جامعة بيرزيت
BIRZEIT UNIVERSITY
Faculty of Engineering and Technology
Master Program of Computing

Arabic Search Results Disambiguation: A Supervised Approach to Unsupervised Learning

Master's Thesis in Computing

By:

Haytham Salhi

Supervised by:

Dr. Adnan Yahya
and
Dr. Radi Jarrar

This Thesis was submitted in partial fulfillment of the requirements for the Master's Degree in Computing from the Faculty of Engineering and Technology at Birzeit University, Palestine

January 12, 2019

Statement in Lieu of an Oath

I hereby declare that I have written this thesis on my own and have not used any other resources/materials other than the ones referred to in this thesis.

Declaration of Consent

I gladly agree to make my thesis publicly accessible by having it added to Birzeit's library or its online libraries.

January 12, 2019 Haytham Salhi

Abstract

Web search engines aim at retrieving relevant results as a response to a given query, or more precisely an information need. However, the query can be ambiguous, which means it might refer to different meanings or senses. Search results clustering (SRC) is a powerful approach that dynamically attempts to find groups of sense-relevant results. The preprocessing stage of SRC highly affects the effectiveness, and though there is a lot of research on SRC, the research has not yet clearly shown the best source from which features could be selected nor the best representation by which features could be represented. Moreover, a little amount of research, with the lack of Arabic datasets, has been paid to Arabic.

The major contributions of this thesis are fourfold: 1) It examines the influence of feature source (i.e., title, snippet, etc.) and feature representation on the effectiveness of SRC, figuring out the best combination that results in a high-quality clustering of Arabic Web search results. 2) It introduces a set of benchmarks for Arabic, called AMBIGArabic, and a new framework, called Spread, for data labeling, search results acquisition, and performing SRC experiments. 3) It shows how useful the blind relevance feedback concept is in SRC. 4) Lastly, it proposes a new SRC approach, called SAUL, along with an implementation of this approach based on Wikipedia as a source of the senses. The results show that feature sources and feature representations significantly affect the effectiveness of SRC, and combinations like (title with snippet, single words) and (title with snippet, single words with 2-gram and 3-gram words) are amongst the best. Also, by comparing the best combinations, the proposed approach outperforms the baseline approach.

Acknowledgements

محضُ فضلٍ وتوفيقٍ منَ الله. الحمدُ للهِ حمداً كثيراً طيباً مباركاً فيه.

Throughout the thesis's journey, I have gained various skills that, I do believe, will help me improve both my technical and research capabilities. As a result, I would like to express my gratitude and appreciation to Dr. Adnan Yahya and Dr. Radi Jarrar for giving me such exclusive opportunity to work under their supervision and for their valuable advice throughout the thesis. I would also like to show my appreciation to Dr. Eibe Frank, one of the authors of Weka framework, for his quick support and aid in resolving issues in Weka that appeared during the thesis work.

I would like to show my sincere appreciation to all doctors who taught me throughout the master program including: Dr. Adel Taweel, Dr. Mamoun Nawahda, Dr. Samer Zain, and Dr. Wasel Ghanem. Each one of them has added much to my knowledge that I hope to extend and pass out to others as well. I would also like to thank Dr. Abed Sayyad, Dr. Majdi Mafarja, and Dr. Thaer Samar for their participation in reviewing the thesis.

I would like to thank my colleagues in work for their support and taking the time to help me label data manually for the thesis. Data collection would not have been possible without them. I would also like to thank Feras Kayyali and Mohammad Arouri for facilitating my research during my work at Harri. I would like to thank Fadi Abu Farha, Feras odeh, Anas Najjar, Yasser Arafat, and Ali Hamed for their continuous support.

On the personal level, words will never be enough to express how I am grateful to my family including: my grandfather, parents, brothers, sister, and fiancee for their sincere support. I do highly appreciate their patience towards my continuous absence.

I would like to thank all my friends at Birzeit. I am lucky, I believe, being surrounded by all those intelligent, caring, and enthusiastic people.

