better.

Third, we observe that *MMR*, which is one of the state-of-the-art SRD approaches, can also improve the performance, but only marginally, compared to *BL*. Applying *MMRE* to a query generates a set of results that are more relevant and diversified than those given by *MMR*. In fact, when the set of retrieval results corresponding to the original query is not diverse, even applying a good reranking strategy (such as *MMR*), we cannot cover any aspect that was not covered by the original retrieval results. This comparison confirms that it is necessary to diversify the query to be able to retrieve more diverse documents. This is a more effective approach than trying to select diverse documents directly from the results of the initial query.

Since the work of Vargas *et al.* [119] using xQuAD [104] is very close to ours (both approaches perform diversified query expansion), we also compare the effectiveness of *MMRE* to it using xQuAD as described in [119]. Recall that xQuAD use the same subset of 116 topics as in Vargas *et al.* [119].

The authors in [119] impose some constraints on the query sets. For example, they consider only the topics having the same number of relevant documents for each TREC subtopic, and each subtopic must have at least six relevant documents according to the TREC assessors. These requirements eliminate 34 topics and leave 116 topics of the 150 WT09, WT10 and WT11 topics. To make a fair comparison between the work of Vargas *et al.* [119] and ours, we use the same subset of 116 topics. The results are reported in Table 3.VI.

Query sets	*Model*	*MAP*	*nDCG*	*α-nDCG*	*ERR-IA*	*S-recall*
116 topics of WT09, WT10 and WT11	*xQuAD*	0.160-	0.387-	0.538-	0.433-	0.792-
	$MMRE_C$ $(\rho = 1)$	0.175-	0.399-	0.526-	0.412-	0.864*-
	$MMRE_C$ $(\rho = 2)$	**0.206***+-	**0.425**-	**0.547**-	**0.440**-	**0.895*** -
	$MMRE_C$ $(\rho = 3)$	0.060	0.101	0.218	0.135	0.365

Table 3.VI: Results for the selected queries in [119]. *, + and - means the improvement over *xQuAD*, $MMRE_C$ $(\rho = 1)$ and $MMRE_C$ $(\rho = 3)$, respectively is statistically significant (p<0.05 in Tukey's test).

As shown in Table 3.VI, *MMRE* with $\rho = 2$ outperforms *xQuAD* on all the measures. This shows that our method can better diversify search results than *xQuAD*. This could be due to the resource used for selecting expansion terms. In fact, Vargas *et al.* [119], the authors rely on the PRF to select candidate expansion terms. Therefore, the quality of expansion terms usually depends on that of the retrieval results, which may involve non-relevant documents, especially for ambiguous and difficult queries [3].

*** Resource Combination and Impact of Different Resources**

To understand the effect of different resources in our task, as well as the impact of combining multiple resources compared to the use of a single one, we run additional experiments. Table 3.VII reports our evaluation results, from which we make four main observations.

First, we observe that $MMRE_r$ using any resource statistically outperforms *MMR* (which is a document level diversification approach) in most of the adhoc and diversity measures. This clrealy shows that DQE is more effective than traditional diversification.

Second, among all the resources used alone, query logs often yields significantly better adhoc retrieval performance and diversity than other resources. This can be because the candidate expansion

Query sets	*Model*	*MAP*	*nDCG*	*α-nDCG*	*ERR-IA*	*S-recall*
WT09	*BL*	0.161	0.240	0.188	0.097	0.367
	MMR	0.166	0.246	0.191	0.103	0.377
	$MMRE_C$	0.195*‡	0.293*‡	0.269*-‡	0.140*-‡	0.482*-
	$MMRE_W$	0.208*-‡	0.319*-‡	0.274*-‡	0.146*-‡	0.510*-‡
	$MMRE_L$	0.221*-+‡	0.340*-+§‡	0.295*-+‡	0.153*-‡	0.599*-+§‡
	$MMRE_F$	0.188	0.276*	0.224*	0.115	0.435*
	Comb	**0.258***-+§♮‡	**0.379***-+§♮‡	**0.328***-+§♮‡	**0.181***-+§♮‡	**0.672***-+§♮‡
WT10	*BL*	0.103	0.115	0.198	0.110	0.442
	MMR	0.106	0.119	0.209	0.111	0.459
	$MMRE_C$	0.146*-‡	0.196*-‡	0.293*-‡	0.165*‡	0.664*‡
	$MMRE_W$	0.149*-‡	0.203*-‡	0.317*-+‡	0.174*-‡	0.683*-‡
	$MMRE_L$	0.158*-§‡	0.221*-+‡	0.341*-+§‡	0.182*-+‡	0.694*-‡
	$MMRE_F$	0.117	0.142*	0.225*	0.148*	0.508*
	Comb	**0.173***-+§♮‡	**0.239***-+§♮‡	**0.352***-+§♮‡	**0.195***-+§♮‡	**0.703***-+§‡
WT11	*BL*	0.093	0.155	0.380	0.272	0.700
	MMR	0.096	0.159	0.382	0.269	0.714
	$MMRE_C$	0.155*-§‡	0.320*-§‡	0.552*-§‡	0.397*-§‡	0.975*-§‡
	$MMRE_W$	0.124*-‡	0.255*-‡	0.449*-‡	0.313*-‡	0.798*-‡
	$MMRE_L$	0.160*-§‡	0.342*-§‡	0.578*-§‡	0.411*-§‡	0.982*-§‡
	$MMRE_F$	0.104	0.163	0.397	0.279	0.733
	Comb	**0.167***-+§‡	**0.359***-+§♮‡	**0.586***-+§‡	**0.422***-+§‡	**0.990***-+§‡

Table 3.VII: Experimental results of different models on TREC Web tracks query sets. $MMRE_C$, $MMRE_W$, $MMRE_L$, and $MMRE_D$ refer to the *MMRE* model based on ConceptNet, Wikipedia, query logs, and feedback documents, respectively; *Comb* denotes the model combining all the four resources. *, -, +, §,♮, and ‡ indicate significant improvement ($p < 0.05$ in Tukey's test) over *BL*, *MMR*, $MMRE_C$, $MMRE_W$, $MMRE_L$, and $MMRE_F$, respectively.

terms generated from query logs are those suggested by users (through their query reformulations), which could better reflect the user intents. This suggests the important role of query logs for the diversity task. Besides, as most of the queries of WT09, WT10 and WT11 that we consider are from the MSN query logs of 2006, which have a good coverage of the topics, candidate expansion terms suggested from this resource are of good quality.

Third, Wikipedia outperforms ConceptNet for WT09 and WT10 topics, but not significantly in general. However, ConceptNet significantly outperforms Wikipedia for WT11 topics in all the measures. To understand the reason, we manually assessed the different queries to see whether they have an exact matching page from Wikipedia. We found that 36/50, 34/48 and 18/50 queries from WT09,

WT10 and WT11 respectively, have exact matching pages from Wikipedia (including the disambiguation and redirection pages), and that only when the query corresponds to a known concept (*i.e.* page) from Wikipedia, the candidate expansion terms suggested by Wikipedia tend to be relevant. These numbers are consistent with the improvements we obtain with $MMRE_W$. This means that Wikipedia helps promoting the diversity of the query results, if the query corresponds to a known concept.

Fourth, the set of feedback documents has the poorest performance among all resources under consideration. Its performance drastically decreases from WT09 to WT10 to WT11 in terms of relevance and diversity. This may be due to the fact that the topics of WT11 are harder than the topics of WT10, and the topics of the latter are harder than those of WT09 (based on the MAP values). The more the collection contains difficult queries, the more likely the set of top returned documents are irrelevant. Hence, the candidate expansion terms generated from these documents tend to include a lot of noise.

Finally, combing all these resources gives better performance, and in most cases the improvement is significant on almost all the measures. In particular, the diversity scores obtained (for α-nDCG@20, ERR-IA@20, and S-recall@20), are the highest. This means that the considered resources are *complementary* in term of coverage of query subtopics: the subtopics missed by some resources can be recovered by other ones, as demonstrated by the example query "*defender*" that we described before in Section 3.3.1.

3.5.2 Illustrative Query Example

Let's consider "*Neil Young*", the #73 query from the WT10 [31], as an example. This query is not ambiguous and has four different subtopics [6], as described in Table 3.VIII.

Subtopic	*Description*
1	Find albums by Neil Young to buy.
2	Find biographical information about Neil Young.
3	Find lyrics or sheet music for Neil Young's songs.
4	Find a list of Neil Young tour dates.

Table 3.VIII: List of the TREC subtopics for the query "*Neil Young*".

To generate expansion terms for this this query, we run $MMRE_r$ using the different resources that

6. `http://trec.nist.gov/data/web/10/wt2010-topics.xml`

we consider in this work. We usually add the original query terms to the set of selected expansion terms. Results are reported in Table 3.IX.

Model	*Expansion Terms (in decreasing order of importance)*
$MMRE_C$	**neil young** person[2] canadian[2] music[1,3] wife[*] star[2] bio[2] award[2] album[1] film[2] song[1,3]
$MMRE_W$	**neil young** canadian[2] acoustic[*] harvest[1] singer[2] award[2] rock [2] buffalo[2] music[1,3] california[*] band[2]
$MMRE_L$	**neil young** chords[3] lyrics[3] ticket[4] concert[4] alabama[3] song [1,3] tour[4] war[1] photo[*] drawings[*]
$MMRE_F$	**neil young** canada[2] man[2] instrument[*] birth[2] sun[*] mp3[3] legend[2] guitar[*] philadelphia[2] song[1,3]

Table 3.IX: Expansion terms for "*Neil Young*" generated by using different resources and outputted by $MMRE_r$. We manually tag each expansion term by its corresponding TREC subtopic number (from 1 to 4). * means that the expansion term does not clearly correspond to any of the subtopics. One expansion term could be simultaneously relevant to more than one subtopic.

From Table 3.IX, we observe that different resources cover different subtopics for the query "*Neil Young*". For instance, based on our manual investigation of the expansion terms suggested by different resources, we find that: Each of the resources ConceptNet, Wikipedia and pseudo-feedback documents covers subtopics 1, 2 and 3; while query logs covers subtopics 1, 3 and 4. By combining all these resources, on may expect a better coverage of all the subtopics underlying the query. In Table 3.X, we show the effectiveness of different resources using *MMRE* for the same query "*Neil Young*", as well as the effectiveness of resource combination (*Comb*), and that of traditional SRD approaches (*MMR*).

Model	*MAP*	*nDCG*	*α-nDCG*	*ERR-IA*	*S-recall*
MMR	0.129	0.134	0.254	0.117	0.250
$MMRE_C$	0.217	0.192	0.284	0.183	0.500
$MMRE_W$	0.220	0.188	0.278	0.169	0.500
$MMRE_L$	0.235	0.210	0.295	0.191	0.500
$MMRE_F$	0.145	0.147	0.266	0.126	0.250
Comb	**0.291**	**0.273**	**0.330**	**0.218**	**0.750**

Table 3.X: Experimental results of *MMRE* across different resources, *Comb* and *MMR* on "*Neil Young*".

From the statistics reported in Table 3.X, we clearly see that combining multiple resources yields

a better result in terms of relevance and diversity, compared to the use of any single resource. In particular, by observing S-recall@20 measure (which reports the percentage of query subtopics covered by the retrieval results), we can further confirm the capability of our resource combination method (*Comb*) to cover the query aspects. Moreover, $MMRE_r$ using any resource r outperforms *MMR* in all the measures, which highlights the role that plays DQE compared to traditional SRD approaches.

3.5.3 Parameter Sensitivity Analysis

The purpose of this section is to answer our third and last research question on whether our proposed framework is sensitive to the choice of some parameters. It is interesting to assess the sensitivity of *MMRE* to K, which refers to the total number of expansion terms that we keep at the end for each query. To do this, we vary K = 5, 10, 15, 20 and 30, and compare the performance of *Comb* and $MMRE_r$ across different resources r. Here, we only show the results of *Comb* on WT09 queries, but we make similar observation for the other models $MMRE_r$ and using the other query sets (*i.e.*, WT10 and WT11). Our results are plotted in Figure 3.2.

First, we observe that K=10 corresponds to the optimal parameter value yielding to the best relevance and diversity scores of *Comb*. Second, from K=5 to K=10, both relevance and diversity scores drastically increase. A possible explanation is the more we add expansion terms, the more likely we clarify the query meaning (increase relevance scores) and also the more likely we cover different aspects of the query (increase diversity scores). Besides, even among a few expansion terms, our approach can ensure good results in both relevance and diversity. This is because the expansion terms selected by our method are relevant to the original query and can cover different aspects of the query, from the earlier iterations of the *MMRE* procedure. However, starting from K=15, we observe decreasing relevance and diversity scores. This when a large number of expansion terms are introduced, we have a higher chance of incorporating redundant and noisy terms, resulting in less relevant documents. For $K = 30$, the performance of *Comb* becomes even lower than the standard baseline (*BL*).

Another parameter that we consider in $MMRE_F$ is *wind*, which is the window size that we used to select candidate expansion terms that co-occur with the query terms from the feedback documents. To study the sensitivity of $MMRE_F$ to the window size, we also vary this parameters *wind* = 5, 10, 15, 20, and 30, and compare the performance of $MMRE_F$. Our results are plotted in Figure 3.3. We

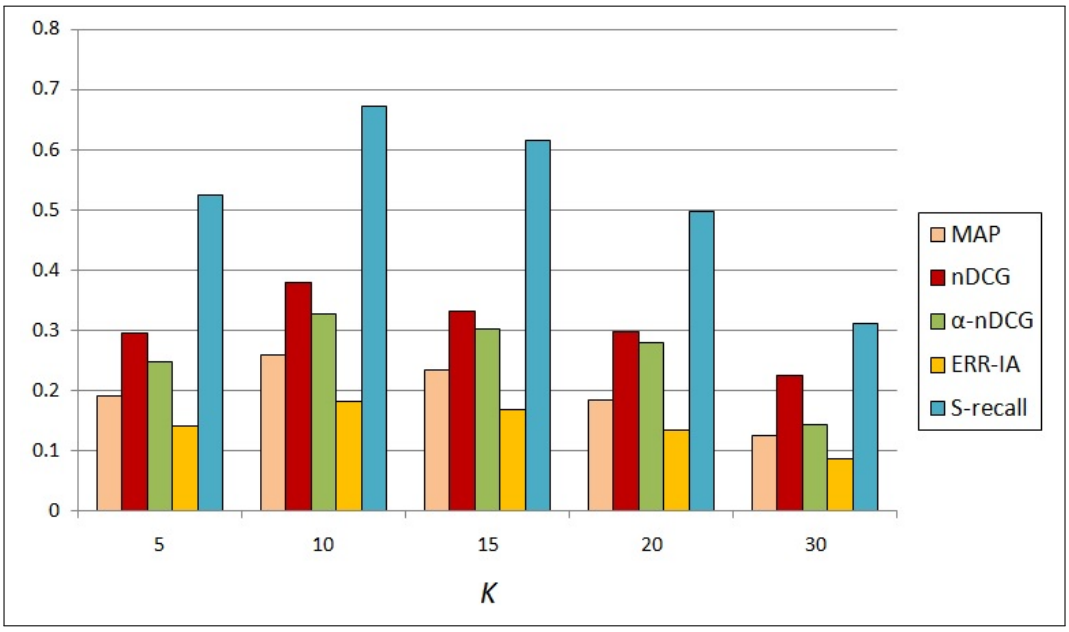

Figure 3.2: Performance of *Comb* when varying the number of expansion terms (K) on WT09 queries.

only report our statistics on WT09 queries (we observe similar trends on WT10 and WT11 queries).

From Figure 3.3, we observe that when we increase the window size from 5 to 10 to 15. both relevance and diversity performance of $MMRE_F$ is improved. In fact, the window size parameter allows us to look at different scales. Smaller window sizes (*e.g.* $wind = 5$) will identify expansion terms that co-occur within short ranges (*i.e.* appear near each other), and which are directly related [29, 114]. Larger window sizes (*e.g.* $wind = 10$ and 15) will include more related terms within larger contexts. The latter may represent a higher diversify.

Starting from $wind = 20$, we observe that the performance of $MMRE_F$ drastically decreases. This may be due to the fact that, when the window size becomes larger, we run the risk of introducing noise expansion terms (which are far from being related to the query and its subtopics). Such noise expansion terms will have a negative impact on the effectiveness of $MMRE_F$. Consequently, the window size plays an important role on deciding the effectiveness of $MMRE_F$: one should carefully set this parameter (*wind*).

Finally, we conjecture that different queries may require different window sizes. For instance,

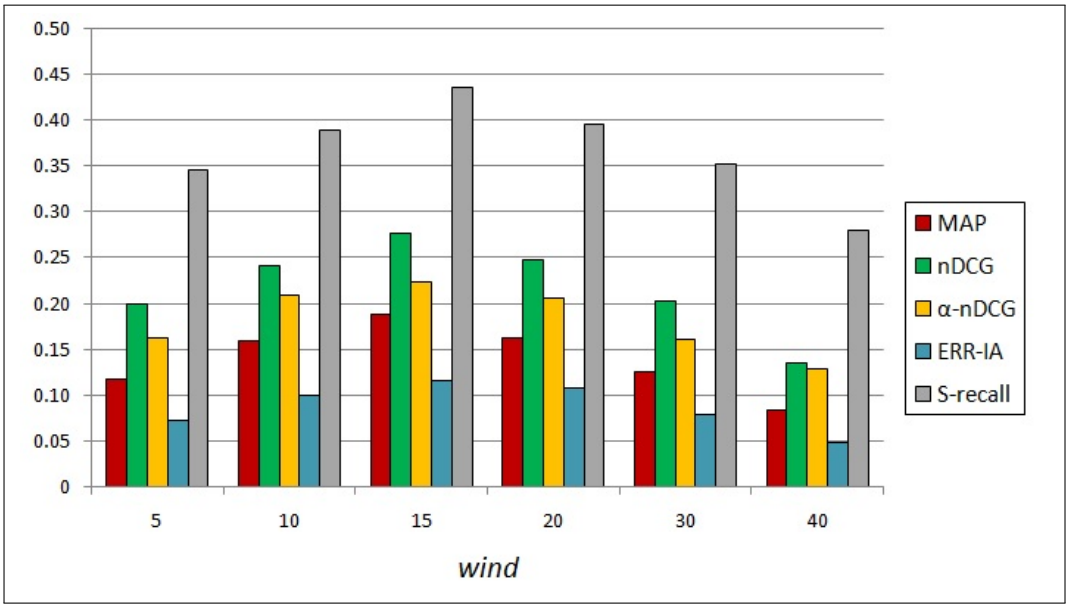

Figure 3.3: Performance of $MMRE_F$ when varying the window size (*wind*) on WT09 queries.

we observe that queries such as "*defender*" (query #20 in WT09) requires a window size of 30 due to its ambiguity, while a window size of 5 is enough to suggest good expansion terms for the query "*mothers day songs*" (query #132 in WT11). This observation inspires us to design an approach for $MMRE_F$ which selectively chooses the window size with respect to each query, when selecting candidate expansion terms for DQE. We leave that for our future research.

3.6 Approach Analysis

In this section, we analyse in more depth our proposed approach. In particular, we will study the impact of relevance and diversity components in *MMRE*, and the impact of the choice of similarity functions to the overall results. We will also discuss the complexity of our method.

3.6.1 Relevance/Diversity Analysis

From Table 3.VII, one can observe that when diversity measures increase, relevance measures also increase. In [8], Bendersky *et al.* make the same observation. So a legitimate question is whether the improvements in diversity are mainly due to the fact that good expansion terms are selected,

regardless of their diversity.

To answer this question, we perform another test in which traditional query expansion approaches are used, with the same four resources: For each resource r, we run the corresponding $MMRE_r$ procedure (similar to Formula 3.6) for each query. Recall that parameter λ_r in Formula 3.6 controls the trade-off between relevance and diversity. By setting λ_r=1 for each resource r, we only consider the relevance and ignore the diversity, which corresponds to a standard query expansion method. Table 3.XI shows the results where we set λ_r to 1 or to a non-zero value according to cross validation. We only show the results of the queries of WT09. On results of the queries of WT10 and WT11, we make comparable observations.

Model	**nDCG**	**ERR**	**α-nDCG**	**ERR-IA**	**S-recall**
BL	0.240	0.117	0.188	0.097	0.367
$MMRE_C(\lambda_C=1)$	**0.314**	**0.129**	0.154	0.108	0.204
$MMRE_C$	0.293	0.123	**0.269**	**0.140**	**0.482**
$MMRE_W(\lambda_W=1)$	**0.327**	**0.141**	0.125	0.094	0.219
$MMRE_W$	0.319	0.130	**0.274**	**0.146**	**0.510**
$MMRE_L(\lambda_L=1)$	**0.355**	**0.148**	0.133	0.087	0.178
$MMRE_L$	0.340	0.142	**0295**	**0.153**	**0.599**
$MMRE_F(\lambda_F=1)$	**0.284**	**0.133**	0.119	0.071	0.216
$MMRE_F$	0.276	0.120	**0.224**	**0.115**	**0.435**

Table 3.XI: Comparison of the DQE method with a standard QE method using different resources on WT09 queries.

From the results of Table 3.XI, we observe that this traditional query expansion approach can indeed improve on relevance measures. However, the diversity measures are not improved, and instead they are hurt. This clearly shows the difference between relevance and diversity. A traditional query expansion method is unable to improve diversification search. On the other hand, a diversified query expansion will increase diversity, but it increases relevance less than the traditional QE. These results show that diversity and relevance could be incompatible to some extent: increasing one could hurt the other.

3.6.2 Impact of Similarity Functions

The similarity functions $sim_r(.,.)$ that we used in this work (see Section 3.3.2) measure how similar two candidate expansion terms for a given query with respect to one resource r. One can ask the question: How much these similarity functions contribute to the whole DQE framework? To answer this question, we conduct additional experiments on some special similarity settings. In particular, we examine two cases:

(1) We set all the similarities between expansion terms to a constant value within the range [0, 1] (*e.g.*, 0.5).

(2) We use a random similarity between any pair of expansion terms within the range [0, 1].

Table 3.XII reports the performance of $MMRE_L$ when using different settings of similarity functions (we observe similar trends for $MMRE_W$, $MMRE_C$ and $MMRE_F$). We report our results based on two adhoc relevance metrics (nDCG and ERR) and two diversity metrics (α-nDCG and S-recall), computed on 148 queries from WT09, WT10 and WT11 queries.

Similarity Functions	nDCG	ERR	α-nDCG	S-recall
(Constant values)	0.294	**0.166**	0.112	0.174
(Random values)	0.128	0.109	0.236	0.385
(As defined in Section 3.3.2)	**0.301**	0.152	**0.405**	**0.758**

Table 3.XII: Performance of $MMRE_L$ when using different settings of similarity functions on 148 queries from WT09, WT10 and WT11.

From the results of Table 3.XII, we can clearly see that $MMRE_L$ with constant and random similarity values perform worse than the similarity functions that we defined in this work, in almost all the relevance and diversity measures. In particular, our model with random similarity functions perform even worse that the standard baseline (*BL*). This observation demonstrates the importance of similarity functions in our framework; These functions should be properly defined and should reflect well the semantic relation between expansion terms. A bad choice of these similarity functions can drastically decrease a lot the performance of our framework.

Surprisingly, we find that $MMRE_L$ with constant similarity functions yields a very competitive relevance scores (in terms of nDCG and ERR) compared to $MMRE_L$ when using the similarity functions that we defined in this chapter. This result can be explained as follows: when a constant similarity

function is used, the non-redundancy part of the *MMRE* Formula (see Equation 3.6) is neglected since all pair of expansion terms have the same similarity score. Hence, only the relevance part of the *MMRE* Formula is considered to explicitly distinguish between the candidate expansion terms that should be selected. Consequently, our *MMRE* procedure is reduced to a standard QE method. This may explain why we obtain good relevance scores for this model, while diversity scores (α-nDCG and S-recall) are hurt.

3.6.3 Complexity Analysis

Complexity issues can be tackled by noting that expansion terms similarity based on each resource is computed off-line (except for PRF), thus eliminating any additional on-line costs. During this process, we select from each resource, and for each query, a few candidate expansion terms. As there are a limited number of resources (we used 4 resources in this study), and a limited number of candidate expansion terms for each query, and from each resource (this number is set to 10), the whole amount of computation is generally limited and its complexity is $\mathcal{O}(1)$.

The on-line process is to select K expansion terms for each query and regarding to each resource (see Algorithm 3.1). In each iteration of the *MMRE* procedure, we compute the similarity between an expansion term and the original query, and between a pair of expansion terms. For the former computation, we need to perform only M calculations in the first iteration, which is $\mathcal{O}(M)$, where M denotes the total number of candidate expansion terms that we consider from each resource, and each query. Note that we also consider the combination of all the possible subsets of the query terms. This yields $2^{|Q|} * M$ similarities to be calculated, where $|Q|$ is the number of query terms. Since $|Q|$ is generally small (a few query terms), $2^{|Q|}$ could be ignored. Note that these similarity scores could be directly used for the next iterations of *MMRE* and we don't need to re-compute them for each iteration of *MMRE*. Similarly, the latter computation (*i.e.*, the similarity between a pair of expansion terms) which requires $\frac{M \cdot (M-1)}{2}$ calculations is also done only during the first iteration of *MMRE*. This is because $sim(e_i, e_j) = sim(e_j, e_i)$ where e_i and e_j are two expansion terms for the same query. Therefore, the complexity of the whole *MMRE* on-line process is of $\mathcal{O}(M^2)$. As there are a limited number of candidate expansion terms for a query (M=50 in our experiments), the whole amount of computation is generally limited. This make it possible to deploy our approach in a real system. In fact, the computation of similarity between pairs of terms can be done offline. The remaining online

calculation is for the similarity between an expansion candidate and the query.

3.7 Discussion

Several issues in this work are worth some further discussions. First, one can notice that there is a temporal offset of the data sources that we used in our experiments. For instance, Wikipedia data set is from 2013, query logs are from 2006, and TREC queries are from 2009-2011. We believe that this temporal offset could slightly affect the end results. If we use more recent click logs data for example, several query aspects will be better covered. As an example, let's consider the query #113 of WT11: "hp mini 2140". This latter electronic device was announced for the first time in 2009. Our log data of Microsoft Live Search are from May 2006, thus are unable to suggest expansion terms for this query. Ideally, one should consider queries and data sources from the same temporal interval, so that expansion terms suggested by each resource are 'up to date' with the user queries and her intents.

Second, the similarity functions $sim_r(.,.)$ that we proposed in this work provide good results in practice which explains why our framework outperforms other state-of-the-art approaches. We believe that better similarity functions yield better results. However, we did not strive to define the best similarity function in our work. We leave this for our future work. Besides, when defining these similarity functions based on different resources, we didn't take into account the different types of semantic relations between terms. For example, ConceptNet incorporates 20 different semantic relations between terms (nodes), and these relations do not necessarily have the same importance. In the future, we will investigate in more depth the choice of these functions by considering the different types of relations involved in the graph of resources and try to design more complete and accurate similarity functions that reflect better the query aspects and their dependency.

Third, recall when extracting candidate expansion terms from ConceptNet, we consider different values of the radius which refer to the depth (*i.e.* the number of edges) that we explore in the graph of ConceptNet. However, when expansion terms are weighted, we don't consider the values of the radius. In particular, an expansion term of radius 1 should be highly weighted than any other expansion term of radius 2 or 3 since it is more closely to the query terms. We don't address this issue in this dissertation and we leave that for our future research.

Fourth, in this work, we consider different resources for a better coverage of the query aspects.

However, all the resources are used uniformly and no proper weighting is considered. In practice, different resources do not necessarily contribute by the same degree for different queries, and that using good strategies to define the weights of resources could substantially improve our results. Different methods could be adopted. For example, given a query, the weight of a resource could be proportional to the number of different expansion terms suggested by the resource for that query; or it could be defined based on some metric that we want to optimize (such as α-nDCG [33]). Ideally, one can also learn the weight of the resource based on the features of the query. These issues will be investigated in detail in the chapter 4 of this dissertation.

Finally, when varying the number of expansion terms that we should select from each resource (parameter K), we observe that different queries require a different number of expansion terms. For instance, the best performance of the ambiguous query "defender" (query #20 from WT09) is reached when K=30, while 5 expansion terms are enough for the query "mothers day songs" (query #132 from WT11) to obtain good results. In the future, it could be useful to develop ways to automatically determine the appropriate K according to the query. In particular, one could think to develop a model which learns the number of expansion terms for each query based on a set of features.

3.8 Conclusion

In this chapter, we present a unified framework for diversified query expansion, which may integrate a single or multiple resources. By implementing two functions, one to generate expansion term candidates and the other to compute the similarity of two terms, any resource can be plugged into this framework. Experimental results on TREC 2009, 2010 and 2011 Web tracks and using four resources (ConceptNet, Wikipedia, query logs and feedback documents) show that our proposed DQE method significantly outperforms traditional diversification methods in both relevance and diversity, and that combining several complementary resources performs better than using any single resource.

When analyzing our results, we have observed that the degree of the contribution of a resource to SRD depends on the query. In chapter 4, we will develop other approaches to tackle this problem.

Chapter 4

Query-Dependent Resource Weighting for Diversified Query Expansion

4.1 Introduction

In chapter 3, we introduced a new DQE method as a way to generate diversified search results, motivated by the fact that the initial search results of the original query may not be diverse enough and some of the subtopics of the original query may be missing. One critical step of DQE is to expand the original query in different directions so as to identify better diversified results. This expansion step often relies on one or multiple external resources, *e.g.*, ConceptNet, Wikipedia, and query logs. Since multiple resources tend to complement each other for DQE, integrating multiple resources can often yield substantial improvements (better diversified results) compared to using one single resource as we observed in the previous chapter.

However, all resources do not contribute equally to the diversity of search results; different resources may have different impact on different queries. This is because a resource may better cover the topic of a query than another resource. In this chapter, we advocate that we should assign proper resource weights while building a DQE based SRD system with multiple resources. This work focuses on the problem of proper resource weighting. Once the resources are weighted, we use the *MMRE* approach proposed in chapter 3 to incorporate the resources into the SRD system, *i.e.*, selecting a number of expansion candidates from a resource that is proportional to the weight of that resource, and using resource weights to adjust the weights of the finally selected expansion terms.

One straightforward approach to modeling resource weight is to compute the average contribution of a resource to SRD on all the queries for training. Experimentally, we find this overall resource weighting approach, though simple, significantly improves the α-nDCG [33] and S-recall [125] scores on the three TREC topic sets. However this approach suffers from one issue: it ignores the fact the contribution of a resource to SRD varies depending on the query. To address this limitation, we develop, in this chapter, a linear regression model to compute query level resource weighting, which considers 39 features (Bouchoucha *et al.* [16]). We will experimentally show that

the SRD performance can be further improved using query-dependent resource weighting. The main content of this chapter corresponds to the following paper published at ECIR 2015: "Towards Query Level Resource Weighting for Diversified Query Expansion" [16]. Some minor modifications are made.

4.2 Motivation Example

In chapter 3, we showed an example to motivate the use of multiple resources. Let us consider another query - "*avp*", the #52 query from WT10. This query is ambiguous and has seven subtopics [1] as described in Table 4.I. Using different resources - Wikipedia, query logs and feedback documents, we can respectively cover the following subsets of subtopics: {1, 5, 6, 7}, {2, 3, 6, 7} and {4, 6}. For this query, ConceptNet does not cover any of the subtopics. It can be seen that each single resource can cover only part of the subtopics and by combining all these resources, one may expect to get better coverage of all the query subtopics.

Subtopic	*Description*
1	Go to the homepage for the AVP, sponsor of professional beach volleyball events.
2	Find information about pro beach volleyball tournaments and events sponsored by AVP.
3	Find the homepage for AVP antivirus software.
4	Find reviews of AVP antivirus software and comparisons to other products.
5	Find information about the Avon Products (AVP) company.
6	Find sites devoted to the "Alien vs. Predator" movie franchise.
7	Find information about Wilkes-Barre Scranton International Airport in Pennsylvania (airport code AVP).

Table 4.I: List of the TREC subtopics for the query "*avp*".

When multiple resources are considered, DQE faces the challenge of properly weighting a resource, or computing a non-negative real number for a resource which indicates the degree of the contribution of that resource to the SRD performance. Resource weighting should be done for two reasons. On one hand, the usefulness of a resource can greatly change depending on the queries. Resource weighting gives us a means to estimate how useful it is for a query. On the other hand,

1. `http://trec.nist.gov/data/web/10/wt2010-topics.xml`

the weight of a resource is a key factor in selecting candidate expansion terms: the expansion terms recommended by a resource with a larger weight should be preferred since they are more likely to be related to one or several subtopics of the query, and their combination tends to cover a good part of all the subtopics.

Existing studies that combine multiple resources to perform DQE based SRD largely overlooked this problem. Different resources were either simply merged together [61] or assigned the same weight [14] as we did in the previous chapter, regardless of the resource and of the query. Even though using several resources can potentially increase the coverage of subtopics, the lack of a proper resource weighting can jeopardize the real impact of the resources. Intuitively, a proper utilization of different resources depending on the query could yield better SRD performance because more appropriate expansion terms can be selected for the query.

To be convinced, let us examine again the example of the query "*avp*" that we showed before. Table 4.II shows 2 sets of expansion terms corresponding to this query. These terms are selected from 2 resources (Wikipedia and feedback documents) by using our proposed DQE method *MMRE* described in chapter 3.

Wikipedia	*volleyball*[1,2]*, enterprise*[5]*, alien*[6]*, violence*[*]*, avon*[5]*, film*[6]*, beach*[1,2]*, pennsylvania*[7]*, wilkes-barre*[7]*, casting*[*]*.*
Feeback documents	*news*[*]*, price*[*]*, product*[4]*, planet*[*]*, movie*[6]*, game*[*]*, world*[*]*, version*[4]*, alien*[6]*, download*[*]*.*

Table 4.II: Two sets of expansion terms selected for the query "*avp*", from Wikipedia and feedback documents, respectively. We manually tag each expansion term by its corresponding TREC subtopic number (from 1 to 7). * means that the expansion term does not clearly correspond to any of the subtopics. One expansion term could be simultaneously relevant to more than one subtopic.

From Table 4.II, we clearly observe that the expansion terms from Wikipedia are more related to the query than the ones selected from feedback documents: Expansion terms from Wikipedia cover subtopics 1, 2, 5, 6 and 7, while expansion terms from feedback documents cover only subtopics 4 and 6. Expansion terms from Wikipedia are more closely related to the subtopics manually identified by TREC assessors for the query "*avp*", which are already described in Table4.I. This means that Wikipedia is a good resource for the query "*avp*", while the feedback documents seem less appropriate for the same query. In the absence of a proper weighting of these two resources, one can only select terms *uniformly* from both resources, thus introducing noise terms (those that are irrelevant to the

query). To benefit from the high-quality of expansion terms obtained from Wikipedia, one should assign a higher importance to it.

4.3 Proposed Framework

In this section, we first give a formal definition of our task. Then we present the details of our query level resource weighting framework based on linear regression. Finally, we describe the set of features used to learn the regression model for resource weighting.

4.3.1 Task of Resource Weighting

In the context of DQE based SRD with multiple resources, given a query and a set of resources as input, the task of resource weighting outputs a non-negative and normalized real number for each resource that is proportional to the degree to which that resource can help to diversify the search results for that query. Hereafter, we will use Q to denote the query, r a resource, R the set of resources under consideration, and $w(r,Q)$ the weight of resource r for query Q.

In this study, resource weights are used in the same way as in *MMRE*. In particular, we generate a set of candidate expansion terms from each resource $r \in R$, which has a strong relation with the query (query terms). The similarity of a candidate expansion term e to an original query Q (denoted by $sim_r(e,Q)$ hereafter) is measured according to the resource r as already explained in Section 3.3. For example, ConceptNet can suggest terms that are connected to query terms in the ConceptNet graph; feedback documents can suggest terms that co-occur often in text windows with query terms; Wikipedia suggest terms that share the same anchor text and Wikipedia categories; and query logs suggest terms that appear in the same query sessions as the query.

Afterwards, we decide the number of expansion terms ($n(r,Q)$) that we should keep from each resource. We set this number proportionally to the weight of that resource $w(r,Q)$ (which is to be determined by a regression method), as follows:

$$n(r,Q) = \lceil \frac{w(r,Q)}{\sum_{r' \in R} w(Q,r')} \cdot K \rceil \tag{4.1}$$

where K is the total number of expansion terms to select. Equation 4.1 encodes our intuition that the

more a resource is important for a query, the more we should select expansion terms from it. Note that, in our experiments,we generally select 10 expansion terms for each query. However, due to taking the ceiling for each terms in Equation 4.1, it happens that we select more than 10 expansion terms for some queries.

With the above proportion determined, we apply our *MMRE* method to select expansion terms iteratively as follows: the number $n(r,Q)$ expansion terms are to be selected from each resource, starting from the most important resource. Each selected expansion term e is assigned a weight which is computed according to Equation 4.2, with the intention to promote expansion terms from highly weighted resources.

$$w(e,Q) = \sum_{r \in R \wedge e^{'} \in E_r(Q)} w(r,Q) \cdot sim_r(e,e^{'}) \tag{4.2}$$

where $E_r(Q)$ is the set of expansion terms that we select from resource r with respect to the query Q and $sim_r(e,e^{'})$ denotes the similarity score between two expansion terms e and $e^{'}$ based on resource r, as we defined before in Section 3.3.2.

The weighted expansion terms are then used to construct a new search query, which is sent to an information retrieval system (such as Indri) to obtain a diversified set of search results. Note that we only select the determined number of expansion terms from each candidate list without performing round diversification at the document level. This is because the candidate lists have already been diversified using *MMRE*. So, selecting the top candidates from each list will naturally result in a diversified set of expansion terms.

4.3.2 Linear Regression Model for Resource Weighting

A simple model of resource weighting is to assign the same weight to all the resources, *e.g.*, $w(r,Q) = \frac{1}{|R|}$. This model totally ignores the contribution differences among resources. Another model is to give a query independent constant weight to each resource, for example, weighting a resource according to the average performance of a SRD system using that resource on all the training queries. This model considers the overall contribution difference among resources, but ignores the differences between individual queries. Here we present a query level resource weighting model based on regression.

First, we characterize the resource weighting task by a set of features. One example feature can

be the number of different expansion candidates generated by a resource (*i.e.*, the number of terms that are judged similar to query terms using the resource). Let x_i denote the i^{th} feature derived from resource query pair (Q,r), and ω_i the weight of the i^{th} feature, then $w(r,Q)$ can be expressed as the weighted combination of all the features plus an offset (denoted by b), as defined in Equation 4.3.

$$w(r,Q) = \sum_i \omega_i \cdot x_i + b \tag{4.3}$$

Then, we learn the feature weights by using Support Vector Regression (SVR) [108], *i.e.*, resolving the following optimization problem as defined in Equation 4.4.

$$argmin_{\omega_i,\xi_{r,Q},\xi^*_{r,Q}}\{\frac{1}{2}\cdot\sum_i \omega_i^2 + C\cdot\sum_{r\in R,Q\in T}(\xi_{r,Q}+\xi^*_{r,Q})\} \tag{4.4a}$$

$$s.t.\begin{cases} w_{r,Q} - w(r,Q) \leq \varepsilon + \xi_{r,Q}, \\ w(r,Q) - w_{r,Q} \leq \varepsilon + \xi^*_{r,Q}, \\ \xi_{r,Q}, \xi^*_{r,Q} \geq 0. \end{cases} \tag{4.4b}$$

where T denotes the queries for training; $w_{r,Q}$ denotes the *ideal* weight of resource r for query Q; the constant C determines the trade-off between the L_2 regularization on the resource weights and the ε-insensitive loss on the observations; ε is the tolerance to errors; $\xi_{r,Q}$ and $\xi^*_{r,Q}$ are slack variables used to cope with infeasible constraints of the optimization problem [108]. These slack variables correspond to the experimental errors of the observation. This optimization problem is convex, and can be efficiently resolved. It is worth noting that the values of the variables that we want to optimize, *i.e.*, ω_i for each feature x_i and $\xi_{r,Q}$ and $\xi^*_{r,Q}$ for each observation query-resource pair, are updated during the sub-gradient process.

For the above linear regression, we need training queries, *i.e.*, the features and the corresponding ideal weight $w_{r,Q}$ of each resource. The training queries correspond to part of the TREC queries available (while the other part is used for testing). To obtain the ideal weights, for each $Q \in T$, we run our method *MMRE*, with all possible resource weights, *i.e.*, $(w_{r_1,Q}, w_{r_2,Q}, \cdots, w_{r_{|R|},Q})$, where $w_{r_i,Q} \geq 0$ and $\sum_{i=1,\cdots,|R|} w_{r_i,Q} = 1$. Then, we select the resource weight sequence that yields the best α-nDCG@20 and consider them as the ground-truth resource weights. In our experiments, we use a

grid search of step 0.05.

4.3.3 Resource Weighting Features

We derive a set of features related to the contribution of resource r for diversifying the search results of query Q. Table 4.III describes all the features, organized into two groups: features common to all resources and resource specific features. These latter are further organized into four categories, depending on the resource they are derived from (Wikipedia, ConceptNet, query logs or feedback documents). It is worth noting that, in case a resource cannot generate a resource specific feature, the value of that feature is set to 0. For example, for the resource feedback documents, we will have 3 resource-nonspecific features and 5 resource-specific features. Other features will have 0 values. Note that resource weights are *independently* learnt by our proposed regression model, *i.e.*, the weight of a resource does not depend on the weights of the other resources (except due to the constraint that they should sum to 1). However, in practice, the weights may not be independent: if we give a high weight to a weak resource, then the stronger resources should have higher weights. To tackle this problem, we perform a normalization of the learnt weights (similar to Equation 4.1) to ensure that the sum of weights of all resources with respect to one query is equal to 1.

*** Resource-Nonspecific Features**

For the features that are common to all resources, we use the number of different candidate expansion terms suggested by each resource (`DiffExpanTerms`), since we believe that the more a resource suggests expansion terms, the more it is likely to cover the different aspects of the query. The average Inverse Document Frequency (`AvgIDF`) of these terms could also be a good indicator of the specificity of expansion terms obtained from each resource.

A new feature that we define in this work is `ContribExpan` ($c(r,Q)$) which denotes the aggregated contributions of all the suggested expansion terms by resource r to the diversity of the search results of a given query Q. In other words, the contribution of a resource regarding to a query is determined by the relation of the expansion terms it suggests with the query and their novelty. A greater $c(r,Q)$ indicates that resource r is more effective to SRD for query Q. $c(r,Q)$ is normalized into $[0,1]$, and meets the constraint that the contribution scores of all considered resources sum up to 1. $c(r,Q)$

Category	Description	Total
***** Resource-nonspecific***		
DiffExpanTerms	Number of different candidate expansion terms suggested by resource r	4
AvgIDF	Average IDF score of the top 10 expansion terms obtained from resource r	4
ContribExpan	Contribution score to Q after being expanded using top 10 expansion terms from resource r	4
***** Resource-specific***		
** Feedback documents:*		
PropFD	Proportion of the feedback documents that contain the terms of Q, computed on F	1
AvgPMI	Average pointwise mutual information score between the terms of Q and the top 10 terms that co-occur a lot with the terms of Q in F	1
ClarityScore	Clarity score of Q computed on F and the whole document collection [38]	1
CoocFreq	Co-occurrence frequency of the query terms computed at window of size 15 on F	1
TFIDF	TFxIDF score of the terms of Q computed on F	1
** Wikipedia:*		
PropWiki	Proportion of the terms of Q having an exact Wikipedia matching page	1
PageRank	PageRank score [95] of the Wikipedia page that matches Q	1
NumInterp	Number of (possible) interpretations of Q in the Wikipedia disambiguation page of Q	1
WikiLength	Wikipedia page length (number of words) that matches with Q	1
** ConceptNet:*		
PropConcep	Proportion of the terms of q that correspond to a node in the graph of ConceptNet	1
NumDiffNodes	Number of different adjacent nodes that are related to the nodes of the graph of Q	1
AvgCommonNodes	Average number of common nodes shared between the nodes of the graph of Q (*i.e.*, nodes that are connected to at least two edges)	1
NumDiffRelations	Number of different relation types defined between the adjacent nodes in the graph of Q	1
** Query logs:*		
PropQL	Proportion of the terms of Q that appear in the query logs	1
NumClicks	Max, Min and average number of clicked URLs for Q in all the sessions	3
PercentageClicks	Percentage of shared clicked URLs between different users who issued Q	1
ClickEntropy	Click entropy of the query Q [50]	1
NumSessions	Total number of sessions with Q	1
SessionLength	Max, Min and average session duration (in seconds) with Q	3
NumTermsReform	Total number of different terms added by users to reformulate Q in all the sessions	1
ReformLength	Max, Min and average number of terms added by users to reformulate Q in all the sessions	3
Grand Total		**39**

Table 4.III: All features computed in this work for automatically weighting resources. (Here, Q denotes an original query, F denotes the set of top 50 retrieval results of Q, and r denotes a resource that could be Wikipedia, ConceptNet, query logs, or feedback documents).

is computed using Equation 4.5:

$$c(r,Q) \propto \frac{1}{|gen_r(Q)|} \sum_{k=1}^{|gen_r(Q)|} c(r,e_k) \tag{4.5}$$

where e_k denotes the k^{th} expansion term for query Q when using resource r, and $gen_r(Q)$ is the set of candidate expansion terms generated using resource r. Following Dang and Croft [42], we use Equation 4.6 to compute the contribution of an expansion term:

$$c(r,e_k) = \max\{0, p(e_k|Q) - \sum_{j=1}^{k-1} p(e_k|e_j)\} \tag{4.6}$$

where $p(e_k|Q)$ represents the individual contribution of e_k to Q, and $p(e_k|e_j)$ denotes the probability of e_k being predicted given e_j, which is estimated based on the co-occurrences between the two terms calculated on the whole document collection. Now, to estimate $p(e_k|Q)$, we divide the computation into two parts [2], as follows:

$$p(e_k|Q) = p(e_k|Q,r) \cdot p(r) \tag{4.7}$$

where $p(r)$ corresponds to the a priori contribution of the resource, which is approximated by the average contribution of resource r on the set of training queries. $p(e_k|Q,r)$ is the importance of expansion term e_k in the query Q, with respect to the resource r, which is estimated as follows:

$$p(e_k|Q,r) = \max_{s \in Q} sim_r(s,e_k) \cdot \frac{|s|}{|Q|} \tag{4.8}$$

where s is a subset of Q, $|s|$ denotes the number of words in s, and $sim_r(s,e_k)$ is the similarity between s and e_k according to r, as described before in Section 3.7.

*** Resource-Specific Features**

Most of the features in this category are straightforward and have been used in previous studies. So we only provide a brief explanation here. The features `PropWiki`, `PropQL`, `PropConcep` and `PropFD` are used to calculate the proportion of query terms that are covered by the resource. We

2. We marginalise $p(e_k|Q)$ over all resources.

observed that the longer the query is (in terms of number of words), the less it is likely to appear in the resource. To tackle this problem, we allow that the resource matches part of the query, but in that case, the corresponding feature value of the resource is proportional to the number of terms in the query it matches. The more a resource matches several parts of the query, the more we have *confidence* on this resource and on the *quality* of expansion terms it suggests.

All the feedback documents-based features are computed on the top 50 results returned for the original query. These features are useful to assess the quality of search results in terms of relevance and diversity, and help to decide whether we should rely on these results. In particular, the clarity score introduced in Cronen-Townsend *et al.* [38] is a good indicator of the ambiguity level of a query. It was shown that the returned search results of an ambiguous query are in general ineffective [38].

For Wikipedia, we use the pages that match with the original query (or a part of the query terms) to derive our features. For example, `PageRank` score [95] is adopted to measure the importance of the Wikipedia pages corresponding to the query: the more important a Wikipedia page is, the more we expect selecting candidate expansion terms from it that are relevant to the query.