Finally, I would like to extend my sense of appreciation to everyone who expressed his support and/or made duaa for me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Search results disambiguation is the notion of how to disambiguate search results that are retrieved upon a user query having different word senses[3]. Thousands of documents could be returned by a web search engine in response to an ambiguous or imprecise query. A query for “Python”, as an example, will return documents pertaining to the programming language, as well as to the snake, and the movie. Historically, this is very problematic, making it difficult for users to browse or identify relevant search results. This thesis intends to address this issue by providing a mechanism to cluster search results into different senses. This chapter introduces the thesis by starting with the motivation of this work, identifying the problem, stating the research gaps the thesis attempts to fill, briefly presenting the research methodology, and highlighting proposed solutions and contributions. Lastly, this chapter outlines the whole thesis report.

1.1 Motivation

Over the recent years, Web search engines have become an important part of our everyday lives as they make the lookup for an information rather easy and trivial. When a user poses a query, traditional Web search engines return a list of ranked results, often ordered by relevance to the query. Then, the user starts looking at the top results and goes down, until the needed information is found. This is indeed very useful when the query conveying the information is *clear* and *precise*. However, traditional Web search engines might be less effective when dealing with *ambiguous* or *broad* queries (i.e., queries that have more than one meaning or cover a variety of subtopics, respectively [4]) because the list of results would very likely contain results on different subtopics or meanings, thus making users walk through many irrelevant results.

Therefore, helping users find results satisfying their information needs, in the light of ambiguous queries, is of a great importance. To this end, researchers in academia and industry [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17], have proposed many approaches that mitigate or solve this problem. These approaches relate to many disciplines such as Information Retrieval (IR), Machine Learning (ML), Natural Language Processing (NLP), and Human Computer Interaction (HCI).

1.2 Research Problem

One approach to search results disambiguation is *search results clustering* (SRC). It aims at grouping search results into clusters, each representing the results for one sense of the query. Typically, this is achieved by applying a clustering algorithm to search results returned by a Web search engine [5, 6, 7, 8, 9, 10]. Another one of the early approaches is based on the idea of classifying the entire Web pages, which results in Web categories and directories such as DMOZ project[1] (previously known as Open Directory Project). Even though it might be a good idea, for a Web search engine it is hard to maintain a large number of Web pages with respect to human labeling. Other approaches [14, 13, 15] tried to exploit some of natural language processing tasks such as word sense induction and word sense disambiguation. However, such approaches are usually based on knowledge repositories, which might cause additional overhead to the process of grouping. Other researchers [18, 19, 12] used a totally different approach that presents search results by achieving the maximum diversity between them, using what-so called diversification techniques. Others [20, 16, 17] exploit the query logs to find aspects of ambiguous queries by applying mining techniques. However, the last two approaches are usually studied and investigated by researchers who own such data (i.e., the query logs) like Google and Microsoft.

With all these approaches, this research looks into the SRC approach. This approach is challenged by three core requirements [21, 1]:

1. The clusters or groups should be of good quality (i.e., the degree to which search results in a cluster belong to same meaning) which is represented by the effectiveness of clustering.

2. The cluster labels must be understandable.

3. The clustering process must be efficient in terms of processing time needed to generate the clusters.

Each of these challenges is considered as a self-contained research problem. Therefore, the first requirement (i.e., effectiveness) represents the scope of this research. An important factor highly affecting the effectiveness is feature generation and space representations [21, 1, 22], which is often done in the preprocessing stage of search results clustering. Moreover, the preprocessing stage highly depends on the language being processed.

Knowing that most of proposed work has been done for the English, and though there has been increasing interest in studies [23, 24, 25, 26] concerning search results clustering for Arabic, the gaps that this research addresses can be summarized as follows:

1. In contrast to English, there is still no published benchmarks for Arabic, designated for performing experiments of search results clustering.