On query logs, we develop a number of additional features that are derived from the query reformulations, the click-through data and the query sessions. By investigating the past usage of the original query in the log, one can expect to get candidate expansion terms corresponding to the user intents. For instance, `ClickEntropy` introduced in Dou *et al.* [50] is a good indicator of the amount of variation in the search results searchers click on (*i.e.*, the number of different URLs the users click on), which may be useful to suggest good and diverse candidate expansion terms from the search log data.

Finally, for ConceptNet, we construct a graph for each query, such that the nodes of the graph are those connected to the query terms, from the graph of ConceptNet. The four considered features based on ConceptNet are then computed based on the graph of the query.

4.4 Experiments

In this section, we evaluate our proposed method for query level resource weighting (denoted by *QL-RW* hereafter) for SRD. In particular, we compare our method to uniform resource weighting, which assigns uniform weights to the resources for all queries and which have been used in our

previous studies (see chapter 3) and have shown competitive effectiveness against other state-of-the-art approaches. We also compare our method to non-query level resource weighting, which assigns to each resource a query independent constant proportional to the average contribution of resource for an SRD system on a set of training queries.

4.4.1 Experimental Setup

Data, System and Evaluation Metrics

We use exactly the same experimental setting that we described in Section 3.4. In particular, we consider the same document collection, the same query sets and the same resources. We also evaluate our approaches using the same metrics that we described in Section 2.1.3.

To make a fair comparison with the other baselines, we have also applied the publicly available Waterloo Spam Ranking to the ClueWeb09 (B) collection [3] as described by Cormack *et al.* [35]. The authors in [35] have experimentally shown that spam filtering yields to significant and substantive improvements on the overall results. Following Bendersky *et al.* [7], we consider a spamminess percentile of 60% which is shown to be optimal for the ClueWeb dataset. As we will see, the experimental results on the filtered document collection will be better than on the unfiltered one used in Chapter 3.

Reference Systems and Parameter Setting

For comparison purpose, we consider the following reference systems:
- *BL*, the basic retrieval system, which uses a query generative language model with Dirichlet smoothing (μ=2000), Krovetz stemmer [75], and stopwords removal using the standard INQUERY stopword list;
- *MMR*, the system based on search results re-ranking [22] which trade-off relevance to non-redundancy at the document level;
- *xQuAD*, a probabilistic framework for search result diversification, which explicitly models a query as a set of sub-queries [105].
- *PM-2*, a term-level diversification system [42, 43] that considers aspect popularity;

3. `https://plg.uwaterloo.ca/~gvcormac/clueweb09spam`

We also build the following two reference systems:
- *nQL-RW*, non query level-resource weighting, which assigns to each resource a query independent constant proportional to the average contribution of resource *r* for an SRD system on the whole training queries;
- *U-RW*, uniform resource weighting, which assigns uniform weights to the resources for all queries.
Note that *nQL-RW*, *U-RW*, and *QL-RW* use the same SRD framework (that is *MMRE*), the same resources, and the same parameter settings as described in Section 3.4.4. Besides, for a fair comparison between the three methods, each query is expanded with exactly the same words, but with different weights according to the method. We fix the expansion terms and change their weights in different methods according to the weights of resources. The different weights are assigned to the terms directly. Parameters *C* and the SVM weights in Equation 4.4a, as well as the trade-off parameter of each of the methods *MMRE*, *MMR*, *xQuAD* and *PM-2* are set using 3-fold cross validation: we use in turn each of the query sets from WT09, WT10 and WT11 for test while the other two sets for training. During this procedure, we optimize for α-nDCG@20. To resolve the regression problem described in Section 4.3.2, we directly use SVM-Light tool [4] with option "-z r". Parameter *C* in Equation 4.4a is set to 1.5 using 3-fold cross validation. For the other parameters in SVM-Light, their default values are used in our experiments.

4.4.2 Results

We report the performance numbers in Table 4.IV on queries of WT09, WT10, and WT11, respectively.

From Table 4.IV, we observe that *nQL-RW* performs better than *U-RW*. This shows that a global average weighting is more appropriate than a uniform weighting. We also observe that our method (*QL-RW*) consistently and significantly outperforms the other two reference systems, on both relevance and diversity measures, on almost all datasets. This observation confirms that resource weighting plays an important role in SRD and suggests that resources should be incorporated according to their possible impact on the given query, rather than using query-independent or uniform weights. In Table 4.V, we also report the performance numbers on 144 queries from WT09, WT10, WT11. The set of 144 queries are used because some of the existing methods (*PM-2* [42]) require the queries to

4. `http://svmlight.joachims.org`

Queries	Method	nDCG	ERR	α-nDCG	ERR-IA	NRBP	Prec-IA	S-recall
WT09	U-RW	0.380	0.156	0.367	0.237	0.205	0.155	0.544
	nQL-RW	0.393	0.159	0.386^{U}	0.251^{U}	0.219	0.163	0.587^{U}
	QL-RW	$\mathbf{0.413}^{UN}$	$\mathbf{0.169}^{U}$	$\mathbf{0.428}^{UN}$	$\mathbf{0.274}^{UN}$	$\mathbf{0.243}^{UN}$	$\mathbf{0.172}^{U}$	$\mathbf{0.628}^{UN}$
WT10	U-RW	0.239	0.175	0.391	0.246	0.236	0.219	0.592
	nQL-RW	0.258^{U}	0.179	0.405	0.259^{U}	0.241	0.236^{U}	0.627^{U}
	QL-RW	$\mathbf{0.283}^{UN}$	$\mathbf{0.192}^{U}$	$\mathbf{0.429}^{UN}$	$\mathbf{0.285}^{UN}$	$\mathbf{0.253}^{UN}$	$\mathbf{0.258}^{UN}$	$\mathbf{0.664}^{UN}$
WT11	U-RW	0.371	0.169	0.611	0.522	0.459	0.287	0.802
	nQL-RW	0.387^{U}	0.176	0.629^{U}	0.540^{U}	0.463	0.298	0.821^{U}
	QL-RW	$\mathbf{0.402}^{UN}$	$\mathbf{0.187}^{U}$	$\mathbf{0.657}^{UN}$	$\mathbf{0.575}^{UN}$	$\mathbf{0.476}^{U}$	$\mathbf{0.323}^{UN}$	$\mathbf{0.851}^{UN}$

Table 4.IV: Results of different methods on TREC Web tracks query sets. *U* and *N* indicate significant improvement (p <0.05 in Tukey's test) over *U-RW* and *nQL-RW*, respectively.

exist in the logs and only these 144 queries are in them. We use the same set to make our results comparable.

Method	nDCG	ERR	α-nDCG	ERR-IA	NRBP	Prec-IA	S-recall
BL	0.267	0.133	0.385	0.279	0.241	0.179	0.578
MMR	0.263	0.131	0.387	0.278	0.240	0.179	0.579
PM-2	0.304^{BM}	0.152^{BM}	0.461^{BMX}	0.340^{BMX}	0.308^{BMXU}	0.206^{BM}	0.625^{BM}
xQuAD	0.305^{BM}	0.152^{BM}	0.437^{BM}	0.314^{BM}	0.278^{BM}	0.207^{BM}	0.617^{BM}
U-RW	0.326^{BMXP}	0.169^{BM}	0.451^{BMX}	0.332^{BMX}	0.291^{BM}	0.216^{BM}	0.639^{BMX}
nQL-RW	0.338^{BMXPU}	0.172^{BM}	0.469^{BMXU}	0.347^{BMXU}	0.304^{BMX}	0.229^{BM}	0.667^{BMXPU}
QL-RW	$\mathbf{0.359}^{BMXPU}$	$\mathbf{0.178}^{BMXPU}$	$\mathbf{0.504}^{BMXPUN}$	$\mathbf{0.368}^{BMXPUN}$	$\mathbf{0.317}^{BMXU}$	$\mathbf{0.243}^{BMXPUN}$	$\mathbf{0.703}^{BMXPUN}$

Table 4.V: Comparison of our method with existing SRD methods, on a set of 144 queries from WT09, WT10 and WT11. *B*, *M*, *X*, *P*, *U* and *N* indicate significant improvement ($p < 0.05$ in two-tailed T-test) over *BL*, *MMR*, *xQuAD*, *PM-2*, *U-RW*, and *nQL-RW*, respectively.

From Table 4.V, we observe that our method (*QL-RW*) consistently outperforms existing state-of-the-art SRD approaches (*MMR*, *xQuAD* and *PM-2*) by large margins for most of the relevance and diversity metrics. The improvements are also significant on almost all the measures. This highlights the role that our approach plays and its capability to improve the diversity of search results over the other state-of-the-art methods.

4.4.3 Feature Effects

In this section, we investigate the usefulness of each group of features that we derived in this work. Table 4.VI shows the performance of each group of features, in terms of nDCG@20 and α-nDCG@20, computed on the set of 144 queries. In each row, only features of the corresponding category are selected (*e.g.*, *QL-RW* (Wikipedia) uses only features based on Wikipedia). Recall that *U-RW* uses a uniform weighting and corresponds to the approach with no feature selection.

Feature set	**nDCG**	**α-nDCG**
U-RW	0.326	0.451
QL-RW (resource nonspecific)	0.350	0.493
QL-RW (feedback documents)	0.331	0.471
QL-RW (Wikipedia)	0.338	0.479
QL-RW (ConceptNet)	0.335	0.478
QL-RW (query logs)	0.346	0.489
QL-RW (all features)	**0.359**	**0.504**

Table 4.VI: Performance with different feature sets in terms of nDCG and α-nDCG.

First, we observe that every category of features produces some positive impact on the results, compared to *U-RW*. This highlights the role that our features play. Also, it is clear that considering all features yields larger improvements than using only a single group of features. Second, resource nonspecific features constitute the most robust group of features, yielding the best performance among the groups. In particular, `DiffExpanTerms`, `AvgIDF`, and `ContribExpan` are among the most useful features for improving the overall results. In particular, our feature `ContribExpan` that we introduced in this work has been assigned a high importance. Finally, when comparing the groups of resource specific features, we observe that the features derived from query logs contribute more than the others. A possible reason is that the 144 queries used in this experiment are all well covered by the query logs, which may not be the case for the other resources.

4.4.4 Parameter Sensitivity Analysis

In this work, we set $K = 10$ as the number of expansion terms that we select from each resource. It is interesting to assess the sensitivity of our framework when varying K. We test with $K = 5, 10, 15, 20, 30, 50$, and compare the performance of our method, on the set of WT09 queries.

Here, we only show the results on the set of WT09 queries. We observe similar results for WT10 and WT11 queries. Our results are plotted in Figure 4.1.

We observe that the best performance is reached when $K = 10$ and $K = 15$. When K moves from 5 to 10 and 15, both relevance (nDCG) and diversity (α-nDCG) are improved. This is because, when the number of expansion terms selected from each resource becomes higher, we have a higher chance of finding documents that are relevant, and also covering different aspects of the query. Starting from $K = 20$, the performance of *QL-RW* drops slowly. This could be explained by the fact that our method is likely to select noise expansion terms which are more redundant compared to the previously selected ones, thus hurting the overall performance.

4.4.5 Robustness Analysis

In this section, we analyse the robustness of our framework compared to the other baselines. Following previous studies [42, 43], we define *robustness* as the Win/Loss ratio which is the number of queries that each diversification approach improves (Win) or degrades (Loss) compared to the standard baseline (*BL*), in term of α-nDCG@20. The comparisons are shown in Table 4.VII.

From these results, it is easy to see that our approach is more stable than the other baselines, which means that the improvement that we observe with *QL-RW* is not due to a high improvement over a small set of queries, but because of an improvement on a large number of queries. This suggests that our method can be suitable for a wide range of queries.

Model	**WT09**	**WT10**	**WT11**	**Total**
MMR	16/18	19/15	20/17	55/50
PM-2	25/14	32/10	36/9	93/33
xQuAD	23/16	28/14	29/11	80/41
U-RW	25/11	33/15	37/10	95/36
nQL-RW	25/9	34/11	37/10	96/30
QL-RW	28/10	36/9	38/8	**102/27**

Table 4.VII: Statistics of the Win-Loss ratio of diversification approaches.

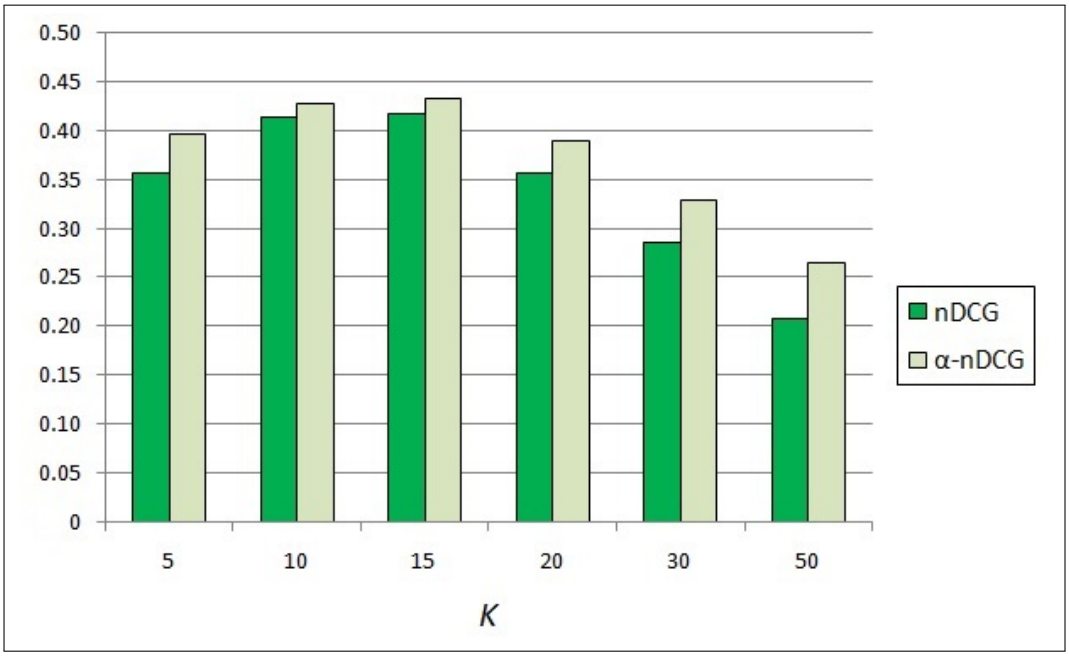

Figure 4.1: Performance of *QL-RW* when varying K, on WT09 queries.

4.4.6 Learnt Resources' Weights vs. Ideal Resources' Weights

An important research question that one should consider in this work, is whether the resources' weights obtained by our method for each query resource pair (r,Q) are comparable to the ideal weights. To answer this question, we consider the set of 144 queries from WT09, WT10 and WT11 query sets, and compute for each query Q the following percentage score:

$$score(Q) = \frac{100}{|R|} \cdot \sum_{r \in R} |w(r,Q) - w_{r,Q}| \tag{4.9}$$

where $w(r,Q)$ (respectively, $w_{r,Q}$) denotes the weight obtained by our method (respectively, the ideal weight) for each query resource pair (r,Q), and R is the set of resources that we consider.

The average score computed on the set of 144 queries is 2.47%. From this result, it is easy to see that the resources' weights computed by our model are very close to the ideal weights that maximize the diversity results. This clearly shows that the linear regression method is powerful enough for resource weighting. It is however possible to use a different regression method and more features.

We leave this to a future work.

4.5 Conclusion

In this chapter, we propose a new query-level resource weighting method in the context of DQE. For that, we develop a regression model enabling us to learn, for each query, the weights of resources based on a set of features. Expansion terms are selected from those suggested by the resources proportionally to the weights of the resources. We evaluated our approach on three topic sets, and using four representative resources. Our results demonstrate the advantage of our method over uniform weighting and non-query level resource weighting.

In this work, we considered four external resources. We believe that other resources could also be effective in our task, such as WordNet, anchor text collections and other resources, from which we can derive additional features for resource weighting. Another aspect where further improvement can be gained is the learning method: instead of using linear regression, other algorithms could be tested, such as those implemented in Weka [5]. These are some interesting work for future studies.

5. http://www.cs.waikato.ac.nz/ml/weka

Chapter 5

Learning Latent Aspects for Search Result Diversification using Multiple Resources

5.1 Introduction

In the previous chapters of this dissertation, we showed that DQE can substantially improve the quality of SRD. While most previous studies on DQE try to select different expansion terms at the word level, no attention has been paid on how well we can cope with the semantic relations between terms. In the previous chapters, we performed implicit SRD, *i.e.*, the query aspects or subtopics are not considered explicitly. There have been studies [105] relying on explicit query subtopics or aspects. However, in most cases, these subtopics are defined manually, which is not realistic. In some studies, query aspects have been extracted automatically [119]. However, the aspects are extracted through document or term clustering. This latter relies on a similarity measure defined on a word-based representation. A critical aspect is that two different terms in such a representation are not comparable, even if they are synonymous or are related to the same query aspect.

In this chapter, we propose a method for DQE relying on an explicit modeling of query aspects, which is defined using word embedding rather than words. Word embedding is trained in a supervised manner according to the principle that related terms (those that are connected in some resource) should correspond to the same aspects. This method allows us to define for each individual query its corresponding aspects that reflect the known semantic relations between terms. We expect that the aspects can correspond loosely to the intended query subtopics, although there is usually not a strict correspondence. Through the latent aspects extracted, we could better select expansion terms so as to cover as much as possible the aspects of a given query. We will experimentally show that our method significantly outperforms other state-of-the-art approaches, and that the explicit modeling of query aspects brings significant gains.

5.2 Problem of Existing DQE Approaches

A typical DQE approach -as we used in the previous chapters- consists of three steps. It first generates a set of expansion term candidates using one or several external resources, *e.g.*, ConceptNet [113], Wikipedia, query logs, or initial feedback documents. Then it selects a set of diverse expansion terms from the candidates, following some principled method. Finally, it combines expansion terms and the original query into one extended query (in which each term has a weight) and uses that query to obtain a set of diversified search results. As subtopics of a query are not explicitly specified in a realistic situation, the DQE approaches try to select diverse expansion terms based on word-level similarities - two terms are assumed to be different if they are not identical or related by some resource. These approaches do not consider how well the expanded query covers different subtopics or aspects of the query. However, a potential problem with such an approach is that an expansion term can appear different from the previous expansion terms, yet it describes exactly the same semantic intent. For example, once the term *library* has been selected as an expansion term for the query "Java", the term *class* could be viewed as an independent one, thus added as an additional expansion term. Yet both expansion terms are related to the same query intent - *Java programming language*.

To be convinced, let's consider the query #78 from the TREC 2010 Web track [31]: "*dieting*". This query is ambiguous and has six subtopics [1] identified by TREC assessors as described in Table 5.I. Table 5.II shows the candidate expansion terms suggested by query logs and outputted using *MMRE*.

Subtopic	*Description*
1	Find "reasonable" dieting advice, that is not fads or medications but reasonable methods for weight loss.
2	Find tips and charts for counting calories while dieting.
3	Find crash diet plans that promise quick weight loss in a short period of time.
4	Find herbal diet supplements and appetite suppressants.
5	Find recommendations for dieting and exercising.
6	Find information on low-carbohydrate diets.

Table 5.I: List of the TREC subtopics for the query "*dieting*".

We observe that some expansion terms selected by *MMRE* appear different from other ones, yet they describe exactly the same semantic intent behind the query. For example, expansion terms *water*,

1. `http://trec.nist.gov/data/web/10/wt2010-topics.xml`

Query logs	**dieting**, plan[3], calories[2], dangers*, water*, body*, benefits*, tea*, grapefruit*, hypothyroidism*, juice*.

Table 5.II: The set of expansion terms selected for the query "*dieting*", from query logs. We manually tag each expansion term by its corresponding TREC subtopic number (from 1 to 6). * means that the expansion term does not clearly correspond to any of the subtopics.

tea and *juice* are viewed to be independent. However, all these terms are about the same semantic query intent: they correspond to different liquors that could be used for dieting. Similarly, expansion terms *dangers* and *hypothyroidism* seem to be very different. Yet, *hypothyroidism* is in fact a kind of body disorder which could be due to an unhealthy dieting [2], hence, *hypothyroidism* could be seen as one of the dangers of dieting. Consequently, both two terms *dangers* and *hypothyroidism* are about the same semantic intent of the query.

The missing element in the previous DQE approaches is an explicit modeling for the underlying *aspects* of the query, with respect to which the selected expansion terms should be diversified. By query aspects, we mean the latent semantic dimensions, similar to topic models in LDA [10], that could be used to describe different query intents/subtopics. However, there is not necessarily an exact match between an aspect and an intent. Diversified expansion terms are thus terms that cover different aspects of the same original query.

In this work, we address this problem by creating an aspect vector space so that each term is mapped to a vector of aspects. The aspects are determined by leveraging the existing resources, which relate different terms by some relations. We assume that two related terms tend to correspond to the same aspect. Therefore, the aspects we will define try to make the known related terms close, and to put unrelated terms apart. In the remainder of this chapter, we provide the details about our method.

5.3 Latent Aspect Embedding

5.3.1 Overview of our Approach

In this work, we propose an approach based on embedding to automatically learn, for each query, its possible aspects. Note that users' queries are very different. For this reason, in this work, we learn

2. http://en.wikipedia.org/wiki/Hypothyroidism

for each individual query its corresponding aspects, *independently* of the other queries. A noticeable difference from previous approaches such as LDA is that in our case the latent aspects are learnt to reflect some known semantic relations between terms (*e.g.*, through existing linguistic resources such as ConceptNet [113] or WordNet [91]), rather than merely to generate the documents (*i.e.* to maximize the likelihood of documents). For example, for query "java", if *programming* and *algorithm* are known to be semantically related (similar), then we would like to create aspects such that these terms can be mapped into the same aspect(s), while *indonesia* will be mapped into a different aspect since it is semantically related neither to *programming* nor to *algorithm* (it corresponds to a different aspect of Java which is *tourism*). In so doing, the created aspects can naturally leverage our knowledge about the semantic relations between terms. Another way to look at our approach is to consider the relations between terms found in different resources as constraints when the latent aspects are generated - Similar terms are constrained to correspond to the same aspects. Such constraints are natural: Without an explicit definition of aspects a priori (which is a difficult task in itself), the best way to define aspects is to rely on the known relations between terms.

A second constraint we impose is that the aspect embedding space should be sparse, *i.e.*, the resulting aspects should be such that a term is associated only with a small number of aspects. This sparsity constraint reflects the fact that the number of subtopics defined manually is usually limited. Without such a sparsity constraint, one would obtain a set of aspects such that each term will be related to a large number of aspects.

Given a query, the expansion terms are selected in turn based on their relation to the initial query, as well as their dissimilarity to the previously selected expansion terms measured according to the aspects. The principle is similar to Maximal Marginal Relevance (*MMR*) [22], but the dissimilarity with the previous expansion terms is measured on the (semantic) aspect level, rather than the (surface) term level.

Our approach relies on an embedding function that maps query expansion terms to aspect vectors for a given query. Similar to *MMRE*, the query expansion terms are generated using a set of heterogeneous resources, each of which provides a means to define semantic similarity. The embedding function is discriminatively trained so that two expansion terms are pushed close in the aspect vector space if they are similar according to some resource. The learning procedure is formulated as an optimization problem similar to matrix factorization [73], in which some task-specific constraints like

sparsity are considered. The optimization problem is then approximately resolved using the standard gradient descent strategy [12]. Once we get an aspect vector for an expansion term for the given query, we can measure the similarity between any two expansion terms according to how much they correspond to the same query aspects, based on which we can remove redundant expansion terms using clustering, *MMR* or other standard approaches. Hereafter, we will first present in detail our aspect embedding framework, then we will describe the similarity functions that we use regarding to each resource that we consider in this work.

5.3.2 Embedding Framework

Similar to latent Dirichlet allocation (LDA) [10], our embedding framework does not need to know the explicit subtopics of the given query, and attempts to obtain a vector for each expansion term with each dimension of the vector representing an implicit aspect of the query. However, different from LDA, our embedding framework uses supervised learning to learn the vectors, and enforces no probabilistic interpretations of the learnt vectors.

Let us assume that we have a set of resources, each suggesting a set of candidate expansion terms for a query and a measure of term similarity. This was already explained in the Section 3.3.2. Let q represent an original query, E_r be the suggested query expansion terms for q using some resource r, and $sim_r(e_i, e_j)$ be the similarity between two expansion terms e_i, e_j from resource r. $sim_r(.,.)$ is a prejudged *local* similarity based on resource r, and used to estimate how well two terms are semantically related according to resource r. Let M_r be the number of expansion terms, *i.e.*, aspect vectors ($M_r = |E_r|$) that we consider for each query (M_r is set to 10 in our experiments) and which are obtained from resource r, and we assume N is the number of dimensions of the aspect space. Our goal is to learn a vector $\overrightarrow{e} =< e^1, e^2, \cdots e^N >$ with its corresponding weight, for any expansion term $e \in E_r$. Here, e^k $(1 \leq k \leq N)$ represents the value of k^{th} dimension of the aspect vector $\overrightarrow{e}$. Let η be a positive scalar which denotes the weight of aspect vector $\overrightarrow{e}$. All the weights of aspect vectors are initialized to $\frac{1}{M_r}$ since we don't want to promote any aspect over the other at the beginning. We denote $sim(\overrightarrow{e_i}, \overrightarrow{e_j})$ the *global* similarity between $\overrightarrow{e_i}$ and $\overrightarrow{e_j}$. Finally, we denote $\overrightarrow{q} =< \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \cdots >$ the constant vector corresponding to the original query q. Each dimension in $\overrightarrow{q}$ is set to $\frac{1}{\sqrt{N}}$ since we don't want to promote any aspect to the other. Besides, such setting ensures that the vector of the query is normalized to 1 (ℓ_2-norm). Note that in this work, the vector of the query is not learned and

stays that constant vector.

If terms e_i, e_j are (strongly) semantically related according to some resource (based on $sim_r(e_i, e_j)$), then our goal is to make closer the two vectors corresponding to these two terms, since they are assumed to correspond to the same aspect of q. In addition, our purpose is also to make the learnt vectors as representative as possible of the original query. These could be addressed by solving the following optimization problem:

$$\begin{aligned} \min_{\overrightarrow{e},\eta_l} \quad & \{ \frac{1}{2} || \sum_{l=1}^{M_r} (\eta_l \cdot \overrightarrow{e_l} - \overrightarrow{q}) ||_2^2 \; + \; \theta \cdot \sum_{e_i,e_j \in E_r, i \neq j} (sim(\overrightarrow{e_i}, \overrightarrow{e_j}) - sim_r(e_i, e_j))^2 \; \} \\ \text{subject to: } & ||\overrightarrow{e}||_2^2 \leq 1; \; e^k \geq 0; \; \eta_l \geq 0; \sum_{l=1}^{M_r} \eta_l = 1; |E_r| = M_r; \\ & k = 1, 2, \cdots, N; \; l = 1, 2, \cdots, M_r; \; \forall e \in E_r. \end{aligned} \tag{5.1}$$

where θ is a parameter that controls the trade-off between the two kinds of loss in Formula 5.1. All the aspect vectors are normalized to 1 (ℓ_2-norm). Note that the objective function in Formula 5.1 is guaranteed to converge towards a minimum since the solution space E_r is usually finite for any query. Given a query q, there is usually a finite number of expansion terms E_r that could be suggested by some resource r regarding to that query. Consequently, $|E_r|$ is finite, which means that there is usually an expansion term in E_r that minimizes our objective function. Our objective function is also general, and could be applied for any resource r provided that the similarity between any pair of expansion terms based on that resource is correctly defined.

The basic idea is that a good aspect representation should satisfy the two following constraints: (i) it makes two known similar terms similar, whatever the resource used to recognize the similarity between them, and (ii) the aspect vectors that we learn should be a good representative of the original query vector (that is $\overrightarrow{q}$). Constraint (i) is satisfied based on the second part of Formula 5.1: we want to push closer in the aspect space two vectors whose corresponding expansion terms are similar according to some resource r. Constraint (ii) is satisfied using the first part of the same formula: we want to learn the weight η_l of each aspect vector $\overrightarrow{e_l}$ ($1 \leq l \leq M_r$) in such a way that the linear combination of these aspect vectors (using the learnt weights) is a good representative of the original query q. When optimizing our objective function described by Formula 5.1, all the aspect vectors and their corresponding weights are updated to satisfy constraint (i) and constraint (ii), simultaneously.

In this work, $\overrightarrow{e}$ is an embedding vector that corresponds to a semantic dimension, similar to topic models in LDA, which could be used to describe different query aspects. An embedding vector $\overrightarrow{e}$ is learnt for each candidate expansion term e. Ideally, each dimension of $\overrightarrow{e}$ is intended to correspond to one possible aspect of the query q that are manually identified, with the value of k^{th} dimension of $\overrightarrow{e}$ representing the association strength of expansion term e to that aspect. However, since the true aspects of the query are unknown, and we do not even know the exact number of these aspects, we can only extract a fixed number N of aspects. In our work, N is experimentally set to 30, which is enough to cover all the aspects for most queries.

Following Koren *et al.* [73], we use the dot product to define the global similarity $sim(\overrightarrow{e_i}, \overrightarrow{e_j})$:

$$sim(\overrightarrow{e_i}, \overrightarrow{e_j}) = \overrightarrow{e_i} \cdot \overrightarrow{e_j} = \sum_{k=1}^{N} e_i^k \cdot e_j^k \tag{5.2}$$

where e_i^k (resp. e_j^k) represents the value of the k^{th} dimension of aspect vector $\overrightarrow{e_i}$ (resp. $\overrightarrow{e_j}$). Note that the ℓ_2-normalization to 1 of any aspect vector $\overrightarrow{e}$ ensures that $sim(\overrightarrow{e}, \overrightarrow{e}) = 1$. This means that the most similar vector to a given aspect vector is the vector itself, which is reasonable. As $\overrightarrow{e_i}$ and $\overrightarrow{e_j}$ are normalized, $sim(\overrightarrow{e_i}, \overrightarrow{e_j})$ is the cosine similarity between $\overrightarrow{e_i}$ and $\overrightarrow{e_j}$. Formula 5.2 encodes our intuition that two vectors corresponding to the same aspect of q should be similar, and their similarity is also proportional to the association strength of each vector to that aspect. For instance, $\overrightarrow{e_3} =< 0.1033; 0.6947; 0; ...; 0.4750 >$ and $\overrightarrow{e_8} =< 0; 0.7122; 0.1966; ...; 0.0948 >$ are two vectors automatically generated by our system for the query "penguins" (query #58 in TREC 2010 Web track), and corresponding to expansion terms *hockey* and *pittsburgh*, respectively. By manually investigating the aspect vectors generated for the query "penguins", we find that each of the first, second, third and last dimensions corresponds to a specific aspect of that query. Despite that the two vectors $\overrightarrow{e_3}$ and $\overrightarrow{e_8}$ share the second and last aspect of the query [3], their similarity according to the second aspect is much higher than the last one. This is because the values involved in the second dimension of each vector are higher than those of the last dimension for each vector. On the other hand, vectors $\overrightarrow{e_3}$ and $\overrightarrow{e_8}$ do not share the first and the third aspect of the query, despite the fact that $\overrightarrow{e_3}$ (resp. $\overrightarrow{e_8}$) has a non-zero value at dimension 1 (resp. dimension 3). This is realistic, because when one of the two vectors has no connection with an aspect of the query, the two vectors should never be considered

3. This is because both vectors $\overrightarrow{e_3}$ and $\overrightarrow{e_8}$ have non-zero values at the second and last dimensions.

similar according to that aspect.

In practice, there is not a single universal resource that covers all semantic relations. Since multiple resources tend to complement each other, for DQE, integrating multiple resources can often yield substantial improvements compared to using one single resource [8, 14, 47, 48]. Using multiple resources, the above Formula 5.1 is extended as follows:

$$\begin{aligned} \min_{\overrightarrow{e},\eta_l,\omega_r} \quad & \{ \frac{1}{2} || \sum_{l=1}^{M} (\eta_l \cdot \overrightarrow{e_l} - \overrightarrow{q}) ||_2^2 \ + \ \theta \cdot \sum_{r \in R} \sum_{e_i,e_j \in E, i \neq j} \omega_r \cdot (sim(\overrightarrow{e_i}, \overrightarrow{e_j}) - sim_r(e_i,e_j))^2 \ \} \\ \text{subject to: } & ||\overrightarrow{e}||_2^2 \leq 1; \ e^k \geq 0; \ \eta_l \geq 0; \sum_{l=1}^{M} \eta_l = 1; |E| = M; \omega_r \geq 0, \ \forall r \in R; \ \sum_{r \in R} \omega_r = 1; \\ & k = 1,2, \cdots, N; \ l = 1,2,\cdots,M; \ \forall e \in E. \end{aligned} \tag{5.3}$$

where $R = \{r_1, r_2, \cdots, r_m\}$ denotes a set of resources, $E = \bigcup_r E_r$ means all expansion terms; $\omega_r \geq 0$ is the weight of resource r, and M is the total number of expansion terms (*i.e.*, aspect vectors) that we consider from the different resources, *i.e.*, $M = \sum_{r \in R} M_r = |E|$. In our experiments, $M = 40$.

We use gradient descent to resolve the optimization problem (defined in Formula 5.3). This optimization algorithm has one desirable property: when the learning rate is small enough, it is guaranteed to converge towards a minimum of the loss function defined by Formula 5.3 [12]. We iteratively update the aspect vectors and the resources' weights using the gradient descent rule until we observe no significant updates of the gradients with respect to all the aspect vectors. $\frac{||\nabla_i^{(t+1)} - \nabla_i^{(t)}||_2^2}{||\nabla_i^{(t)}||_2^2} < 0.0001, \forall e_i \in E$, where $\nabla_i^{(t)}$ means the gradient with respect to $\overrightarrow{e_i}$ after the t^{th} iteration. More specifically, during each iteration, we first compute the associated prediction error for each given training case e_i, e_j from resource r:

$$loss_{ij}^r = \omega_r \cdot (sim(\overrightarrow{e_i}, \overrightarrow{e_j}) - sim_r(e_i,e_j))^2 \tag{5.4}$$

Then the gradient of the loss function (Formula 5.3) with respect to vector $\overrightarrow{e_i}$ and resource r can be determined using Formula 5.5 and Formula 5.6, respectively.

$$\nabla_i = \sum_{r \in R} \sum_{e_j \in E, i \neq j} loss_{ij}^r \cdot \overrightarrow{e_j} + \eta_i \cdot \sum_{l=1}^{M} (\eta_l \cdot \overrightarrow{e_l} - \overrightarrow{q}) \tag{5.5}$$

$$\nabla_r \propto \sum_{e_i,e_j \in E_r, i \neq j} \frac{1}{2} \cdot (sim(\overrightarrow{e_i}, \overrightarrow{e_j}) - sim_r(e_i, e_j))^2 \tag{5.6}$$

In Formula 5.4, the associated prediction error of term e_i is computed regarding to each expansion term $e_j \in E, j \neq i$. In case e_i and e_j do not appear simultaneously in the same resource r, then $sim_r(e_i, e_j)$ in Formula 5.4 is set to 0. In Formula 5.6, ∇_r refers to the proportion of the error due to resource r. Formula 5.6 encodes our intuition that, if a resource is responsible for a large part of the error, then its weight ω_r should be updated largely, and vice versa. Subsequently, we update both $\overrightarrow{e_i}$ ($\forall e_i \in E_r$) and ω_r by a magnitude proportional to γ in the opposite direction of the gradient, yielding these two gradient descent updating rules:

$$\overrightarrow{e_i} \leftarrow \overrightarrow{e_i} - \gamma \cdot \nabla_i \tag{5.7.a}$$

$$\omega_r \leftarrow \omega_r - \gamma \cdot \nabla_r \tag{5.7.b}$$

where γ is the learning rate, which we fix at 0.001 as suggested by both Koren *et al.* [73] and Johnson and Zhang [69].

To ensure that constraints in Formula 5.3 hold during training, we initialize ω_r to $\frac{1}{|R|}$ and we set negative e_i^k and negative ω_r to zero and re-normalize the vectors each time after Formula 5.7.a and 5.7.b are applied. The reason of setting negative e_i^k to zero is as follows: In our objective function described in Formula 5.1 and 5.3, we enforced the constraint $e^k \geq 0$ since e^k is the association strength of expansion term e to the k^{th} aspect of the query, which should be a positive value. Note that, after each iteration, we normalize $\overrightarrow{e_i}$ according to the first constraint of our objective function (*i.e.*, $||\overrightarrow{e}||_2^2 \leq 1$) ensuring that $e^k \leq 1$. Similarly, we set negative ω_r to zero since the weight of a resource could not be negative ($\omega_r \geq 0$). Note that the update of ω_r (using Formula 5.7.b) and its normalization (*i.e.*, $\sum_{r \in R} \omega_r = 1$) are made once all aspects have been updated.

It is worth noting that ω_r reflects the contribution of each resource r in the calculation of the similarity score between a pair of terms. The idea is that, we want to promote resources that contribute more to the similarity calculation between terms, by assigning them high weights. By doing so, we can benefit from resources that contain semantic similarities between expansion terms.

Finally, once all the aspect vectors $\overrightarrow{e_i}$ were updated based on Formula 5.7.a, we update now their

corresponding weights η_i using the following Formula:

$$\eta_i \leftarrow \eta_i - \gamma \cdot \overrightarrow{e_i}^T \cdot \sum_{l=1}^{M} (\eta_l \cdot \overrightarrow{e_l} - \overrightarrow{q}) \tag{5.8}$$

where $\overrightarrow{e_i}^T$ denotes the transpose of the vector $\overrightarrow{e_i}$. We also set negative η_i to zero for each $\overrightarrow{e_i}$ and re-normalize all the weights of aspect vectors (*i.e.*, $\sum_{i=1}^{M} \eta_i = 1$).

The minimum of the loss function defined in Formula 5.3 depends on how the vectors are initialized. We have tried two methods to initialize a vector: 1) assigning each dimension with a random number while forcing the constraint $||\overrightarrow{e}||_2^2 \leq 1$; and 2) setting each dimension to $\frac{1}{\sqrt{N}}$, without promoting any aspect to the other. We adopt the second method which experimentally works better than the first one. Algorithm 5.1 describes the working scheme of our proposed embedding framework.

According to our investigation, an expansion term usually covers only a few aspects of the query. This inspires us to consider the sparsity constraint on each aspect vector $\overrightarrow{e}$ that we want to learn, *i.e.*, only a few dimensions of $\overrightarrow{e}$ are non-zero. Following Donoho *et al.* [49], we achieve this by incorporating ℓ_1-norm [4] penalization into Formula 5.3, yielding:

$$\begin{aligned}
&\min_{\eta_l, \overrightarrow{e}, \omega_r} \ \{ \ \frac{1}{2} \ || \sum_{l=1}^{M} (\eta_l \cdot \overrightarrow{e_l} - \overrightarrow{q}) ||_2^2 \ + \ \theta \cdot \sum_{r \in R} \sum_{e_i, e_j \in E, i \neq j} \omega_r \cdot (sim(\overrightarrow{e_i}, \overrightarrow{e_j}) - sim_r(e_i, e_j))^2 + \phi \cdot \sum_{l=1}^{M} ||\overrightarrow{e_l}||_1 \ \} \\
&\text{subject to: } ||\overrightarrow{e}||_2^2 \leq 1; \ e^k \geq 0; \ \eta_l \geq 0; \sum_{l=1}^{M} \eta_l = 1; |E| = M; \omega_r \geq 0, \ \forall r \in R; \ \sum_{r \in R} \omega_r = 1; \\
&\qquad k = 1, 2, \cdots, N; \ l = 1, 2, \cdots, M; \ \forall e \in E.
\end{aligned} \tag{5.9}$$

Here ϕ controls the trade-off between two kinds of losses. Note that the objective function (5.9) is non-differentiable at points $\overrightarrow{e}$ with any $e^k = 0$. We therefore use sub-gradient to attack this problem

4. $||\overrightarrow{e}||_1 = \sum_{k=1}^{N} |e^k|$

Embedding Framework [q, R, M_r, N, γ, θ]

$E \leftarrow \emptyset$
for $r \in R$ **do**
  $E_r \leftarrow$ { top M_r candidate expansion terms that are relevant to q }
  $\omega_r \leftarrow \frac{1}{|R|}$; $E \leftarrow E \bigcup E_r$
end for
$A \leftarrow \emptyset$ *//Set of learnt aspect vectors*
$nbV \leftarrow 0$ *//Computes the total number of vectors having no significant updates of the gradients*
$t \leftarrow 0$
for $e_i \in E$ **do**
  Map expansion term e_i with an aspect vector $\overrightarrow{e_i}$
  Initialize $\overrightarrow{e_i} =< \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \cdots >$
  $A \leftarrow A \bigcup \{\overrightarrow{e_i}\}$; $\nabla_i^{(t)} \leftarrow 1$
end for
do
  $nbV \leftarrow 0$; $t \leftarrow t+1$
  for $e_i \in E$ **do**
    for $e_j \in E, j \neq i$, **do**
      for $r \in R$ **do**
        compute $loss_{ij}^r$ using Formula 5.4
      end for
    end for
    compute $\nabla_i^{(t)}$ using Formula 5.5
    update $\overrightarrow{e_i}$ using Formula 5.7.a
    for k **from** 1 **to** N **do**
      if $(e_i^k < 0)$ **then** $e_i^k \leftarrow 0$
      end if
    end for
    if $(\frac{||\nabla_i^{(t)} - \nabla_i^{(t-1)}||_2^2}{||\nabla_i^{(t-1)}||_2^2} < \gamma)$ **then** $nbV \leftarrow nbV + 1$
    end if
  end for
  for $e_i \in E$ **do**
    update η_i using Formula 5.8
    if $(\eta_i < 0)$ **then** $\eta_i \leftarrow 0$
    end if
  end for
  for $e_i \in E$ **do**
    normalize η_i satisfying $\sum_i \eta_i = 1$
  end for
  for $e_i \in E$ **do**
    compute ∇_r using Formula 5.6 and update ω_r using Formula 5.7.b
    if $(\omega_r < 0)$ **then** $\omega_r \leftarrow 0$
    end if
  end for
  for $r \in R$ **do**
    normalize ω_r satisfying $\sum_r \omega_r = 1$
  end for
while $(nbV < |E|)$
return { A, η_i ; $1 \leq i \leq |E|$ }

Figure 5.1: The embedding framework.

[107]. First, we compute the sub-gradient with respect to e_i^k:

$$\nabla_i^k = \begin{cases} loss_i^k + \phi \cdot sign(e_i^k) & \text{if} \quad e_i^k \neq 0 \\ loss_i^k + \phi & \text{if} \quad e_i^k = 0, loss_i^k < -\phi \\ loss_i^k - \phi & \text{if} \quad e_i^k = 0, loss_i^k > \phi \\ 0 & \text{if} \quad e_i^k = 0, -\phi \leq loss_i^k \leq \phi \end{cases} \tag{5.10}$$

where $loss_i^k$ is defined as follows:

$$loss_i^k = \eta_i \cdot (\sum_{l=1}^{M} \eta_l \cdot \overrightarrow{e_l} - \overrightarrow{q})^k \cdot e_i^k + \theta \cdot \sum_{r \in R} \sum_{e_j \in E_r, i \neq j} loss_{ij}^r \cdot e_j^k \tag{5.11}$$

Then we use sub-gradient to replace gradient in Formula 5.7.a, obtaining the following update rule for each iteration:

$$e_i^k \leftarrow e_i^k - \gamma \cdot \nabla_i^k \tag{5.12}$$

For the updating values of ω_r and η_i, we use the same rules as described by Formula 5.7.b and Formula 5.8, respectively.

Finally, note that the working scheme of our embedding framework with the sparsity constraint is very similar to that described in Algorithm 5.1, with the differences that:
(1) The loss functions used in line 19 of Algorithm 5.1 should be replaced with the loss function described in Formula 5.11, which should be computed for each dimension (e_i^k) of each aspect vector $\overrightarrow{e_i}$;
(2) The gradient descent in line 22 of Algorithm 5.1 should be replaced by the sub-gradient with respect to each dimension (e_i^k) of each aspect vector $\overrightarrow{e_i}$, as defined by Formula 5.10;
(3) The update of each aspect vector $\overrightarrow{e_i}$ in line 23 of Algorithm 5.1 should be replaced by the update of each dimension (e_i^k), as described by Formula 5.12.

5.3.3 SRD with Embedding

Given a query q, any term related to its terms through a resource r is considered as a candidate expansion term. From each resource, we select a subset of expansion terms which are relevant to the query. Once each expansion term is mapped to an aspect vector, we first apply our embedding frame-

work described in Section 5.3.2 to learn the aspect vectors for q. Then, to generate diversified search results, we run Maximal Marginal Relevance-based Expansion (*MMRE*) to obtain a global list of diversified aspect vectors, with the goal of removing redundancy among expansion terms while covering as many aspects of the original query as possible. The global similarity between two embedding vectors (*i.e.*, two expansion terms) is computed using Formula 5.2.

Now, to compute the relevance of an embedding vector to the original query (which is exactly the global similarity between the query and the expansion term corresponding to that vector), we simply re-use the dot product, as defined by Formula 5.13:

$$sim(\overrightarrow{e}, \overrightarrow{q}) = \overrightarrow{e} \cdot \overrightarrow{q} = \sum_{k=1}^{M} e^k \cdot q^k \tag{5.13}$$

where $\overrightarrow{e}$ (resp. $\overrightarrow{q}$) is the vector corresponding to expansion term e (resp. query q), and e^k (resp. q^k) is the value of k^{th} dimension of $\overrightarrow{e}$ (resp. $\overrightarrow{q}$).

By combining Formula 5.2 and 5.13, we obtain the formal definition of the *MMRE* procedure:

$$\overrightarrow{e^*} = \arg max_{e \in E} \{\beta \cdot sim(\overrightarrow{e}, \overrightarrow{q}) - (1-\beta) \cdot \max_{\overrightarrow{e'} \in ES} sim(\overrightarrow{e}, \overrightarrow{e'})\} \tag{5.14}$$

Here, ES represents the expansion terms already selected; $\beta \in [0,1]$ controls the trade-off between relevance and redundancy of the expansion terms (which will be set using validation data).

We iteratively apply the *MMRE* procedure described in Formula 5.14 to select a set of diversified aspect vectors, thus leading to a set of diversified expansion terms for q. At the end, we keep K expansion terms (K is set to 20 in our experiments). Note that, for some queries, different resources may suggest the same expansion terms. In the case of *multiple copies* of the same term, once a copy of a term has been selected, the other copies will be discarded (because they are similar) due to the non-redundancy component of *MMRE* (see Formula 5.14).

The selected expansion terms are then combined with the initial query to formulate a new query, in which each term e is weighted by $sim(\overrightarrow{e}, \overrightarrow{q})$. By submitting this query to a retrieval system (*e.g.*, Indri), we finally obtain a set of diversified search results for the original query. Notice that the retrieved results are not processed by any additional document selection process (such as *MMR* [22] or *xQuAD* [105]) for further diversification, although this is possible. In Section 5.5.4, we will

investigate in more details the impact of diversifying the search results after diversifying the query, and we will show that a further step of diversifying search results does not improve the retrieval results of a query whose expansion terms were already diversified.

Finally, note that $sim(\overrightarrow{e}, \overrightarrow{e'})$ and $sim(\overrightarrow{e}, \overrightarrow{q})$ play a center role in our system, which are both computed on top of the aspect vectors learnt with our embedding framework. This is in sharp contrast with previous studies in chapter 3 in which use conventional term similarity measures based on different resources without considering the query and query aspects. Our method gives us a clear advantage: the selected expansion terms will tend to cover different query aspects. This advantage will be confirmed in our experiments.

5.4 Experimental Setup

In this section, we will conduct several experiments which aim to answer the four following research questions:

1. Is our embedding proposed approach effective at improving search results in terms of both relevance and diversity, compared to the state-of-the-art approaches?
2. What is the impact of the sparsity constraint on the performance of the whole framework?
3. Do we need to further diversify the search results after diversifying the query?
4. What is the robustness of our framework?