[1] `https://www.dmoz.org`

2. There is no clear evidence that shows the source from which features are best selected nor the best space representation in vector space model, resulting in a high quality of search results clustering (in terms of effectiveness).

3. A big challenge still exists that relates to how to design a suitable model for clustering that competes with other traditional approaches in terms of clustering effectiveness [27].

1.3 Research Objectives and Methodology

The main objective of this research is to address some of the research gaps related to search results disambiguation for Arabic. Given some of the research gaps brought up in the previous paragraph, this research aims to:

1. Produce a corpus of Arabic Web search results that can be used to evaluate any search results clustering method.

2. Build a framework for conducting search results clustering experiments that is capable of fetching search results from different search engines like Google, Bing and Yahoo! and labeling search results with different approaches including a human assessment interface. This makes the running of experiments smooth and reproducible.

3. Study the influence of different feature sources and space representations on the effectiveness of search result clustering.

4. Propose a new clustering model for improving the effectiveness of search results clustering and as a solution to the problem of search results disambiguation.

To achieve that, this study followed the scientific method of research by conducting an experimental research to observe the influence of feature sources and space representations and to look into baseline and proposed approaches. The method fundamentally involves:

- Collecting data: one of the challenges in this study is to collect real Web search results in Arabic as there are no built benchmarks for that purpose according to best of our knowledge. English, in contrast, has already-built datasets for that purpose. The details of how data is collected, how the benchmarks are built, and how they are labeled are detailed in Chapter 4.

- Experimental design: one of the important things in experimental research is to specify the research hypothesis, the independent variables of interest, the dependent variable, and the neutralized ones. This is detailed in Chapter 5.

- Running experiments based on the design by building a framework to facilitate conducting and running the experiments.

- Evaluating the results against labeled search results to find the best of feature sources and space representations and to compare the traditional approach (i.e., the baseline) with the proposed approach. All evaluation metrics and results are detailed in Chapter 7.

1.4 Proposed Solution: AMBIGArabic, Spread, and SAUL

The contributions of this thesis are represented in the proposed solutions that mitigate some of the research gaps mentioned above in Section 1.2. In particular, this research has two major contributions:

1. It proposes a thorough experimental design that investigates the influence of feature sources and space representations on effectiveness of Arabic search results disambiguation and finding out the best combination for the clustering approach.
2. A new model (called SAUL) is proposed as a solution to search results disambiguation that augments a supervised approach into unsupervised learning and leverages the concept of blind relevance feedback (BRF) as well as clear queries. Additionally, a fully working demonstration of the SAUL approach is implemented by fetching meanings/senses from Wikipedia Disambiguation Pages (WDP), then fetching search results for each sense, and then building a supervised model based on those results and using the blind relevance feedback.

Other contributions include:

1. Building a crawler component that fetches search results for any query from different search engines like Google, Bing, and Yahoo!.
2. Developing a framework (called Spread) for performing search results clustering experiments that generates clustering results along with evaluations and graphs, and supports many parameters related to many aspects like data preprocessing, vector space representation, and algorithm-specific parameters.
3. Building a labeled corpus for Arabic search results (called AMBIGArabic) from Google and Bing, which can be used to evaluate any search results grouping method.
4. Proposing new approaches that assist in labeling the search results as well as building a human relevance assessment interface for labeling the search results by humans as a part of the framework.
5. Improving the evaluation strategy that is used in Weka for unsupervised learning as well as fixing a lot of Weka bugs[2].

[2] During the course of thesis work, about four critical bugs were reported and solved closely with Weka authors. These bugs are related to issues in ClassificationViaClustering components. Consequently, five minor releases were pushed to maven repository `https://mvnrepository.com/artifact/nz.ac.waikato.cms.weka/classificationViaClustering`. Their release notes can be found at `http://weka.sourceforge.net/packageMetaData/classificationViaClustering/index.html`

1.5 Outline

The rest of this report is structured as follows. Chapter 2 presents the main concepts and foundations. Chapter 3 discusses the approaches of search results disambiguation that have been developed in literature as well as related studies on Arabic. The remaining chapters represents the contributions of this thesis. Chapter 4 presents the data collection, showing the proposed AMBIGArabic benchmarks. The experimental design and methodology are discussed in Chapter 5. The Spread framework is explained in Chapter 6. Chapter 7 shows evaluation and statistics of experiments as well as main findings. Finally, chapter 8 concludes the report and plans for future work.