The first three research questions will be addressed in Section 5.5 in which we run extensive experiments on TREC diversification data to evaluate our approach and compare it to other existing methods, as well as the impact of SRD on DQE. Section 5.6 will be mainly dedicated to answer our fourth research question. In the remainder of this section, we detail the document collection, the used resources, the topics, and the metrics used to evaluate our work. Besides, we describe the baselines and diversification frameworks with which we will compare our approach, as well as the training procedure to set the different parameters.

Datasets and Evaluation Metrics

We use exactly the same experimental setting that we described in Section 3.4. In particular, we consider the same document collection, the same query sets and the same four resources. We also evaluate our approaches based on the same metrics that we described in Section 2.1.3.

To make a fair comparison with the other baselines, we have also applied the publicly available Waterloo Spam Ranking to the ClueWeb09 (B) collection [5] as described by Cormack *et al.* [35], and we consider a spamminess percentile of 60% which is shown to be optimal for the ClueWeb dataset [7].

Baselines and Diversification Frameworks

We compare our embedding system with the following systems:
- *BL*, the basic retrieval system, which is built with Indri and is based on a query generative language model with Dirichlet smoothing (μ=2000), Krovetz stemmer [75], and stopwords removal using the standard INQUERY stopword list;
- *MMR*, the system based on search results re-ranking [22];
- *PM-2*, a term-level diversification system [42, 43] that considers aspect popularity;
- *xQuAD*, a probabilistic framework for search result diversification, which explicitly models an ambiguous query as a set of sub-queries [105].

Hereafter, we denote by *eRS* our embedding framework with resource weighting and the sparsity constraint. To further study the effectiveness of all the core components of our system, we build two reference systems: *eR* and *Comb*. *eR* is an embedding system based on Formula 5.3, which ignores the sparsity constraint; *Comb* is the model that we proposed in Section 3.4.3 which uniformly combines different resources. Given a query q, *Comb* combines different sets of retrieved documents, each with an expanded query using $MMRE_r$ with resource r. We choose to compare with this method as it also uses multiple resources and it has been found to be effective.

As we have pointed out before, our approach is different from LDA in that our aspects are created by leveraging the term relations in four resources. To show the benefit of doing so, we build another reference system which expands an original query with the set of topics which are obtained

5. https://plg.uwaterloo.ca/~gvcormac/clueweb09spam

by applying LDA to the top 50 documents returned for the original query. Hereafter, we use QE_{LDA} to denote this system. Note that QE_{LDA} is similar to the method presented of Vargas *et al.* [119] with the difference that their method selects expansion terms from groups of documents that cover the same query subtopic.

As we mentioned earlier, the work of He *et al.* [61] is similar to ours, which also uses external resources for the purpose of SRD. Recall that in He *et al.* [61] the Multi-Search Subtopics (*MSS*) are created based on random walks on three resources, namely click logs, anchor texts and Web n-grams. We reimplement *MSS* with different resources - the four resources we described. Hereafter, we denote this method by MSS_{modif}. Similarly to He *et al.* [61], we also define a graph-based structure for each resource that we consider for MSS_{modif}. For query logs, we use the same graph representation described in [61]. For Wikipedia, each node in the graph corresponds to one Wikipedia page, and two nodes are connected if they share at least one anchor text. For the feedback documents, each node corresponds to one term from the top 50 returned documents of a given query, and two nodes (terms) are connected if they co-occur in the same window size (in our experiments, we fix our window size to 15). Note that ConceptNet is already a graph-based representation which encompasses nodes (concepts) that are connected together [113].

Finally, as we already mentioned in Section 2.5, the work that we describe in this chapter is similar to our previous work [84] in which, we also used embedding to learn query aspects for the purpose of better diversifying the results and introduce *compact aspect embedding* for DQE. However, these two methods are trained in different ways: Our method is trained in a supervised manner according to the principle that related terms should correspond to the same aspects, while in the method described in [84], we exploit trace norm regularization to learn a low rank vector space for the query. Hereafter, we call this approach *CompAE*. In [84], we used one resource (query logs) to select candidate expansion terms. For a fair comparison, we compare *CompAE* with *eRS* when using only query logs.

Parameter Setting

Our model and our considered baselines and diversification frameworks come with a number of parameters. Firstly, for γ (the learning rate), we follow both Koren *et al.* [73] and Johnson and Zhang [69] and set it to 0.001.

The other parameters are determined using 3-fold cross validation. We use in turn each of the

query sets from WT09, WT10 and WT11 for test while the other two sets for training. During this procedure, we optimize for α-nDCG@20. θ the trade-off parameter of two types of loss in Formula 5.1 and ϕ, the trade-off parameter of two types of loss in Formula 5.9 are set using random search [9]. For that, we consider values of θ and ϕ in the range of [0.1, 1], and apply a sampling process to generate 1000 subsets from the interval [$log(0.1)$, $log(1)$]. Then, we randomly select 50 values of θ (resp. ϕ) from the sampling subsets and we consider the average of these 50 values of θ (resp. ϕ) as the optimal value of θ (resp. of ϕ).

Each of the methods *MMR*, *xQuAD* and *PM-2* has one parameter λ to be tuned. We consider values of λ in [0, 1] with an increment of 0.1. All the parameters involved in MSS_{modif} are set according to He *et al.* [61].

The remaining free parameters for the model proposed in this work are the following: N, the number of dimensions of aspect embeddings; K, the number of expansion terms that we consider for each query at the end; M_r the number of expansion terms that we keep from each resource r; and β, which is the trade-off parameter of the *MMRE* procedure that selects expansion terms from multiple resources, according to Formula 5.14. To optimize these parameters' values, we use *coordinate ascent search* technique [88]: β in the range of [0.1, 1] with an increment of 0.1, and the others (N, K and M_r) in the range of {5, 10, 15, ..., 50}.

5.5 Experimental Results

In this section, we aim to answer our first three research questions. In particular, we will answer the first question by investigating the impact of our approach on result diversification, and its effectiveness compared to existing works. Then, to answer our second question, we will show the impact of the sparsity constraint on the whole performance of our framework. Finally, the last sub-section will be dedicated to answer our third question.

Table 5.III and Table 5.VI report the performance numbers on queries of WT09, WT10, WT11 and on a set of 144 queries, respectively. The set of 144 queries are used because some of the existing methods, namely *PM-2* [42], require the queries to exist in the query logs and only these 144 queries are in them.

From these two tables, we can observe that our approach *eRS* consistently outperforms all the other

Queries	Model	nDCG	ERR	α-nDCG	ERR-IA	NRBP	Prec-IA	S-recall
WT09	*BL*	0.312	0.125	0.297	0.195	0.162	0.111	0.430
	MMR	0.310	0.119	0.296	0.191	0.161	0.120	0.442
	Comb	0.392*-	0.153-	0.374*-	0.235*-	0.212*-	0.154*-	0.549
	eR	**0.422***-+	**0.179***-+	**0.436***-+	**0.279***-+	**0.258***-+	**0.194***-+	**0.673***-+
	eRS	**0.451***-+♮	**0.198***-+	**0.474***-+♮	**0.293***-+	**0.275***-+	**0.198***-+	**0.709***-+♮
WT10	*BL*	0.182	0.139	0.320	0.203	0.163	0.170	0.543
	MMR	0.191	0.142	0.329	0.213	0.170	0.172	0.562
	Comb	0.244*-	0.171*-	0.390*-	0.243*-	0.223*-	0.212*-	0.594*-
	eR	**0.294***-+	**0.207***-+	**0.450***-+	**0.302***-+	**0.278***-+	**0.284***-+	**0.692***-+
	eRS	**0.314***-+♮	**0.221***-+♮	**0.462***-+♮	**0.319***-+	**0.290***-+	**0.294***-+	**0.733***-+♮
WT11	*BL*	0.298	0.139	0.542	0.440	0.399	0.240	0.764
	MMR	0.304	0.141	0.544	0.433	0.397	0.250	0.741
	Comb	0.377*-	0.161*	0.612*-	0.509*-	0.440*-	0.279*	0.782-
	eR	**0.416***-+	**0.192***-	**0.676***-+	**0.600***-+	**0.508***-+	**0.344***-+	**0.866***-+
	eRS	**0.434***-+	**0.217***-+♮	**0.692***-+♮	**0.628***-+♮	**0.527***-+♮	**0.371***-+♮	**0.907***-+ ♮

Table 5.III: Experimental results of different models on TREC Web tracks query sets. *, -, + and ♮, indicate significant improvement ($p < 0.05$ in Tukey's test) over *BL*, *MMR*, *Comb*, and *eR*, respectively.

systems in terms of both relevance and diversity on all data sets, and in most cases the improvements are statistically significant. This observation confirms the overall advantage of our proposed system. In particular, in Table 5.VI, comparing our approaches with other state-of-the-art approaches, we can see that our method outperforms *xQuAD* and *PM-2* by large margins. It also outperforms MSS_{modif} in most of the measures, and the differences are in general statistically significant. We will analyze in more detail these results in the remainder of this section.

5.5.1 Effectiveness of Latent Aspect Embedding

In Table 5.III, *eRS* and *eR* are the two methods that use aspect embedding trained using supervised learning. The counterpart method that uses the same resources without aspect embedding is *Comb*. Recall that term similarity in the latter is obtained directly from the similarity functions defined for different resources. We can see clearly that *eRS* and *eR* outperform *Comb* significantly on all the measures and for all the query sets. This is a clear indication of the advantage of using aspect embedding to represent the possible query intents and to determine the appropriate expansion terms accordingly. Notice that we also tested *Comb* with non-uniform weights for the four resources, but the results are

generally similar. For brevity, we do not report this case.

Let us show the impact on one particular query "cell phones" in WT09, which is a typical example showing the general trends (Figure 2.1 above shows the query and its subtopics, as identified by TREC assessors). Table 5.IV shows the candidate expansion terms suggested by different resources and outputted using *eR* and *eRS*, respectively. In our experiments, we always add the original query terms to the expansion term list for any resource. Recall that E_C, E_W, E_Q and E_D corresponds to the set of expansion terms suggested by ConceptNet, Wikipedia, query logs and feedback documents, respectively, for the same original query "cell phones". From Table 5.IV, we notice that different resources suggest some common terms, *e.g.*, "free", "sale", "smartphone", "ericsson", "nokia", and "service", and that terms such as "nokia", "motorola", "apple" are actually semantically related to the same aspect. This observation confirms the redundancy among expansion terms that are generated using different resources. Interestingly, we find that "apple" (in E_C) is discarded by *eRS* and *eR*. This can be explained by two reasons: 1) "apple", unlike "nokia" or "motorola", is ambiguous, and thus with low relevance degree to the query; and 2) it is close to "iphone" and "company" in the aspect vector space, which are highly related to the query.

When we use *MMRE*, expansion terms are selected based on their surface dissimilarity (term similarity is measured at the term level rather than at aspect level). This explains why in the above example, E_F and E_L select "ericsson" and "motorola", respectively, which refer to the same aspect of the query as "smartphone", a term they has already been selected. In contrast, in *eRS* and *eR*, term dissimilarity considers whether these terms are related to the same aspects. Once the term "smartphone" has been selected, the other phone brands ("ericsson", "motorola") are selected much later, after the selection of terms related to other aspects. This example indicates that the similarities based on the aspect vectors ($sim(\overrightarrow{e}, \overrightarrow{e'})$ and $sim(\overrightarrow{e}, \overrightarrow{q})$) can effectively remove redundant expansion terms covering the aspects that have already been covered. This clearly shows the benefit of using aspects in our approach - the selected expansion terms have a better coverage of different query aspects. This effect is similar to that of the proportionality of subtopics in *PM-2* [42, 43], which forces the selection of terms on aspects that are insufficiently covered by the terms already selected. However, in the case of *PM-2*, manually defined subtopics are required, while our method does not need them.

For a better understanding of the output of our model, we provide in Figure 5.5.1 a visualization

Model	Expansion Terms (in decreasing order of importance)
E_C	**cell**, **phones**, apple, vendor, free, verizon, battery, service, gps, sale, camera, storage
E_W	**cell**, **phones**, mobile, iphone, company, sprint, motorola, prepaid, nokia, service, smartphone, sale
E_L	**cell**, **phones**, unlocked, smartphone, motorola, buy, verizon, information, sprint, nokia, sale, ericsson
E_F	**cell**, **phones**, buy, information, product, unlocked, popular, smartphone, ericsson, free, accessory, vendor
eR	**cell**, **phones**, accessory, prepaid, smartphone, sprint, information, product, sale, popular, nokia, vendor, option, battery, verizon, storage, motorola, service, company, camera, buy, ericsson
eRS	**cell**, **phones**, sprint, accessory, prepaid, smartphone, camera, sale, iphone, product, free, option, nokia, service, buy, motorola, popular, vendor, information, unlocked, company, verizon

Table 5.IV: Expansion terms for "cell phones" generated by using different resources and outputted by *eR* and *eRS*, respectively. E_C, E_W, E_L, and E_F denote the expansion terms obtained using ConceptNet, Wikipedia, query logs and feedback documents, respectively. Different colors represent different aspects of the query.

of the query vector, and the aspect embedding vectors learnt by *eRS* for the same example query "cell phones" (we only show some of these aspects for illustration). For that, we used *Vector Visualizer in 3D*, which is an online free tool for visualizing vectors in three dimensions [6]. Since the number of dimensions of aspect embedding (parameter N) is high (N=30 in our experiments), we applied PCA (Principal Component Analysis) technique [70] to reduce the dimension of our vectors to 3. To do that, we use *XLSTAT* [7] which is a software that could be coupled with MS Excel as a supplementary module and provides facilities to analyzing data and running statistics on them, such as dimensionality reduction, clustering, logistic regression.

From Figure 5.5.1, it is clear that our proposed approach can select terms from different aspects by ensuring a good coverage of the different query subtopics. More interestingly, our approach succeeds to group together similar expansion terms that share the same semantic aspect, by pushing closer in the space the vectors corresponding to the same semantic aspect of the original query. For instance, aspect vectors mapped to terms like "nokia", "motorola" and "smartphone", respectively, appear very close in the semantic space of the query since they correspond to the same aspect of "cell phones" which is *phone brands*. Also, note that terms corresponding to the same semantic aspect do not appear successively together in the expanded query obtained by *eR* or *eRS* (see Table 5.IV), due to the non-redundancy component that we consider in both two approaches. For example, in the expanded query

6. `http://www.bodurov.com/VectorVisualizer`
7. `http://www.xlstat.com`

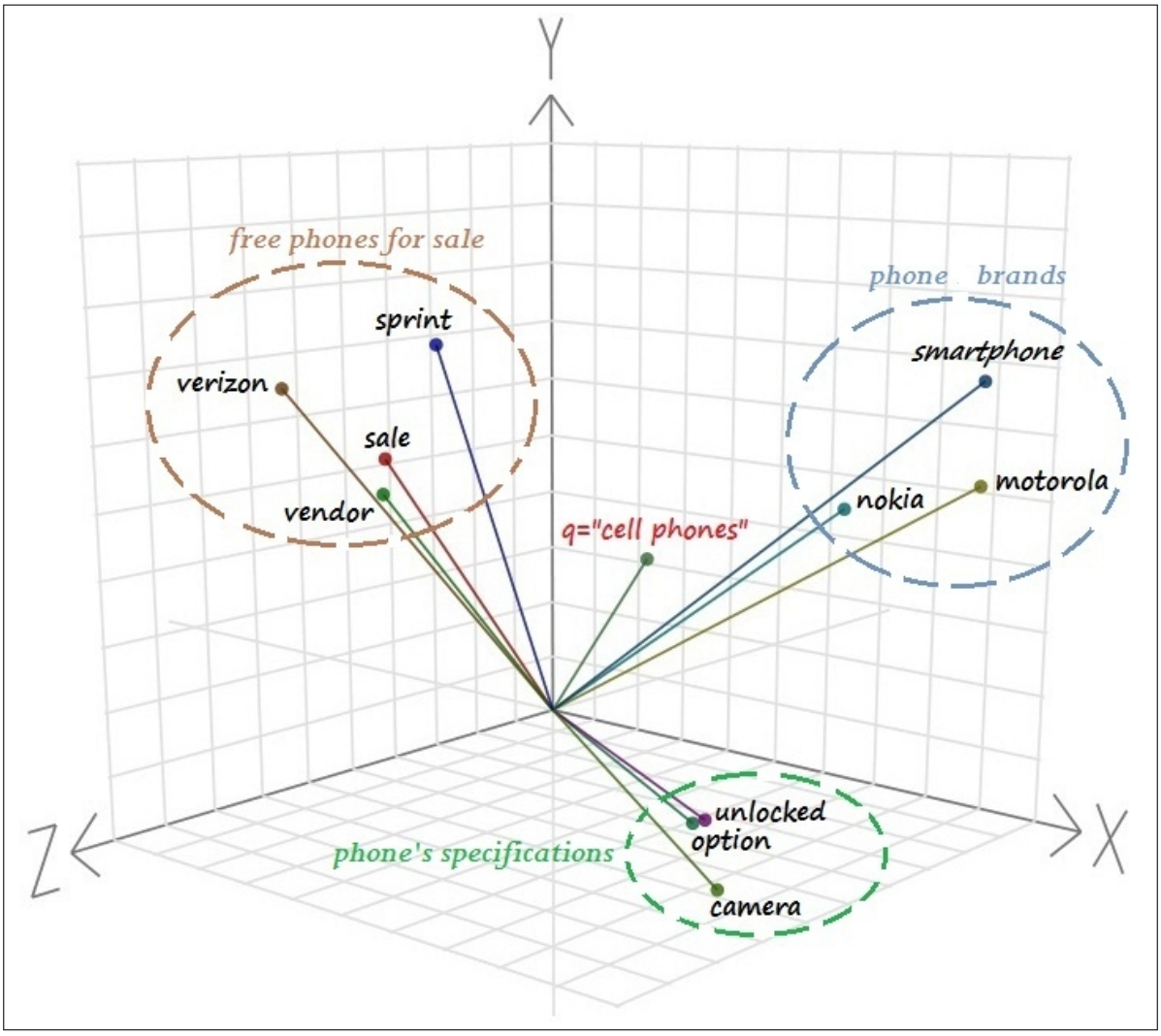

Figure 5.2: Visualization of the query's vector and the aspect embedding vectors learnt by *eRS* for the query "cell phones".

generated by *eRS*, terms like "camera", "option" and "unlocked" appear far from each other, since they correspond to the same aspect, which is *phone's specifications*.

As expected, the effectiveness with the expanded queries reflect well our above analysis. Table 5.V shows the effectiveness of different expansion methods for the same query "cell phones". As our method tries to explicitly cover different query aspects, it is interesting to observe the S-recall@20 measure, which reports the percentage of the query subtopics covered by the results. From Table

5.V, we can see clear improvements with *eRS* and *eR* over *Comb*. This result further confirms the capability of our embedding-based method to account for the coverage of the aspects.

Model	nDCG	ERR	α-nDCG	ERR-IA	NRBP	Prec-IA	S-recall
eRS	**0.227**	**0.092**	**0.574**	**0.503**	**0.516**	**0.063**	**0.750**
eR	0.185	0.072	0.503	0.431	0.428	**0.063**	**0.750**
Comb	0.169	0.064	0.456	0.417	0.433	0.052	0.493

Table 5.V: Experimental results of *eRS*, *eR* and *Comb* on "cell phones".

5.5.2 Comparison with State-of-the-Art

Let's come back to Table 5.VI that reports our results and those of existing SRD frameworks, on a set of TREC queries. First, *MMR* [22], a SRD approach without query expansion and which is based on non-redundancy (*i.e.*, novelty) when selecting documents, produces comparable results to a standard baseline. This comparison shows the limited effect of result diversification if the initial search results are not diversified. This result is in line with existing studies, such as those of Santos *et al.* [103] who show that ["... existing diversification approaches based solely on novelty cannot consistently improve over a standard, non-diversified baseline ranking"]. Second, we found that *PM-2* (which is a typical example of a term-level SRD approach) outperforms *xQuAD* (which explicitly diversifies results based on the set of manually defined query subtopics) on most of the diversity measures. This suggests that there is no need to find the whole description of query subtopics (which is a difficult task in itself) for the purpose of SRD. Yet, selecting a set of 'good' terms that cover the query aspects could be enough to produce good quality search results in term of diversity. Third, *Comb* performs equally well as the state-of-the-art *xQuAD*, and the difference between these two approaches is not significant in terms of both relevance and diversity. Although *Comb* outperforms *PM-2* in relevance scores, the latter produces better diversity scores than the former for most of the diversity measures. This comparison suggests that diversifying search results by using different resources in a simple way (*Comb*) is not usually enough to outperform other diversification approaches that use a single resource (query logs in *PM-2*). The key to success is an appropriate use of the resources, as in *eR* and *eRS*.

A further observation is that, in general, *eR* tends to perform better than other approaches, except-

Model	nDCG	ERR	α-nDCG	ERR-IA	NRBP	Prec-IA	S-recall
BL	0.267	0.133	0.385	0.279	0.241	0.179	0.578
MMR	0.263	0.131	0.387	0.278	0.240	0.179	0.579
xQuAD	0.305*-	0.152*-	0.437*-	0.314*-	0.278*-	0.207*-	0.617*-
PM-2	0.304*-	0.152*-	0.461*-+♮♢	0.340*-+♮♢	0.308*-♮♢	0.206*-	0.625*-
MSS_{modif}	0.378*-+§♮♢	0.191*-+§♮♢	0.506*-+§♮♢	0.382*-+§♮♢	0.320*-+♮♢	0.260*-+§♮♢	0.697*-+§♮♢
Comb	0.317*-♢	0.159*-	0.431*-	0.313*-	0.285*-	0.208*-	0.613*-
eR	0.372*-+§♮♢	0.185*-+§♮♢	0.521*-+§△♮♢	0.393*-+§△♮♢	0.335*-+§△♮♢	0.257*-+§♮♢	0.726*-+§△♮♢
QE_{LDA}	0.288	0.140	0.415*-	0.319*-	0.277*-	0.182	0.596
eRS	**0.392***-+§♭♮♢	**0.213***-+§△♭♮♢	**0.539***-+§△♭♮♢	**0.414***-+§△♭♮♢	**0.355***-+§△♭♮♢	**0.269***-+§♭♮♢	**0.786***-+§△♭♮♢

Table 5.VI: Comparison of our systems with existing SRD systems on 144 queries [42] from WT09, WT10 and WT11. *, -, +, §, △, ♮, ♭ and ♢ indicate significant improvement ($p < 0.05$ in Tukey's test) over *BL*, *MMR*, *Comb*, *PM-2*, MSS_{modif}, *xQuAD*, *eR* and QE_{LDA}, respectively.

ing *eRS*. This highlights the important role of embedding and provides evidence that DQE is better, in practice, than traditional approaches to diversify search results. This may answer our first research question, and we can claim that our embedding framework has shown to be effective at improving search results, and can consistently outperform existing state-of-the-art approaches. Also, by observing that *eRS* significantly outperforms *eR* in all the measures, one can clearly see the role that sparsity constraint plays in our framework. We leave the discussion about the impact of sparsity constraint to the Section 5.5.3, in which we will answer our second research question.

Interestingly, we find out that MSS_{modif} which uses random walks on the same resources adopted in our model, is actually the most competitive approach to our framework *eR*. More precisely, MSS_{modif} provides better adhoc results than *eR* but not significantly. The latter is competitive to the former in term of diversity measures. Now, by incorporating the sparsity constraint, we find that *eRS* outperforms MSS_{modif} significantly on all the measures. One possible reason is that, in He *et al.* [61], all resources are assumed to be of high quality for the query, and then no explicit distinction between resources is made. However, in our model, we weigh resources according to the query, because we believe that, different resources are effective on different queries (as we already showed in chapter 4). Moreover, in He *et al.* [61], all extracted terms using random walks have the same importance, while in our embedding framework, we *quantify* the importance of each expansion term when learning the query aspects vectors. In that way, the importance of a term with respect to a query aspect is taken into account.

Finally, QE_{LDA} which uses topical models to expand a query instead of learning query aspects vectors, performs very poorly. One possible reason is that, the topics obtained by LDA are general because they correspond to a distribution over the whole vocabulary of the documents returned for the query. Such topics correspond to the general theme of the query and do not necessarily match with the query aspects. In that case, retrieval results corresponding to expanded queries using QE_{LDA} will involve several non relevant documents that could not be useful for the purpose of SRD. For illustration, let's consider again the query "cell phones". Most of the expansion terms that our model suggests are representative of the manual subtopics (see Table 5.IV). QE_{LDA}, however, selects terms that correspond to general meaning of the query, such as "generation", "device", "communication", which may be harmful to the whole performance of the search results returned for that query.

5.5.3 Impact of the Sparsity Constraint

In this section, we examine our second research question in order to understand the role that sparsity constraint plays in our framework. As stated before, such role could be understood when directly comparing *eRS* with *eR*. In that case, we find that the sparsity constraint in our embedding framework improves both relevance and diversity, and the improvements are statistically significant (for most of the measures). We explain this result by a better modeling of the query aspects with *eRS*. Indeed, without the sparsity constraint, *eR* could produce a set of aspects that are not distinctive enough among them. A term is then mapped into a large number of the resulting aspects, making it more difficult to clearly separate terms corresponding to different aspects. When we consider the sparsity constraint, the learnt vectors by *eRS* are less dense than those learnt by *eR*, since fewer dimensions of the aspect vectors have non-zero values. This make it easier to distinguish the vectors generated by *eRS*, since they correspond to more clear aspects of the query. Indeed, an expansion term usually has a small number of meanings thus corresponding to one or a few narrow of the original query. So, imposing the sparsity constraint may lead to aspects that are more consistent with the terms semantic meaning. Besides, a user who issues a Web query is generally looking for some specific aspect of that query. By enforcing the sparsity constraint and making the aspect vectors more distinguishable among them, one can expect to find documents that are specific for a particular aspect of the query, rather than documents that cover simultaneously different aspects of the query.

For a better understanding of the impact of sparsity constraint, we compare the set of aspect

vectors learnt by *eR* and *eRS*, respectively, for the same example query "cell phones". Figure 5.3 below shows the aspect vectors (we only show some of these vectors for illustration).

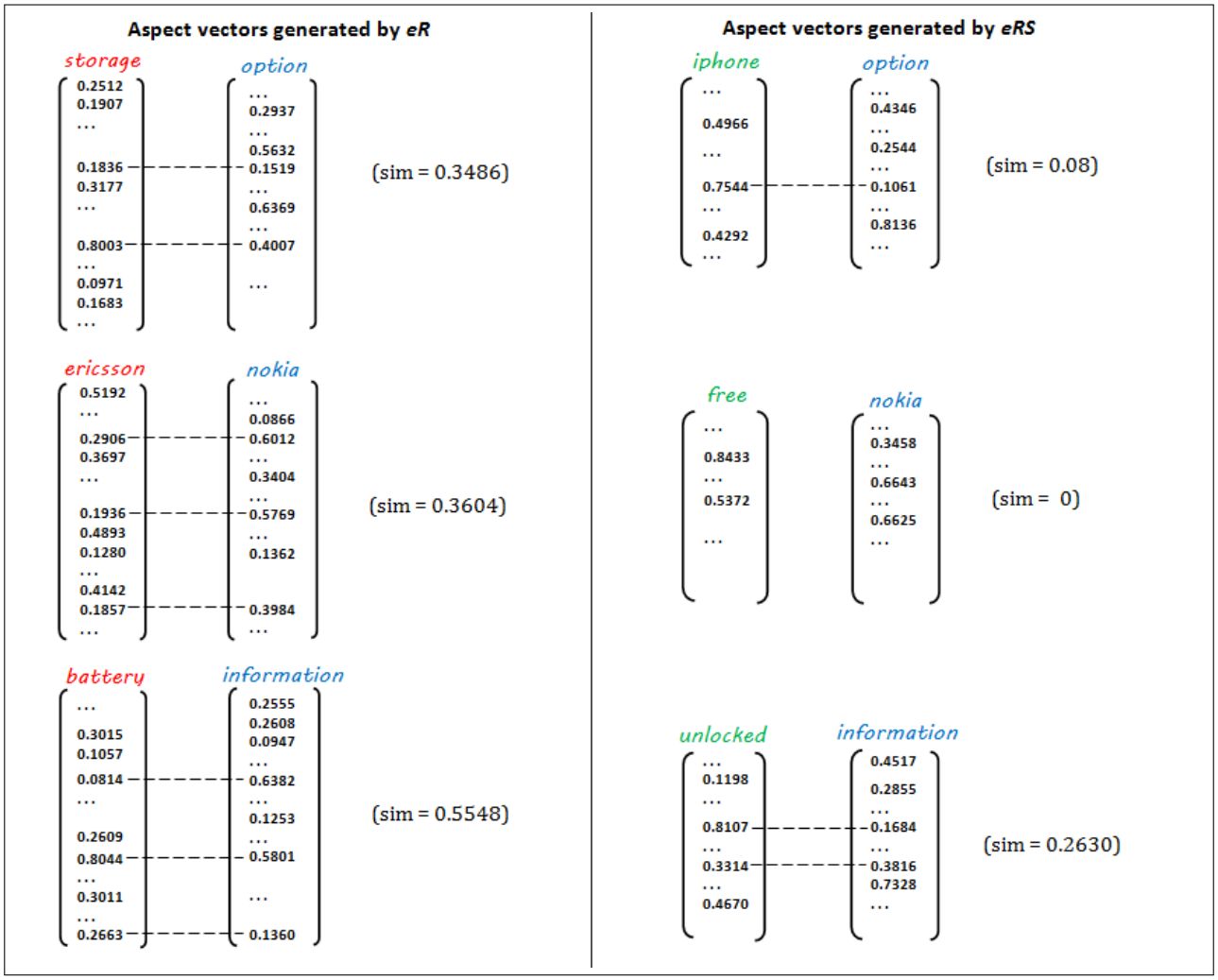

Figure 5.3: Some of the aspect embedding vectors learnt by *eR* and *eRS*, respectively, for the original query "cell phones" (we only show the non-zero values of the vectors' dimensions).

From Figure 5.3, we clearly observe that the aspect vectors learnt by *eR* (at the left) are more dense than those learnt by *eRS* (at the right). This is a clear indication that the sparsity constraint promotes the selection of discriminative expansion terms which are specific to the query aspects, thus making easier the distinction between these aspects. This is because the aspect vectors learnt by *eR* are more similar among them, than those learnt by *eRS*. For example, the average similarity

between the vectors learnt by *eR* for the query "cell phones" is 0.5219, while that learnt by eRS is 0.3977. Besides, we observe that for *eR*, expansion terms "storage", "ericsson" and "battery" are the most similar to terms "option", "nokia" and "information", respectively. Interestingly, we find that the three former terms were replaced by three other expansion terms, namely, "iphone", "free" and "unlocked", for *eRS*. The terms "option", "nokia" and "information" have a lower similarity with the new added expansion terms in *eRS* (*i.e.*, "iphone", "free" and "unlocked") than they have with the other expansion terms of *eR* (*i.e.*, "storage", "ericsson" and "battery"). Finally, we observe that the expansion term "unlocked" is selected by *eRS* but not by *eR*. Such expansion term brings an amount of new information and corresponds to one of the manual subtopics of query "cell phones" (which is subtopic 8). This helps make the learnt aspects more aligned with the manual query subtopics. This example shows that the less distinctive aspects have a lower chance to correspond to the manually defined TREC subtopics. This last point, however, will require a more in-depth investigation in the future to confirm.

5.5.4 Impact of SRD on DQE

The objective of this section is to answer our third research question on whether there is a need to diversify the search results once we diversify the query. Existing SRD approaches usually operate in two stages: In the first stage, an initial set of retrieval results of the original query is obtained. In the second stage, these obtained search results are re-ranked according to a given algorithm, in order to optimize some objective function (*e.g.*, minimize redundancy or maximize coverage or both). A legitimate question is: does merging both two diversification methods yield a larger improvement than only diversifying the query? For that, we further run the following experiments: Given an original query, we first expand it using *eRS* and run Indri on the expanded query to retrieve an initial set of documents. Let D_1 denote this set of retrieved search results. Then, we apply a second stage of document re-ranking using an existing diversification method. To alleviate the impact of the diversification method on the final results, we test with three SRD methods which are state-of-the-art: *MMR* [22], *xQuAD* [105] and *PM-2* [42, 43]. For both *xQuAD* and *PM-2*, we use the set of aspects (*i.e.*, expansion terms) that we learnt to diversify the results. Let D_2 denote the final set of retrieved results after being re-ranked using one of the SRD methods mentioned above. Now, to understand the impact of SRD on DQE, we simply compare the relevance and diversity performance of D_1 and D_2. Table

5.VII shows our statistics on the set of 144 queries from WT09, WT10 and WT11 query sets.

	nDCG	**ERR**	**α-nDCG**	**ERR-IA**	**NRBP**	**Prec-IA**	**S-recall**
D_1	0.392	0.213	0.539	0.414	0.355	0.269	0.786
D_2 (using *MMR*)	0.392	0.213	0.538	0.399	0.357	0.257	0.785
D_2 (using *xQuAD*)	0.398	0.214	0.541	0.414	0.352	0.273	0.790
D_2 (using *PM-2*)	0.400	0.217	0.544	0.409	0.355	0.271	0.788

Table 5.VII: Impact of SRD on DQE using different diversification methods.

From Table 5.VII, we observe that diversifying the search results of a diversified query does not really improve the overall performance. This result is consistent when using any of the diversification methods *MMR*, *xQuAD* or *PM-2*. In particular, for *MMR*, we observe no improvement, and in contrast the performance has been decreased for some metrics, such as ERR-IA and Prec-IA. For *xQuAD* and *PM-2*, most of the results are slightly improved. However, this improvement is not statistically significant for all the metrics. One possible reason is that, most of the documents returned in the first stage [8] are relevant to the original query and cover most of its subtopics. This means that DQE could be enough to select relevant and diversified search results, and there is no need to do a second step of document re-ranking as most of existing SRD approaches do.

By manually investigating the queries that we consider in this experiment, we find that only for the case of ambiguous queries that the second stage of diversification improves the overall results, and by low margins in general. For the other non-ambiguous queries, the second stage of diversification does not bring any improvement, instead, it may hurt the results for some queries. This observation could be explained as follows: for the case of ambiguous queries, the user intents are generally complex, and there is usually room for improvement. However, when the query is not ambiguous, the user information need is generally better defined, and the initial retrieval results of the expanded query are of better quality (they are already diversified). Therefore, it is difficult to further improve the diversity of these results. This may explain why a second diversification stage may not be useful for this kind of queries. This latter point, however, requires a more in depth investigation in the future.

Finally, it is worth noting that for *xQuAD* (respectively *PM-2*), we use the set of aspects (respectively expansion terms) selected by *eRS*, in order to diversify the results at the document level.

8. These documents are the search results of the original query after being expanded using the set of aspects (*i.e.* expansion terms) learnt by our embedding framework.

However, the obtained set of diversified results is compared to the judgments which are build upon the manual TREC subtopics. This may introduce a bias due to the problem of misalignment between the TREC subtopics and the aspects automatically identified by our method. Ideally, one should expand the query using the *reference* terms which are used to represent the TREC subtopics, then these reference terms should be diversified. By doing so, one can better understand the effect of SRD on DQE and draw more general conclusions. We leave this task for a future research.

5.5.5 Latent Aspect Embedding vs. Compact Aspect Embedding

In this section, we compare *eRS* with *CompAE* [84] when using only query logs, since both two methods are similar and attempt to learn the query aspects using embedding in order to solve the same problem. Table 5.VIII shows our results on the set of 144 queries [42] from WT09, WT10 and WT11 query sets.

Method	**nDCG**	**ERR**	**α-nDCG**	**ERR-IA**	**NRBP**	**Prec-IA**	**S-recall**
CompAE	0.359	0.180	0.505	0.379	**0.333**	**0.251**	0.724
eRS	**0.371***	**0.196***	**0.509**	**0.392***	0.330	0.249	**0.751***

Table 5.VIII: Comparison between latent aspect embedding (*eRS*) and compact aspect embedding (*CompAE*). * indicate significant improvement (p<0.05 in T-test) over *CompAE*.

From Table 5.VIII, we observe that *eRS* provides better results than *CompAE* in most of the metrics. In particular, *eRS* outperforms *CompAE* statistically in terms of adhoc relevance. Maybe, this could be explained as follows: When comparing the objective function of *eRS* (Formula 5.9) and the objective function of *CompAE* (Formula 2.20), we observe that the former considers the relevance of aspect vectors compared to the query, since, in *eRS*, aspect vectors are learnt to be good representative of the original query by enforcing that the weighted linear combination of aspect vectors should be very similar to the vector of the original query. However, *CompAE* totally ignores this constraint and considers only the diversity of query aspects. In terms of diversity measures, we observe that *eRS* generally provides better results (except in NRBP and Prec-IA), and the improvement is significant for some metrics (ERR-IA and S-recall). In fact, in addition to the learning method which is different comparing the two approaches, we observe that *eRS* is more flexible. More precisely, our method does not enforce that the norm of each aspect vector sums up to 1 (*i.e.*, $||\overrightarrow{e}||_2^2 = 1$), instead, the learnt

aspect vectors simply satisfy the constraint $||\overrightarrow{e}||_2^2 \leq 1$. This is important since, when $||\overrightarrow{e}||_2^2 \leq 1$, we promote the sparsity by enforcing that each learnt vector (*i.e.*, expansion term) corresponds to a few aspects of the query. This also helps to selecting discriminative expansion terms which are specific to some aspects of the query, rather than selecting *general* terms that may correspond simultaneously to several aspects of the query. In fact, when the user is searching for a query, she is generally seeking for a specific information need (*i.e.*, a particular intent). Consequently, we argue that it is better to select candidate expansion terms that are specific to each aspect of the query, rather than (general) expansion terms that could represent any of the query aspects. This could possibly explain why the aspects learnt by *eRS* are of better quality compared to those learnt by *CompAE*. Finally, by stating that our framework is more general since it allows the integration of multiple resources (with respect to their weights) and supports several constraints such as the sparsity constraint, we conclude that *eRS* is more effective that *CompAE*.

5.6 Approach Analysis

In this section, we answer our fourth and last research question on whether our framework is robust enough and whether it is sensitive to the choice of some parameters.

5.6.1 Robustness Analysis

In this section, we analyze the robustness of our embedding framework compared to the other existing diversification approaches. Following previous studies [42, 43], we define *robustness* as the Win/Loss ratio which is the number of queries that each diversification approach improves (Win) or degrades (Loss) compared to the baseline (*BL*), in term of α-nDCG [33] measured at cut-off 20. The comparisons are shown in Table 5.IX.

From these statistics, it is clear that *eRS* is more robust than the other baselines (it provides a Win/Loss ratio of 5.05). This suggests that the gain that we observe with *eRS* is not only due to a high improvement over a small subset of queries, but also due to a general improvement over almost the whole set of queries. This suggests that our method to SRD can be robustly applied to different types of queries.

Model	WT09	WT10	WT11	Total
MMR	16/18	19/15	20/17	55/50
xQuAD	23/16	28/14	29/11	80/41
PM-2	25/14	32/10	36/9	93/33
MSS_{modif}	33/9	35/7	34/10	102/26
Comb	24/12	29/12	32/9	83/35
eR	33/8	32/6	29/8	94/22
QE_{LDA}	20/17	25/15	18/16	63/48
eRS	39/7	35/8	32/6	**106/21**

Table 5.IX: Statistics of the Win/Loss ratio of diversification approaches.

5.6.2 Parameter Sensitivity Analysis

The results that we report before are calculated on K=20 expansion terms for *eRS*. It is interesting to assess the sensitivity of our system to K. To do this, we vary the number of expansion terms K= 5, 10, 15, 20, 30 and 40, and compare the performance of our system (*eRS*). In Figure 5.4, we plot the results on WT09 queries (we observe similar trends on WT10 and WT11).

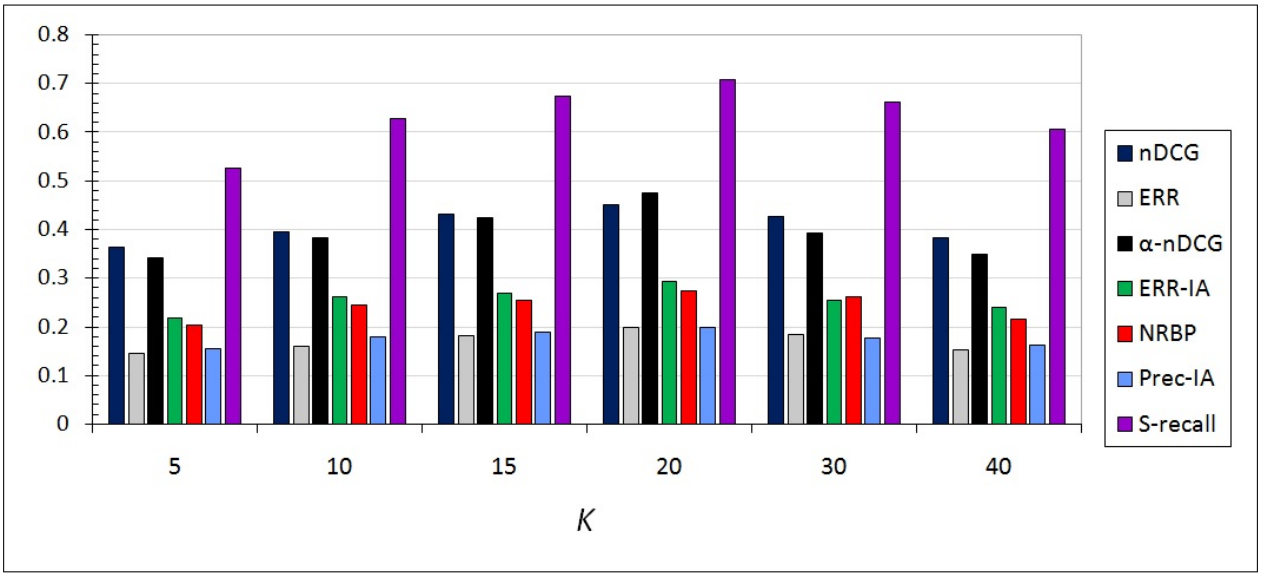

Figure 5.4: Performance of *eRS* when varying the number of expansion terms (K) on WT09 queries.

First, we observe that K=20 corresponds to the optimal parameter value yielding to the best relevance and diversity scores of *eRS*. Second, from K=5 to K=10 to K=15, both relevance and diversity

scores drastically increase. A possible explanation is the more we add expansion terms, the more likely we clarify the query meaning (increase relevance scores) and also the more likely we cover different aspects of the query (increase diversity scores). Besides, even with a few expansion terms, our approach can ensure good results in both relevance and diversity. This is because the expansion terms selected by *eRS* are relevant to the original query and can cover different aspects of the query, from the earlier iterations of the MMRE procedure. However, starting from K=30, we observe a decrease of the relevance and diversity scores when compared to those obtained by *eRS* with K=20. This is because when a large number of expansion terms are introduced, we have a higher chance of incorporating redundant and noisy terms, resulting in less relevant documents.

When varying K, we observe that different queries require a different number of expansion terms. For instance, the best performance of the ambiguous query "defender" (query #20 from WT09) is reached when K=30, while 5 expansion terms are enough for the query "mothers day songs" (query #132 from WT11) to obtain good results. In the future, it would be interesting to determine K according to the query.

Another important parameter in our model is N, the number of dimensions of aspect embeddings. Based on our previous results, we find that MSS_{modif} is the most competitive diversification framework to our approach. So, there is no need to compare *eRS* to all other approaches and we simply do comparison with MSS_{modif}. For that, we vary N in {5, 10, 20, 30, 40, 50} while keeping the other parameters of our model fixed. Figure 5.5 shows the variance of $\triangle$S-recall@20 between *eRS* and MSS_{modif} for each TREC query sets. Here, $\triangle$S-recall refers to the average difference between S-recall scores of *eRS* and MSS_{modif}, computed on different queries of WT09, WT10 and WT11.

First, we observe that our framework usually yields better results in term of S-recall@20 compared to MSS_{modif}. Interestingly, we find that *eRS* is more likely to produce better results than MSS_{modif} in term of subtopics coverage no matter the value of N we consider. Second, when we increase N from 5 to 10 to 20, the difference between the two approaches becomes larger. The main reason of this observation is that, when the number of aspects that we learn increases, the probability of covering TREC subtopics also increases. In other words, when N increases, *eRS* suggests more candidate aspects, which are more likely to match the TREC subtopics. Third, we notice that N=30 corresponds to the best setting yielding the largest improvement of subtopics coverage compared to the baseline. Finally, for higher values of N (N=40 and N=50), the performance of *eRS* slightly decreases. One

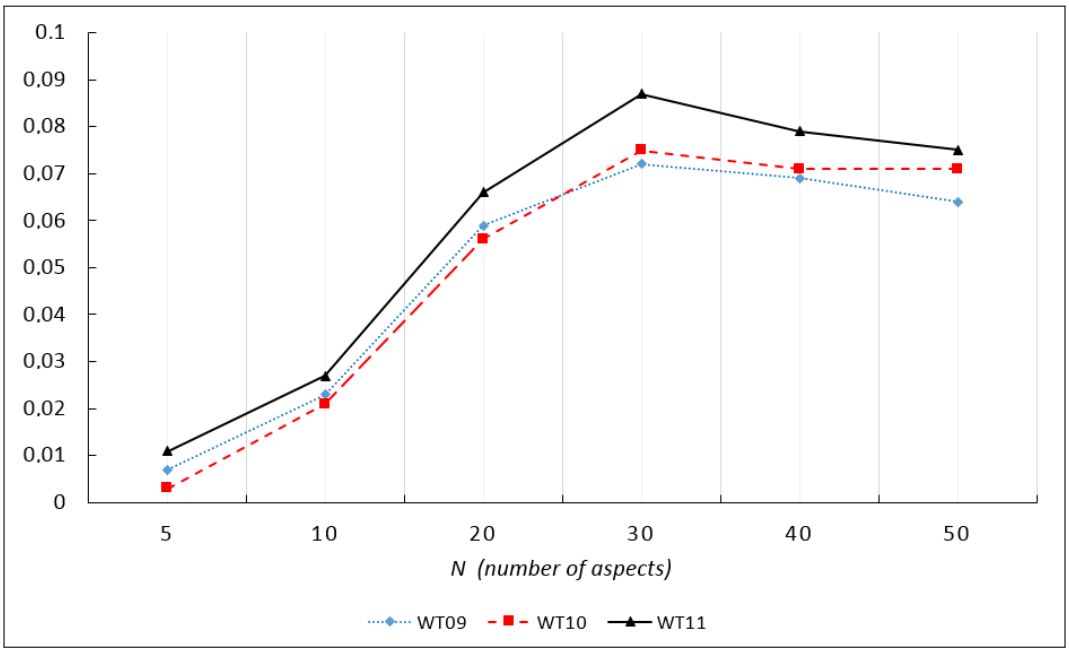

Figure 5.5: Performance difference between *eRS* and MSS_{modif} in term of △S-recall@20 when varying the number of learnt aspects (N).

possible explanation is that, when N becomes large, our model learns more aspects, which provides more chance to cover the subtopics of TREC; but different aspects also have a higher risk to be actually related to the same subtopic. In other words, the aspects become more dependent. In the future, we will tackle this problem by modeling the dependency between the aspects that we learn.

5.6.3 Sensitivity of our Approach to Perturbations

In the previous section, we have shown that our approach can produce good results even within a few number of expansion terms. In this section, we investigate the reason of this robust behaviour. To do so, we propose to apply a simple perturbation to the set of expansion terms selected from some resource, and observe the behaviour of *eRS* compared to a standard DQE approach. A more robust method should be more resistant to perturbation. In particular, we propose to *substitute* one or more terms in the expanded query with one or other terms that are randomly chosen. To preserve the randomness criterion, we select random candidate expansion terms from the whole document collection (ClueWeb09-B in our experiments). In that way, candidate expansion terms used for substitution have

very low chance of being related to the query aspects.