Chapter 2

Background

Dealing with search results disambiguation and search results clustering requires a basic knowledge of topics like information retrieval, machine learning, and natural language processing. This chapter presents the core concepts and definitions in those topics that are required to understand the content of this study.

This begins with a brief introduction of information retrieval and machine learning. It then explains the notion of queries and information needs, then an overview of clustering and how it is typically evaluated. Finally, this chapter goes through a quick overview of feature selection and feature extraction, and how they are perceived in the context of this work.

2.1 Information Retrieval and Machine Learning

For most people, looking up information on the Web has become a daily activity. Because search is one of the popular uses of the Internet, many people in academia and industry are trying to come up with easier and faster ways to improve the process of finding the right information [1]. The field that these people are working in is called *Information Retrieval.*

A good definition proposed by Gerard Salton [28], a pioneer in information retrieval, states: "Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information". Two important concepts of the information retrieval in this research, *query* and *information need*, are discussed next in Section 2.2.

In fact, the field of information retrieval considerably overlaps the field of machine learning. The ranking of documents was improved by introducing a technique called relevance feedback, which is based on user feedback about the relevance of documents. This was achieved by using a simple machine learning algorithm that produced a classifier to separate relevant from non-relevant documents [1]. Machine learning approaches are used by information retrieval researchers for many purposes like learning ranking algorithms, development of sophisticated statistical models of text, or even document categorization/clustering [1].

The idea of making computer systems learn, without being explicitly programmed, using statistical technologies is the heart of machine learning field [29]. The algorithms in machine learning are often characterized as supervised or unsupervised [1, 29].

In *supervised learning*, a model is built using a set of fully labeled data. This set is often called the *training dataset.* After the model is built, it can then be applied to a set of unlabeled data, which is often called the *test dataset*, to automatically assign labels. The problem of classification is often considered as a supervised learning task [1, 29]. As an example, given a set of emails labeled as "spam" or "ham", a classification model can be built based on these labeled data. Then this model can be used to automatically classify unseen emails as "spam" or "ham".

In *unsupervised learning*, on the other hand, the algorithms learn completely based on unlabeled data. Clustering is the most common task in unsupervised learning [1, 29]. As it will be shown further in Section 2.3, a clustering algorithm takes a set of unlabeled data as input and group the items based on some notion of similarity [1, 29].

2.2 Queries and Information Needs

Even though the search engine index (i.e., the place where all documents a search engine has collected is stored in) and ranking algorithms are the main components in any search engine, from a user's point of view, the search engine is basically an interface for entering *queries* and seeing *results.* One of the important things is to distinguish between *query* and *information need* when it comes to information retrieval. They are related to each other but represent two different concepts. Information need represents the required information a user is looking for. In other words, it represents the information that is in user's mind. Query is the actual words that are written by users to express their information need [1].

Given the fact that in some cases it could be difficult for people to exactly define what their information need is due to a gap in their knowledge, a query can represent different information needs and consequently might require different techniques and ranking algorithms to obtain the best ranking. Moreover, the query can happen to be a poor representation of the information need because the user might find it difficult to express their information needs, or more often the user is encouraged to enter queries with small number of words, leaving the query *ambiguous* or *imprecise* [1].

2.3 Clustering

Two important machine learning tasks are widely used in information retrieval tasks, *classification* and *clustering.* These two tasks have many features in common and can be useful for ranking documents [1]. On one hand, classification is concerned with automatically labeling data like emails, web pages, or images based on historical data (i.e., labeled data) on which a classification model is built. Clustering, on the other hand, is concerned with grouping similar items together, resulting in one or more clusters which do not necessarily correspond to a useful label or meaning [1].