Let's consider $MMRE_r$ as a baseline method for DQE which selects candidate expansion terms from resource r (similar to Formula 5.14). For a fair comparison, both two compared methods should use the same resource. For that, we compare $MMRE_r$ with *eRS* when using only resource r. In the remainder of this section, we denote by eRS_r this latter method (similar to Formula 5.1), where resource r could be ConceptNet ($r = C$), Wikipedia ($r = W$), query logs ($r = L$), or feedback documents ($r = F$).

Let q be an original query and q_r be the expanded query whose expansion terms are from resource r. In our experiments, we select for each query q, 10 candidate expansion terms according to their similarity (relevance) to q. We define a procedure that substitutes n terms from q_r with other terms, randomly chosen. Notice that parameter n controls the perturbation level, where $n \in \{0, 1, 2, ..., 10\}$. In particular, when $n = 0$, we have no perturbation, and in that case, q_r remains unchanged. On the other extreme, when $n = 10$, all the expansion terms of q_r have been substituted, thus resulting in a totally new random query expansion. It is worth noting that no one of the original query terms should be substituted by another, otherwise we run a risk of changing the users' original intents (since we have different queries).

Note that not all expansion terms in q_r are equally important and different terms have different weights (which correspond to $sim_r(e,q)$). Hence, the substitution of an important term in q_r by another one may affect more the performance of our approach than when applying a substitution of a less important term. In particular, if a term e in q_r is the unique term that represents some aspect of q, then substituting e by another term leads to the non-coverage of that aspect. On the other hand, if e is not the unique term covering some aspect in q, then, even after the substitution of this term by another, such aspect is still covered due to the other expansion terms that appear in q_r and correspond to that aspect. Let's recall the example query "cell phones" (see Table 5.IV). Terms like "prepaid" and "unlocked" are important since they are the unique terms that correspond to aspect *prepaid phones* and aspect *unlocked phones*, respectively, in the list of expansion terms obtained by *eRS*. Therefore, by substituting one of these two terms by another one, we run the risk of not covering one of the two previous aspects. However, terms like "sale" and "vendor" tend to correspond to the same aspect *phones for sale*. Hence, substituting one of the terms by another will not affect the coverage of that aspect.

To tackle this problem, we propose to iteratively run our substitution process 10 times for each query q_r and for each possible value of $n \in \{0, 1, 2, ..., 10\}$, and consider the average. More precisely, we apply eRS_r and $MMRE_r$ separately on the set of expansion terms of q_r after substituting n of its terms. At the end, each method selects a number of expansion terms among 10. In our experiments, we keep 5 expansion terms in the resulting query, on which we compute relevance and diversity scores. Figure 5.6 and Figure 5.7 show the performance of eRS_r and $MMRE_r$ when varying n, in terms of α-nDCG@20 (diversity) and nDCG@20 (adhoc retrieval), respectively, on WT09 queries. We only show the results of the queries of WT09. On results of the queries of WT10 and WT11, we make comparable observations.

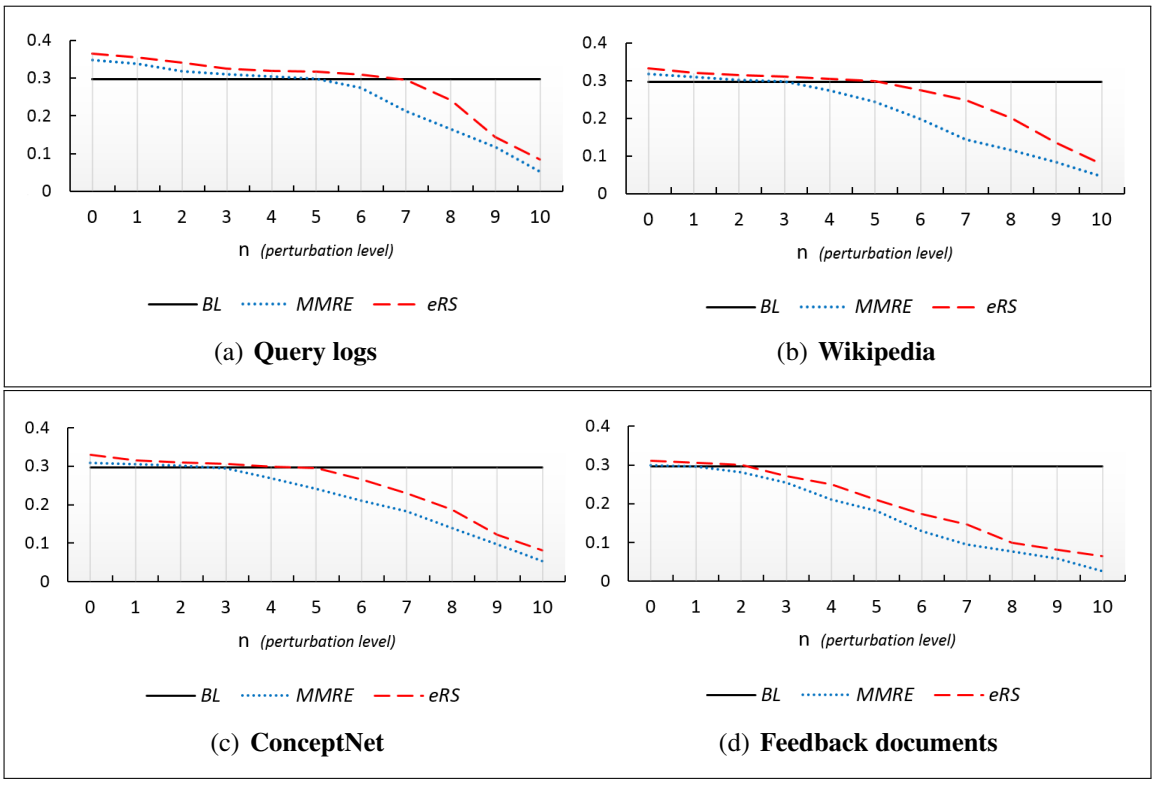

Figure 5.6: Performance of eRS_r and $MMRE_r$ for different resources, in terms of α-nDCG@20 across different levels of perturbations, on WT09 queries.

First, as expected, we found that the more we substitute terms from q_r with other randomly chosen ones, the more the whole performance of our approach decreases. From Figure 5.6 and 5.7, we also observe that our embedding framework eRS_r is more robust to perturbations than $MMRE_r$, since its performance decreases more slowly than $MMRE_r$, and these observations are consistent over all the

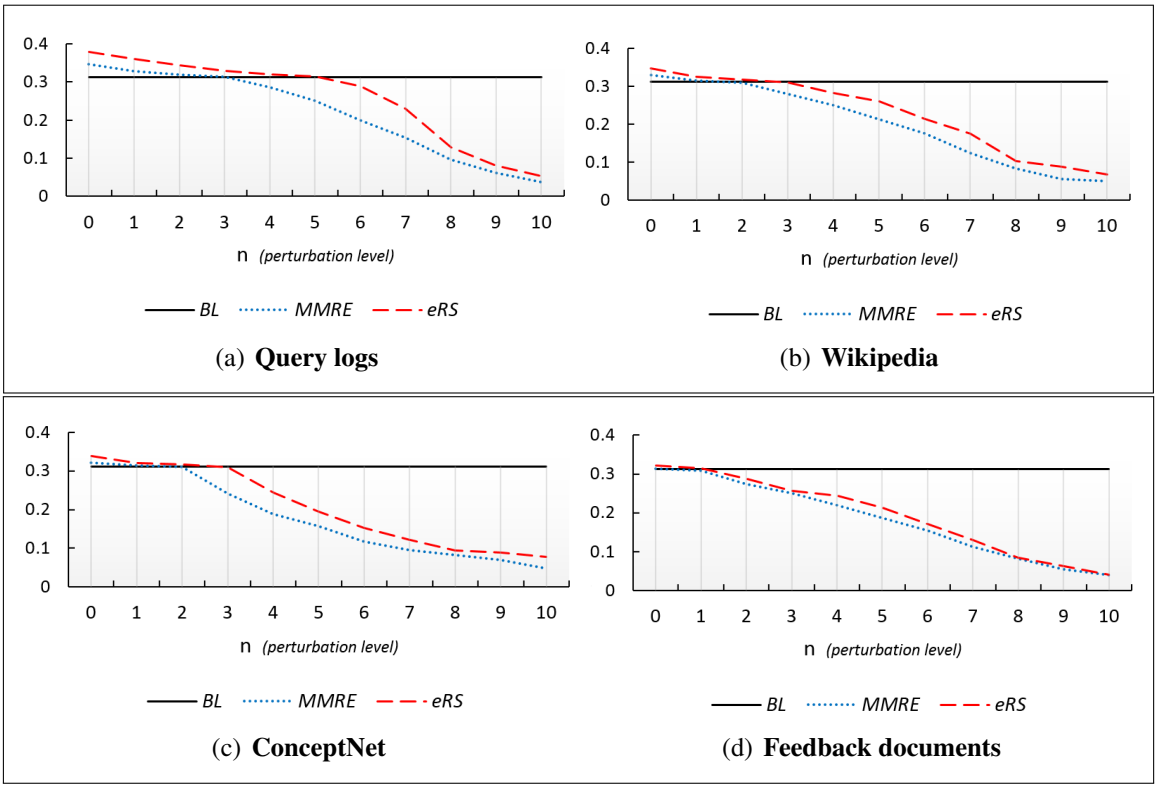

Figure 5.7: Performance of eRS_r and $MMRE_r$ for different resources, in terms of nDCG@20 across different levels of perturbations, on WT09 queries.

resources that we consider here. To understand the reasons, let us examine the formula of $MMRE_r$ and that of eRS_r. Let's consider first the formula of $MMRE$ (similar to Formula 5.14) and assume e to be a noise expansion term. On the one hand, $sim_r(e,q)$ is low since term e is not relevant to q. But on the other hand, term e is very different to the other terms which are related to the query. Given a query, a noise expansion term has higher chance (than the other relevant expansion terms) of being less redundant to any other relevant term, for the same query. Hence, such noise term e usually has chance of being selected by $MMRE_r$. When optimizing the objective function of eRS_r (similar to Formula 5.1) during the gradient descent process, the dimensions of each aspect vector are updated. In particular, the dimensions of aspect vectors corresponding to noise expansion terms will be decreased, and relevant aspect vectors will be promoted. Since we additionally enforce the sparsity constraint, the dimensions' values of such noise aspect vectors will converge to very low values or even zeros. Hence, these obtained vectors will have low similarity with the vector of q and with the other relevant aspect vectors. Consequently, such noise vectors will be penalized during the selection

stage of eRS_r. This highlights the importance of computing the similarity between expansion terms at aspect level (case of eRS_r) rather than the surface term level (case of $MMRE_r$). In the former, the semantic of expansion terms is considered which explains why noise terms were discarded by our method.

Our last observation concerns the resources that we consider in this work. By comparing the performance of both eRS_r and $MMRE_r$ across different resources, we observe that the performance of eRS_L (resp. $MMRE_L$) decreases more slowly than eRS_r (resp. $MMRE_r$) of the other resources. Besides, the performance of eRS_r (resp. $MMRE_r$) when using Wikipedia and ConceptNet are comparable on different queries of WT09. The performance of eRS_F (resp. $MMRE_F$), however, is the lowest compared to that of eRS_r (resp. $MMRE_r$) of the other resources. These observations are in line with our previous works (see chapter 3)in which we observed the same results for the same resources. From that, we can see that different resources contribute differently to the diversity of the search results. More precisely, some resources (*e.g.* query logs) seem to be more effective than other ones (*e.g.* feedback documents) to suggest expansion terms from a good quality, thus improving the relevance and diversity of search results.

5.6.4 Complexity Analysis

Complexity issues can be tackled by noting that raw terms similarity based on each resource is computed off-line (at indexing time), thus eliminating any additional on-line costs. Note that it is possible to pre-calculate the similarity between the terms using each resource. Thus, for each query, one can see the related expansion terms. During this process, we select from each resource, and for each test query, a few candidate expansion terms. As there are a limited number of test queries (we considered 148 from WT09, WT10 and WT11 in our experiments), a limited number of resources (we used 4 resources in this study), and a limited number of candidate expansion terms for each query, and from each resource (this number is set to 10), the whole amount of computation is generally limited and its complexity is $\mathcal{O}(1)$.

The on-line process is to determine the word embedding and to select some expansion terms for each query (see Algorithm 5.1). Let t be the total number of iterations required to learn the aspect vectors for a given query. In each iteration, and for each aspect vector, we compute the prediction error for each pair of aspect vectors which requires $M \cdot (M\text{-}1)$ calculations, where M is the number of

aspect vectors (*i.e.*, expansion terms). We also compute the gradient of the loss with respect to each aspect vector, and we update each vector (within its weight). Hence, the complexity of this process is $\mathcal{O}(M^2 \cdot t)$. Since we consider just 4 resources in this work, all the complexity related to compute the gradient of the loss function with respect to each resource and to update the weight of each resource is negligible.

Finally, in the last step of our method, we apply the *MMRE* procedure (described in Formula 5.14) to select K expansion terms among M. In each iteration, we compute the similarity between an aspect vector and the query, and between a pair of vectors. For the former computation, we need to perform only M calculations in the first iteration, which is $\mathcal{O}(M)$. Note that these similarity scores could be directly used for the next iterations of *MMRE* since the aspect vectors are already learnt at this step. Similarly, the latter computation which requires $\frac{M \cdot (M-1)}{2}$ calculations is also done only during the first iteration of the *MMRE* procedure. This is because $sim(\overrightarrow{e_i}, \overrightarrow{e_j}) = sim(\overrightarrow{e_j}, \overrightarrow{e_i})$ where $\overrightarrow{e_i}$ and $\overrightarrow{e_j}$ are two aspect vectors corresponding to the same query. Hence, the *MMRE* procedure requires a complexity of $\mathcal{O}(M^2)$. Therefore, the complexity of the whole on-line process is of $\mathcal{O}(M^2 \cdot t + M + M^2)$. As there are a limited number of candidate expansion terms for a query (M=40 in our experiments), the whole amount of computation remains limited.

5.7 Discussion

The aspect vectors produced by our embedding framework depend on how these vectors are initialized. By uniformly initializing each value with $\frac{1}{\sqrt{N}}$ (N=30 in our experiments), we obtain good results in practice. However, this may not be the best setting for initial values. We leave the problem of setting of initial values to a future work.

One interesting research question is whether there is a correspondence between the aspects that we learn using *eRS* and those subtopics defined by TREC assessors, and whether this correspondence is necessary. To answer this question, we have to align the automatically created aspects with manual subtopics, and estimate a degree of matching. By investigating the relationship between the degree of matching and the retrieval effectiveness, we will be able to obtain some clues to answer this question. To do that, we proceed as follows: Firstly, we extract the *reference* words (*i.e.*, those of TREC ground truth subtopics) as well as the most representative words for each subtopic of each query, by using the

relevant documents per subtopic which are available from the relevance judgments provided by TREC assessors. Then, one can compare the words that we extracted with the expansion terms suggested by our approach. Such alignment between these two sets could help not only to assess how much our learnt aspects correspond to ground truth subtopics, but also to *quantify* which resource is the most helpful or provides the best coverage. Our investigation of the alignment results for the set of TREC queries shows that there is a *partial* alignment between our learnt aspects and the manually defined subtopics of TREC, which is expected. Indeed, we found that several of our aspects match with several TREC subtopics, but at the same time, other TREC subtopics do not match any of our aspects. In addition, our framework suggests *new* aspects that seem to be relevant but were not identified by TREC assessors. For example, Wikipedia suggests 'Anti-Violence Program' as a candidate aspect for the query "avp" (query #52 from WT10) which seems to be a reasonable aspect for this query. However, such aspect was not identified by TREC assessors. By considering this kind of aspects that do not appear in the ground truth subtopics, one could hurt the performance of the final results obtained by our framework. We believe that *filtering* such aspects that do not match with the TREC subtopics may help improve the performance of our system in term of diversity scores. However, this requires to define a method that automatically determines whether an aspect is relevant or not for a given query. We don't address this issue in this work, and also leave that for our future work.

Finally, note that all the results that we report in this dissertation are compared based on statistical tests. However, it is worth noting that the *statistical significance* (resp. *insignificance*) does not necessarily imply the *practical significance* (resp. *insignificance*) [102]. In fact, the statistical significance refers to the probability that the means differences between systems have occurred due to sampling errors, while the practical significance looks whether the difference between systems is large enough to be noticeable to a user who uses these systems in practice. For instance, a search engine A can consistently and statistically outperform another search engine B in one or multiple metrics. However, in practice, this difference may not be significant since the user submitting the same search query to both search engines A and B will not observe any real difference between the results returned by A and those returned by B. This observation is consistent especially when we consider a sample of topics with a big size, in which one system can outperform another system for (almost) all topics but by small differences. Therefore, it is difficult to conclude about the practical significance based on the statistical significance. In general, to assess the practical significance between two systems, one

can conduct a user study for example, and directly observe the behavior of the user in front of both two systems.

5.8 Conclusion

The basic approaches to search result diversification focused on extracting diversified documents from the initial retrieval results. In our previous studies, we observed that it is important to expand the query to have a better coverage of different aspects. A typical DQE approach uses one or several resources to generate a set of diverse expansion terms to obtain a better coverage of the different aspects of a query. Its focus is mainly on removing redundant expansion terms. However, the diversity (or similarity) of expansion terms is measured directly at term level and it is not guaranteed that the expansion terms cover the aspects. We argue that a better measure of term diversity should rely on a better representation of query aspects that could reflect query subtopics (in the ideal case). In this chapter, we propose a method that uses aspect embeddings to represent implicit query subtopics/intents at a latent semantic level. Diversified expansion terms are determined based on their mapping into the aspect space. By doing so, the selected expansion terms not only are different among them, but also can better cover the underlying aspects of the query. In addition to aspect modeling, we also use several resources to suggest expansion terms. Our experiments on TREC diversification data confirm that our aspect modeling significantly contributes in improving the effectiveness of SRD.

It is worth noting that we participated to NTCIR-IMine task [9] for both subtopic mining and document ranking sub-tasks [15]). Note that NTCIR is an international evaluation campaign which proposes a series of evaluation tasks designed to enhance research in information access and technologies that are related to information retrieval (and other domains). For instance, IMine task (to which we participated), aims to evaluate technologies and methods of satisfying different user intents behind a Web search query which may help generating diversified search rankings. NTCIR provides large-scale test collections reusable for experiments, and evaluates the different methods proposed and tested by participating research groups. During our participation to NTCIR-IMine task, we experimented our latent aspect embedding framework that we proposed in this chapter, using five representative resources: the four resources that we consider in this chapter, in addition to query suggestions

9. http://www.thuir.org/IMine

provided from Bing, Google and Yahoo!. We tested our approach using the collection of documents ClueWeb12-B13 [10] and the set of 50 English queries which were provided by the organizers of NTCIR. Experimental results show that our best run is ranked No. 2 among all 15 runs of participating groups. This highlights the effectiveness of our proposed aspect embedding approach.

10. http://lemurproject.org/clueweb12

Chapter 6

Conclusion and Future Work

In this chapter, we summarise the results and conclusions of the dissertation. We also discuss opportunities for extending our work.

6.1 Summary of Results and Contributions

The main objective of this thesis is to define a new method for SRD which diversifies the expansion terms of the query instead of the initial retrieval results. Our approach is motivated by the fact that the quality of existing document-level diversification methods is strongly dependent on that of initial retrieval results. However, it has been observed that this does not ensure a good coverage of the various search intents due to the problem of query ambiguity and dominant subtopics.

The first contribution of this thesis is a new diversified query expansion method, called *MMRE* (*Maximal Marginal Relevance-based Expansion*), which uses an external resource (namely ConceptNet) to select diversified candidate expansion terms following the Maximal Marginal Relevance principle [22]. The reason for using external resources instead of PRF is that expansion terms derived from feedback documents may still depend on the retrieval results from the original query; should some aspects be not well covered in the initial retrieval results, this method will neither cover them. Our results clearly show the usefulness of diversifying the expansion terms of the query, this outperforms existing state-of-the-art approaches that do not diversify the query.

Since the coverage of *MMRE* based on a single resource may be limited to that of the resource, and that combining several resources may yield a better coverage of the query aspects (multiple resources tend to complement each other for the purpose of SRD), we propose in the second contribution a general and unified framework for DQE by extending *MMRE* with different resources. In particular, we consider three additional resources: Wikipedia, query logs and feedback documents. Our experimental results on several TREC data sets demonstrate its effectiveness compared to existing diversification methods and suggest the usefulness of incorporating different resources for DQE.

When different resources are incorporated for DQE, they are combined in a uniform way in the

literature. However, we observe that different resources may not necessarily have the same importance to different queries. Consequently, a better approach is to promote expansion terms selected from resources with higher contribution to the diversity results of a query, and penalize the expansion terms derived from resources having a lower contribution to the diversity results of the same query. To reach this goal, we present in our third contribution a query-dependent resource weighting method which determines how useful a resource is. We use a set of features to determine the usefulness of a resource. We thoroughly evaluate our approach on TREC 2009, 2010 and 2011 Web tracks and show that our system outperforms the existing methods without resource weighting, and that query level resource weighting is superior to the non-query level resource weighting for the purpose of DQE.

In the previous methods on DQE, word similarity is measured at the term (surface) level. A potential problem is that an expansion term can appear different from the previous expansion terms, yet it describes exactly the same semantic intent. Consequently, term-level DQE methods may not ensure a good coverage of the query intents. To solve this problem, we propose in this thesis a novel method aiming to diversify the expansion terms of a query at the (semantic) aspect level. More precisely, we propose a method for DQE relying on an explicit modeling of query aspects based on embedding, which is trained in a supervised manner according to the principle that related terms should correspond to the same aspects. Based on this novel representation of the query aspects and expansion terms, we design a greedy selection strategy to choose a set of expansion terms to explicitly cover all possible aspects of the query. We call our method *latent semantic aspect embedding* since this method allows us to select expansion terms at a latent semantic level so as to cover as much as possible the aspects of a given query. In addition, this method also allows us to incorporate several different external resources to suggest potential expansion terms, as well as other constraints, such as the sparsity constraint. We test our method on several TREC diversification data sets, and show that our method significantly outperforms the state-of-the-art SRD approaches. In particular, unlike term-level DQE approaches, our latent aspect embedding method ensures that the selected expansion terms not only are different among them, but also can better cover the underlying query aspects. This clearly shows that the explicit modeling of query aspects brings significant gains which improves the overall effectiveness of SRD.

6.2 Future Work

In this thesis, we proposed to use DQE to solve the SRD problem. This opens the door to a range of new research directions for SRD. While the proposed approaches showed improved results compared to the state-of-the-art, our study has several limitations, which could be studied in future work. In the remainder of this section, we discuss these issues which we categorized into either immediate future research directions or farther future research directions.

6.2.1 Short Term Research Directions

Learning the Optimal Number of Expansion Terms / Aspects per Query

When expanding a query using a set of terms, we consider the same number of expansion terms for any query. However, during our experiments, we observed that different queries require different number of expansion terms. Similarly, in our proposed latent aspect embedding method, a fixed number of aspects is used. In practice, the number of aspects can vary from a query to another, depending on how ambiguous it is and how rich the document collection is regarding to the topic. It will be interesting to develop ways to automatically determine the appropriate number of expansion terms and the appropriate number of aspects that should be learnt for each query. For example, it is known that the user information need behind ambiguous queries is much complex compared to that of clear queries in which the user information need is generally well defined. Hence, we believe that ambiguous queries require a higher number of expansion terms and a higher number of aspects that should be learnt compared to the clear queries. If it is the case, then a better approach is to automatically learn the optimal number of expansion terms and aspects regarding to each query. This requires further investigations in the future.