Unlike classification, clustering algorithms are based on unsupervised learning, meaning that they do not require any training dataset. Clustering algorithms take a set of unlabeled items as input and group (cluster) them based on some notion of similarity [1, 29]. While classification has very clear objectives, clustering is often an ill-defined problem. The decision whether the resulting clustering is good, is often defined very subjectively [1].

2.3.1 Clustering Algorithms

Many algorithms exist for clustering. They differ primarily in their definition of what constitutes clusters and how to efficiently find them. One of the reasons why there are many clustering algorithms is that the notion of cluster itself cannot be precisely defined [30]. Because researchers can employ different clustering models and for each of these models different algorithms can be given, clustering algorithms can be classified into different types such as hierarchical clustering, centroid-based clustering, distribution-based clustering, and density-based clustering [30]. A brief description of the hierarchical clustering is given below. Additionally, since the K-means algorithm is the major theme of this study, the next discussion of centroid-based clustering is limited to K-means algorithm.

2.3.1.1 Hierarchical Clustering

Algorithms falling into this type build clusters in a hierarchical fashion. These algorithms are often grouped into two types, depending on how the algorithm works. The first type is called *divisive algorithms*, which begins with a single cluster containing all instances. In each iteration, it selects an exiting cluster and divides it into more clusters.

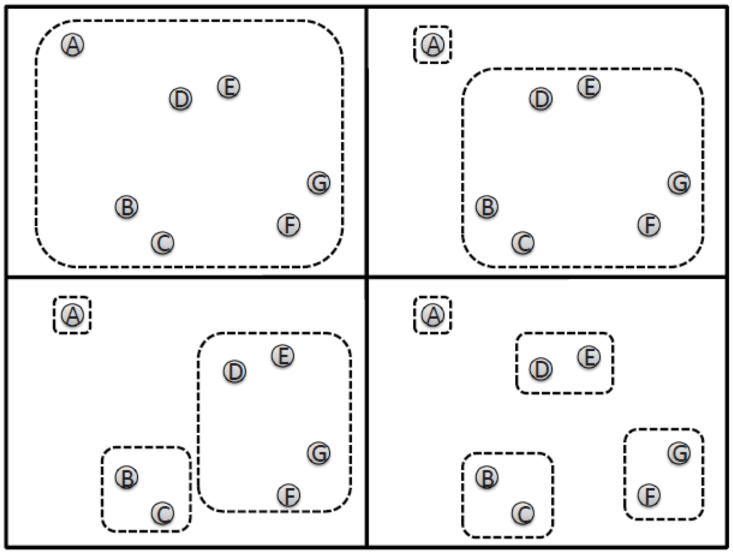


Figure 2.1: An illustration of divisive clustering with $K = 4$ [1].

This process is repeated until having a total of K clusters, where K is a given number. Figure 2.1 [1] gives an example of divisive clustering with $K = 4$. The clustering starts and proceeds from left to right and top to bottom, resulting in 4 clusters.

A similar process happens in the other type, *agglomerative algorithms*, which follows a bottom-up approach. An agglomerative algorithm starts with each input as a separate cluster. Then it proceeds by joining more than one existing clusters to form a new cluster [1]. Figure 2.2 [1] gives an example of agglomerative clustering with $K = 4$. The clustering starts and proceeds from left to right and top to bottom, resulting in 4 clusters.