Modeling the Dependency between the Learnt Aspect Vectors

The aspects that we learnt could or could not be dependent one of the other. For some queries, we observe that different aspects may be related to the same subtopic. This lead to select (redundant) documents that appear different from the previously selected ones, yet describe exactly the same semantic content. This may have a negative impact on the overall performance of our approach:

once a document about some aspect has been selected, a similar document (about the same semantic aspect) will not be useful to the user, since it does not bring any novel information. Hence, it would be interesting to investigate the possible dependency between aspects in diversified query expansion. We believe that this may improve the diversity of results and make our aspect embedding framework more effective.

Selective Diversified Query Expansion

When a DQE approach is proposed, it has usually been used on *all* the queries regardless to their nature. We believe that diversification should not be systematically applied for any query: The results for some queries need to be diversified much more than other queries. For example, we expect that ambiguous queries would require an approach different from that of non-ambiguous queries. If this is the case, then a better diversification strategy is to selectively choose the appropriate diversification level according to the query type. More precisely, the extent (*i.e.*, the interpolation parameter λ which controls the trade-off between relevance and diversity when selecting candidate expansion terms) should be determined according to the query type. A possible strategy is to classify queries into *ambiguous*, *broad* and *clear* categories [110] and to diversify to different degrees for queries in different categories.

6.2.2 Long Term Research Directions

Directly Diversifying Search Results using the Aspect Vector Representation

The ultimate purpose in result diversification is to diversify the search results so as to cover as much as possible the query intents. In this dissertation, we perform a middle step of generating diverse query expansion terms, and map each expansion term into an aspect vector. Then, we directly run the expanded query on an IR system (such as Indri) in order to obtain a diversified set of search results. Theoretically, the results with a diversified query expansion can be further re-ranked to construct a final search result list. In Chapter 5, we coupled DQE with some existing SRD approaches, but this did not show expected gains. This could be due to the way that the existing approaches are used. It would be interesting to study how to use our proposed aspect vector representation to directly generate diverse search results, for example, by mapping a document into the same vector space and choosing

a set of diversified documents by running an algorithm similar to the algorithm described in Figure 5.1.

Time-Aware Diversified Query Expansion

Existing diversification approaches consider a set of *static* query subtopics and no attention has been paid on leveraging the *temporal dynamics* of query aspects. In fact, user query intents are not necessarily stable and may frequently change over time, especially for the so-called *fresh queries* which are time-sensitive [19]. For example, the query "*US Open*" is likely to correspond to *tennis open* in September, or *the golf tournament* in June [93]. Consequently, it is important to consider the popularity of the query aspect with respect to the time. In particular, in addition to the relevance of selected expansion terms, and their non-redundancy, an additional time dimension should also be considered. This will require dynamic mining of latent aspects over time.

Personalized Search Result Diversification

The goal of a diversification approach is to return results that could satisfy the user information need. Existing approaches in SRD (also including ours) attempt to diversify the results for *all* the users. However, different users may have very different intents for the same query. Therefore, we believe that a personalized diversification could be more effective and may increase the user satisfaction, since it focuses on returning the documents that correspond to a particular user. By *personalized diversification*, we mean a diversification that is conducted by the user profile and her preferences. For example, one can think to directly inject the user profile into the objective function of a diversification method, then, the purpose is to select documents that maximize such objective function. However, putting this approach in practice requires the availability of the data about the user profiles, which could be difficult to collect in practice.

Related Publications

I have started my PhD in Fall 2011. The following is a list of our publications related to this dissertation.

Articles in Journals:

1. Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. Learning Latent Aspects for Diversifying Search Results using Multiple Resources. (*To be submitted to Journal of Inf. Retr., 2015*)

Articles in Refereed Conferences:

1. Arbi Bouchoucha, Jing He, and Jian-Yun Nie. Diversified Query Expansion using Conceptnet. In *Proc. of Conference on Information and Knowledge Management (CIKM)*, pp. 1861-1864, Burlingame, USA, 2013. (*Acceptance Rate: 17%*) [13]

2. Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. Integrating Multiple Resources for Diversified Query Expansion. In *Proc. of European Conference on Information Retrieval (ECIR)*, pp. 437-442, Amsterdam, Netherlands, 2014. (*Acceptance Rate: 23%*) [14]

3. Xiaohua Liu, Arbi Bouchoucha, Alessandro Sordoni, and Jian-Yun Nie. Compact Aspect Embedding for Diversified Query Expansions. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*, pp. 115-121, Quebec-City, Canada, 2014. (*Acceptance Rate: 28%*) [84]

4. Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. Towards Query Level Resource Weighting for Diversified Query Expansion. In *Proc. of European Conference on Information Retrieval (ECIR)*, pp. 1-12, Vienna, Austria, 2015. (*Acceptance Rate: 23%*) [16]

5. Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. Selective Diversified Query Expansion using Query Type. In *Proc. of World Wide Web conference (WWW)*. (*To be submitted by November 2015*)

Technical Papers:

1. Arbi Bouchoucha, Jian-Yun Nie, and Xiaohua Liu. Universite de Montreal at the NTCIR-11 IMine Task. In *Proc. of NII Testbeds and Community for Information access Research (NTCIR) IMine task*, pp. 28-35, Tokyo, Japan, 2014 *(Ranked No. 2 over all participating research teams)* [15].

Bibliography

[1] *WSCD '09: Proceedings of the 2009 Workshop on Web Search Click Data*, Barcelona, Spain, 2009.

[2] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of WSDM*, pages 5–14, Barcelona, Spain, 2009.

[3] Giambattista Amati, Claudio Carpineto, Giovanni Romano, and Fondazione Ugo Bordoni. Query difficulty, robustness and selective application of query expansion. In *Proceedings of ECIR*, pages 127–137, Sunderland, UK, 2004.

[4] Ricardo Baeza-Yates, Liliana Calderón-Benavides, and Cristina González-Caro. The intention behind web queries. In *Proceedings of SPIRE*, pages 98–109, Berlin, Heidelberg, 2006.

[5] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. Modern Information Retrieval. Pearson Education Ltd., Second edition, Harlow, England, 2011.

[6] Ranieri Baraglia, Fidel Cacheda, Victor Carneiro, Diego Fernandez, Vreixo Formoso, Raffaele Perego, and Fabrizio Silvestri. Search shortcuts: A new approach to the recommendation of queries. In *Proceedings of RecSys*, pages 77–84, New York, USA, 2009.

[7] Michael Bendersky, David Fisher, and W. Bruce Croft. Umass at trec 2010 web track: Term dependence, spam filtering and quality bias. In *Proceedings of TREC*, 2010.

[8] Michael Bendersky, Donald Metzler, and W. Bruce Croft. Effective query formulation with multiple information sources. In *Proceedings of WSDM*, pages 443–452, Washington, USA, 2012.

[9] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Mach. Learn. Res.*, 13:281–305, 2012.

[10] David M. Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Mach. Learn. Res.*, 3:993–1022, 2003.

[11] Abraham Bookstein. Information retrieval: A sequential learning process. *Journal of the American Society for Information Science*, 34(5):331–342, 1983.

[12] Léon Bottou. Stochastic learning. In Olivier Bousquet and Ulrike von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, pages 146–168. Springer Verlag, Berlin, Germany, 2004.

[13] Arbi Bouchoucha, Jing He, and Jian-Yun Nie. Diversified query expansion using conceptnet. In *Proceedings of CIKM*, pages 1861–1864, Burlingame, USA, 2013.

[14] Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. Integrating multiple resources for diversified query expansion. In *Proceedings of ECIR*, pages 437–442, Amsterdam, Netherlands, 2014.

[15] Arbi Bouchoucha, Jian-Yun Nie, and Xiaohua Liu. Universite de montreal at the ntcir-11 imine task. In *Proceedings of NTCIR IMine task*, pages 28–35, Tokyo, Japan, 2014.

[16] Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. Towards query level resource weighting for diversified query expansion. In *Proceedings of ECIR*, pages 1–12, Vienna, Austria, 2015.

[17] Bert Boyce. Beyond topicality: A two stage view of relevance and the retrieval process. *Information Processing and Management*, 18(3):105 – 109, 1982.

[18] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[19] Ricardo Campos, Gaël Dias, Alípio M. Jorge, and Adam Jatowt. Survey of temporal information retrieval and related applications. *ACM Comput. Surv.*, 47(2):15:1–15:41, 2014.

[20] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of SIGIR*, pages 243–250, Singapore, 2008.

[21] Gabriele Capannini, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. Efficient diversification of web search results. *Journal of VLDB Endow.*, 4(7):451–459, 2011.

[22] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, pages 335–336, Melbourne, Australia, 1998.

[23] Ben Carterette. An analysis of np-completeness in novelty and diversity ranking. In *Proceedings of ICTIR*, pages 200–211, Cambridge, UK, 2009.

[24] Ben Carterette and Praveen Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of CIKM*, pages 1287–1296, Hong Kong, China, 2009.

[25] Benjamin A. Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Trans. Inf. Syst.*, 30:4:1–4:34, 2012.

[26] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of CIKM*, pages 621–630, Hong Kong, China, 2009.

[27] Olivier Chapelle, Shihao Ji, Ciya Liao, Emre Velipasaoglu, Larry Lai, and Su-Lin Wu. Intent-based diversification of web search results: metrics and algorithms. *Inf. Retr.*, 14(6):572–592, 2011.

[28] Harr Chen and David R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. In *Proceedings of SIGIR*, pages 429–436, New York, USA, 2006.

[29] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, 1990.

[30] Charles L. Clarke, Maheedhar Kolla, and Olga Vechtomova. An effectiveness measure for ambiguous and underspecified queries. In *Proceedings of ICTIR*, pages 188–199, Cambridge, UK, 2009.

[31] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Gordon V. Cormack. Overview of the trec 2010 web track. In *Proceedings of TREC*, 2010.

[32] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Ellen M. Voorhees. Overview of the trec 2011 web track. In *Proceedings of TREC*, 2011.

[33] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of SIGIR*, pages 659–666, Singapore, 2008.

[34] Charles L.A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the trec 2009 web track. Technical report, 2009.

[35] Gordon V. Cormack, Mark D. Smucker, and Charles L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Journal of Inf. Retr.*, 14(5):441–465, 2011.

[36] Veronica Gil Costa, Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Sparse spatial selection for novelty-based search result diversification. In *Proceedings of SPIRE*, pages 344–355, Pisa, Tuscany, Italy, 2011.

[37] Maurice Coyle and Barry Smyth. On the importance of being diverse. In *Proceedings of IIP*, volume 163, pages 341–350. Beijing, China, 2005.

[38] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of SIGIR*, pages 299–306, Tampere, Finland, 2002.

[39] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *Proceedings of WWW*, pages 325–332, Honolulu, Hawaii, USA, 2002.

[40] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Query expansion by mining user logs. *IEEE TRANSACTION ON KNOWLEDGE AND DATA ENGINEERING*, 15(4):829–839, 2003.

[41] Van Dang and Bruce W. Croft. Query reformulation using anchor text. In *Proceedings of WSDM*, pages 41–50, New York, USA, 2010.

[42] Van Dang and Bruce W. Croft. Term level search result diversification. In *Proceedings of SIGIR*, pages 603–612, Dublin, Ireland, 2013.

[43] Van Dang and W. Bruce Croft. Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of SIGIR*, pages 65–74, Portland, USA, 2012.

[44] Atish Das Sarma, Sreenivas Gollapudi, and Samuel Ieong. Bypass rates: Reducing query abandonment using negative inferences. In *Proceedings of KDD*, pages 177–185, New York, USA, 2008.

[45] Elena Demidova, Peter Fankhauser, Xuan Zhou, and Wolfgang Nejdl. Divq: Diversification for keyword search over structured databases. In *Proceedings of SIGIR*, pages 331–338, Geneva, Switzerland, 2010.

[46] Ting Deng and Wenfei Fan. On the complexity of query result diversification. *Journal of VLDB Endow.*, 6(8):577–588, 2013.

[47] Romain Deveaud, Eric SanJuan, and Patrice Bellot. Estimating topical context by diverging from external resources. In *Proceedings of SIGIR*, pages 1001–1004, Dublin, Ireland, 2013.

[48] Fernando Diaz and Donald Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of SIGIR*, pages 154–161, Seattle, Washington, USA, 2006.

[49] David L. Donoho. For most large underdetermined systems of linear equations the minimal ?1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math*, 59:797–829, 2004.

[50] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of WWW*, pages 581–590, New York, USA, 2007.

[51] Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. Multi-dimensional search result diversification. In *Proceedings of WSDM*, pages 475–484, Hong Kong, China, 2011.

[52] Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *Proceedings of KDD*, pages 289–298, Paris, France, 2009.

[53] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI*, pages 1606–1611, San Francisco, USA, 2007.

[54] Veronica Gil-Costa, Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Modelling efficient novelty-based search result diversification in metric spaces. *Journal of Discrete Algorithms*, 18:75–88, 2013.

[55] William Goffman. A searching procedure for information retrieval. *Information Storage and Retrieval*, 2(2):73 – 78, 1964.

[56] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *Proceedings of WWW*, pages 381–390, Madrid, Spain, 2009.

[57] Michael D. Gordon and Peter Lenk. When is the probability ranking principle suboptimal? *Journal of the American Society for Information Science*, 43(1):1–14, 1992.

[58] Jayant R. Haritsa. The kndn problem: A quest for unity in diversity. *IEEE Data(base) Engineering Bulletin*, 32:15–22, 2009.

[59] Mahbub Hasan, Abdullah Mueen, Vassilis Tsotras, and Eamonn Keogh. Diversifying query results on semi-structured data. In *Proceedings of CIKM*, pages 2099–2103, Maui, Hawaii, USA, 2012.

[60] Jiyin He, Edgar Meij, and Maarten de Rijke. Result diversification based on query-specific cluster ranking. *Journal of the American Society of Information Science and Technology*, 62 (3):550–571, 2011.

[61] Jiyin He, Vera Hollink, and Arjen de Vries. Combining implicit and explicit topic representations for result diversification. In *Proceedings of SIGIR*, pages 851–860, Portland, USA, 2012.

[62] Dzung Hong and Luo Si. Search result diversification in resource selection for federated search. In *Proceedings of SIGIR*, pages 613–622, Dublin, Ireland, 2013.

[63] Ming-Hung Hsu and Hsin-Hsi Chen. Information retrieval with commonsense knowledge. In *Proceedings of SIGIR*, pages 651–652, Seattle, Washington, USA, 2006.

[64] Ming-Hung Hsu, Ming-Feng Tsai, and Hsin-Hsi Chen. Query expansion with conceptnet and wordnet: An intrinsic comparison. In *Proceedings of AIRS*, pages 1–13, Singapore, 2006.

[65] Ming-Hung Hsu, Ming-Feng Tsai, and Hsin-Hsi Chen. Combining wordnet and conceptnet for automatic query expansion: A learning approach. In *Proceedings of AIRS*, pages 213–224, Harbin, China, 2008.

[66] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. Understanding user's query intent with wikipedia. In *Proceedings of WWW*, pages 471–480, New York, NY, USA, 2009.

[67] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of CIKM*, pages 2333–2338, Burlingame, USA, 2013.

[68] Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the user intent of web search engine queries. In *Proceedings of WWW*, pages 1149–1150, New York, NY, USA, 2007.

[69] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323. 2013.

[70] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.

[71] Diane Kelly. Methods for evaluating interactive information retrieval systems with users. *Found. Trends Inf. Retr.*, 3:1–224, 2009.

[72] Diane Kelly, Karl Gyllstrom, and Earl W. Bailey. A comparison of querysuggestion features for interactive searching. In *Proceedings of SIGIR*, pages 371–378, Boston, MA, USA, 2009.

[73] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[74] Alexander Kotov and ChengXiang Zhai. Tapping into knowledge base for concept feedback: Leveraging conceptnet to improve search results for difficult queries. In *Proceedings of WSDM*, pages 403–412, Seattle, Washington, USA, 2012.

[75] Robert Krovetz. Viewing morphology as an inference process. In *Proceedings of SIGIR*, pages 191–202, Pittsburgh, Pennsylvania, USA, 1993.

[76] Tessa Lau and Eric Horvitz. Patterns of search: Analyzing and modeling web query refinement. In *Proceedings of UM*, pages 119–128, Banff, Canada, 1999.

[77] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of SIGIR*, pages 120–127, New Orleans, Louisiana, USA, 2001.

[78] Uichin Lee, Zhenyu Liu, and Junghoo Cho. Automatic identification of user goals in web search. In *Proceedings of WWW*, pages 391–400, New York, NY, USA, 2005.

[79] Xiao Li, Ye-Yi Wang, and Alex Acero. Learning query intent from regularized click graphs. In *Proceedings of SIGIR*, SIGIR '08, pages 339–346, New York, NY, USA, 2008.

[80] Yinghao Li, Robert Wing Pong Luk, Edward Kei Shiu Ho, and Korris Fu-Lai Chung. Improving weak ad-hoc queries using wikipedia as external corpus. In *Proceedings of SIGIR*, pages 797–798, Amsterdam, Netherlands, 2007.

[81] Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. Fusion helps diversification. In *Proceedings of SIGIR*, pages 303–312, Gold Coast, Queensland, Australia, 2014.

[82] H. Liu and P. Singh. Conceptnet a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, 2004.

[83] Kun Liu, Evimaria Terzi, and Tyrone Grandison. Highlighting diverse concepts in documents. In *Proceedings of SDM*, pages 545–556, Nevada, USA, 2009.

[84] Xiaohua Liu, Arbi Bouchoucha, Alessandro Sordoni, and Jian-Yun Nie. Compact aspect embedding for diversified query expansions. In *Proceedings of AAAI*, pages 115–121, Quebec City, Canada, 2014.

[85] Yuanhua Lv and ChengXiang Zhai. Positional relevance model for pseudo-relevance feedback. In *Proceedings of SIGIR*, pages 579–586, Geneva, Switzerland, 2010.

[86] Hao Ma, Michael R. Lyu, and Irwin King. Diversifying query suggestion results. In *Proceedings of AAAI*, pages 1399–1404, Atlanta, USA, 2010.

[87] Gary Marchionini. Exploratory search: From finding to understanding. *Commun. ACM*, 49: 41–46, 2006.

[88] Donald Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Journal of Inf. Retr.*, 10(3):257–274, 2007.

[89] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Journal of CoRR*, 2013.

[90] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119. 2013.

[91] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[92] Alistair Moffat and Justin Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Syst.*, 27(1):2:1–2:27, 2008.

[93] Tu Ngoc Nguyen and Nattiya Kanhabua. Leveraging dynamic query subtopics for time-aware search result diversification. In *Proceedings of ECIR*, Lecture Notes in Computer Science, pages 222–234, Amsterdam, Netherlands, 2014.

[94] Ahmet Ozdemiray and Ismail Altingovde. Query performance prediction for aspect weighting in result diversification. In *Proceedings of CIKM*, pages 871–874, Shanghai, China, 2014.

[95] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, 1999.

[96] Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. In *Proceedings of SIGIR*, pages 691–692, New York, USA, 2006.

[97] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of ICML*, pages 784–791, Helsinki, Finland, 2008.

[98] Davood Rafiei, Krishna Bharat, and Anand Shukla. Diversifying web search results. In *Proceedings of WWW*, pages 781–790, Raleigh, North Carolina, USA, 2010.

[99] Karthik Raman, Pannaga Shivaswamy, and Thorsten Joachims. Online learning to diversify from implicit feedback. In *Proceedings of KDD*, pages 705–713, Beijing, China, 2012.

[100] S. E. Robertson. Readings in information retrieval. chapter The Probability Ranking Principle in IR, pages 281–286. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[101] Daniel E. Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of WWW*, pages 13–19, New York, NY, USA, 2004.

[102] Tetsuya Sakai. Metrics, statistics, tests. In *Bridging Between Information Retrieval and Databases*, Lecture Notes in Computer Science, pages 116–163. Springer Berlin Heidelberg, 2014.

[103] Rodrygo L. Santos, Craig Macdonald, and Iadh Ounis. On the role of novelty for search result diversification. *Inf. Retr.*, 15(5):478–502, 2012.

[104] Rodrygo L. T. Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. Explicit search result diversification through sub-queries. In *Proceedings of ECIR*, pages 87–99, Milton Keynes, UK, 2010.

[105] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of WWW*, pages 881–890, Raleigh, USA, 2010.

[106] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Selectively diversifying web search results. In *Proceedings of CIKM*, pages 1179–1188, Toronto, ON, Canada, 2010.

[107] Mark Schmidt, Glenn Fung, and RèżŁmer Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *Proceedings of ECML*, pages 286–297, Warsaw, Poland, 2007.

[108] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[109] Kai Song, Yonghong Tian, Wen Gao, and Tiejun Huang. Diversifying the image retrieval results. In *Proceedings of MULTIMEDIA*, pages 707–710, Santa Barbara, CA, USA, 2006.

[110] Ruihua Song, Zhenxiao Luo, Jian-Yun Nie, Yong Yu, and Hsiao-Wuen Hon. Identification of ambiguous queries in web search. *Inf. Process. Manage.*, 45(2):216–229, 2009.

[111] Yang Song, Dengyong Zhou, and Li-wei He. Post-ranking query suggestion by diversifying search results. In *Proceedings of SIGIR*, pages 815–824, Beijing, China, 2011.

[112] Alessandro Sordoni, Jian-Yun Nie, and Yoshua Bengio. Modeling term dependencies with quantum language models for ir. In *Proceedings of SIGIR*, pages 653–662, Dublin, Ireland, 2013.

[113] Robert Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. In *Proceedings of LREC*, Istanbul, Turkey, 2012.

[114] Robert M. Losee Jr. Stephanie W. Haas. Looking in text windows: Their size and composition. *Information Processing and Management*, 30(5):619 – 629, 1994.

[115] Markus Strohmaier, Mark Kröll, and Christian Körner. Intentional query suggestion: Making user goals more explicit during search. In *Proceedings of WSCD*, pages 68–74, Barcelona, Spain, 2009.

[116] Rooh Ullah and Jeefer Jaafar. Exploiting query expansion through knowledgebases for images. In *Proceedings of IVIC*, pages 93–103, Selangor, Malaysia, 2011.

[117] Rooh Ullah and Jeefer Jaafar. Exploiting short query expansion for images retrieval. In *Proceedings of ICCIS*, pages 352–356, Kuala Lumpur, Malaysia, 2012.

[118] Rooh Ullah and Jeefer Jaafar. Queries snippet expansion for efficient images. In *Journal of Theoritical and Applied Information Technology*, pages 23–28, 2012.

[119] Saúl Vargas, Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Selecting effective expansion terms for diversity. In *Proceedings of OAIR*, pages 69–76, Lisbon, Portugal, 2013.

[120] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *Proceedings of SIGIR*, pages 115–122, Boston, MA, USA, 2009.

[121] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009.

[122] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of SIGIR*, pages 4–11, Zurich, Switzerland, 1996.

[123] Yunjie Xu and Hainan Yin. Novelty and topicality in interactive information retrieval. *Journal of the Amercian Society for Information Science and Techology*, 59(2):201–215, 2008.

[124] Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural svms. In *Proceedings of ICML*, pages 1224–1231, Helsinki, Finland, 2008.

[125] Cheng Xiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of SIGIR*, pages 10–17, Toronto, Canada, 2003.

[126] Chengxiang Zhai. Risk minimization and language modeling in text retrieval. Technical report, Carnegie Mellon University, 2002.

[127] ChengXiang Zhai. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, 2008.

[128] ChengXiang Zhai and John Lafferty. A risk minimization framework for information retrieval. *Information Processing and Management*, 42:31 – 55, 2006.

[129] Yi Zhang, Jamie Callan, and Thomas Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of SIGIR*, pages 81–88, Tampere, Finland, 2002.

[130] Wei Zheng, Xuanhui Wang, Hui Fang, and Hong Cheng. Coverage-based search result diversification. *Journal of Information Retrieval*, 15(5):433–457, 2011.

[131] Wei Zheng, Hui Fang, and Conglei Yao. Exploiting concept hierarchy for result diversification. In *Proceedings of CIKM*, pages 1844–1848, Maui, Hawaii, USA, 2012.

[132] Xiaojin Zhu, Andrew Goldberg, Jurgen Van Gael, and David Andrzejewski. Improving diversity in ranking using absorbing random walks. In *Proceedings of HLT-NAACL*, pages 97–104, 2007.

[133] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of WWW*, pages 22–32, Chiba, Japan, 2005.