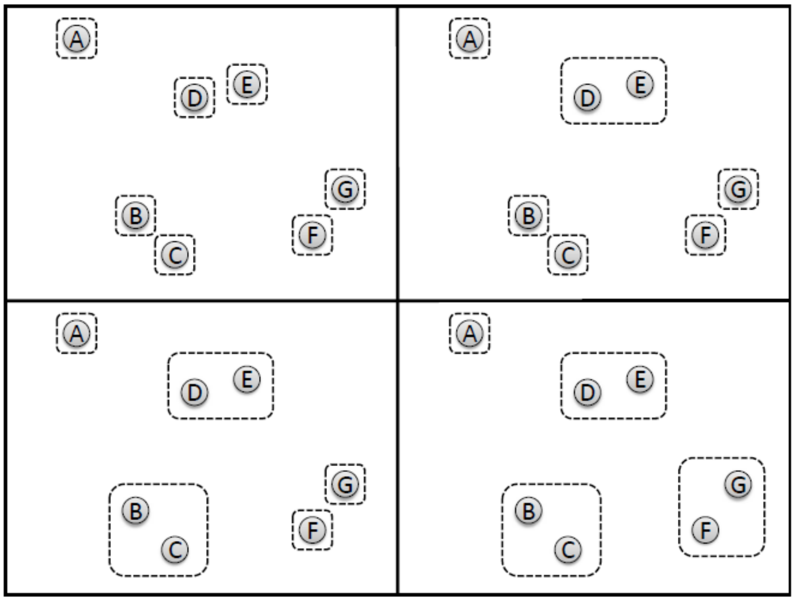


Figure 2.2: An illustration of agglomerative clustering with $K = 4$ [1].

2.3.1.2 Centroid-based Clustering: K-means

This type of algorithms is fundamentally different than the hierarchical clustering previously described. In contrast to hierarchical clustering, the number of clusters in K-means never changes. That is, the algorithm start with K clusters and ends with same number of clusters [1]. A formal definition is usually given as an optimization problem: “find the K cluster centers and assign the objects to the nearest cluster center such that the squared distances from the cluster are minimized” [30]. Therefore, the main goal of the K-means algorithm is to find the cluster assignment vectors $A[1], A[2], ..., A[N]$, that minimize the following cost function [1]:

$$COST(A[1], A[2], ..., A[N]) = \sum_{k=1}^{K} \sum_{i:A[i]=k} dist(X_i, C_k) \tag{2.1}$$

where $dist(X_i, C_k)$ is the distance between instance X_i and class C_k. This distance measure can be any reasonable distance measure. One of the most common measures is the Euclidean distance, as given below [1]:

$$dist(X_i, C_k) = ||X_i - \mu_{C_k}||^2 = (X_i - \mu_{C_k}).(X_i - \mu_{C_k}) \tag{2.2}$$

where μ_{C_k} is the centroid of cluster C_k.

The cosine similarity between X_i and μ_{C_k} can be used instead as the distance measure, especially for some text applications, since it has been shown to be more effective than Euclidean distance [1].

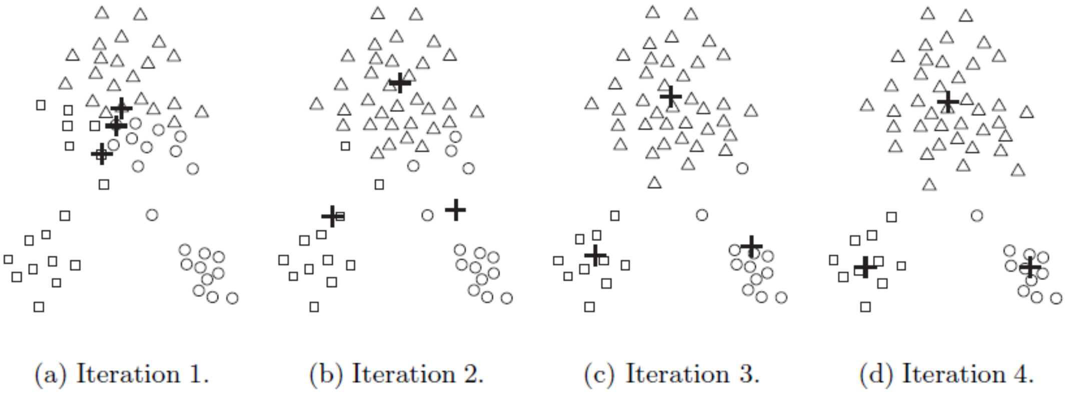
The main steps of K-means algorithm are described by Algorithm 1 [2]. Figure 2.3 [2], which shows how the final clusters are found in four iterations, illustrates the operation of K-means algorithm. Each subfigure shows:

1. The centroids (indicated by the "+" symbol) at the start of the iteration.
2. The assignments of the points to these centroids; all points that are in the same cluster have the same marker shape.

Algorithm 1 K-means algorithm [2].

```
Select $K$ points as initial centroids.
repeat
    Form $K$ clusters by assigning each point to its closet centroid.
    Recalculate the centroid of each cluster.
until Centroids do not change.
```

In the first iteration, the algorithm assigns points, which are all in the larger group, to the initial centroids. After the algorithm assigns points to a centroid, the centroid is then updated. In the second iteration, the algorithm assigns points to the updated centroids, and the centroids are updated again. In iterations 2, 3, and 4, shown in Figure 2.3 (b), (c), and (d), respectively, two centroids move to the two small groups at the bottom [2].



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

Figure 2.3: Finding three clusters using the K-means algorithm [2].

Finally, because no more changes occur, the K-means algorithm terminates, and the centroids have identified the groupings of points [2]. K-means always converges to a solution for some combinations of distance functions and types of centroids. In other words, K-means reaches a state in which the centroids no longer change. Since most of convergence occurs in the early steps of the algorithm, however, the condition in the until statement of Algorithm 1 above is typically replaced by a weaker condition [2], for example, repeat until 1.5% of the points change clusters.

One big advantage for K-means is that, when compared to hierarchical clustering, K-means is more efficient. In particular, $K \times N$ distance computations are needed in each iteration, and the number of iterations is often small. Therefore, implementations of K-means algorithm are $\mathcal{O}(KN)$, which is much better than $\mathcal{O}(N^2)$ complexity of hierarchical methods. In practice, K-means algorithm tends to converge very quickly to a solution. Although it is not guaranteed to find the optimal solution, the solution is often optimal or close to optimal [1]. Moreover, although the final clusters produced by the algorithm depend on the initial seed (i.e., the starting points chosen as initial clusters) and on the ordering of the input data, K-means generally produces clusters of similar quality to hierarchical methods [1]. Given these facts, K-means is a good choice for a wide range of information retrieval-related tasks, especially for large datasets [1].

2.3.2 Clustering Evaluation

Evaluation in clustering is not as comprehensive as the evaluation of classification and thus can be challenging. Since clustering is an unsupervised learning, there is often little or no labeled data to use for the evaluation [2, 1].

When there is no labeled training data, *internal evaluation* or *unsupervised evaluation* is used to evaluate clustering model. If there is labeled data, however, *external evaluation* or *supervised evaluation* is used [2]. In internal evaluation, the clustering output is evaluated based on the data that was originally clustered. Internal evaluation methods usually assign the best score to the algorithm producing clusters with high similarity within a cluster and low similarity between clusters. Examples of such methods include: davies-bouldin index, dunn index, and silhouette coefficient. Though these methods are well-suited to give some insights into situations where one algorithm performs better than another, two main drawbacks exist [2, 30]:

- Getting a high score does not necessarily mean that this is an effective information retrieval application.
- The evaluation can be biased towards algorithms that use same objective function. K-means, as an example, naturally optimizes distances, and therefore a distance-based internal criterion will likely overrate the resulting clustering.

If labeled data is available, then it is possible to use slightly modified version information retrieval/classification metrics, such as accuracy (A) (i.e., the proportion of correctly classified predictions), precision (P) (i.e., the proportion of positive instances that are truly positive), Recall (R) (i.e., the proportion of positive instances that are correctly classified), and F-measure (F) (i.e., combines both precision and recall in a single value) [2, 1]. This is referred to as external evaluation. In other words, the clustering output is evaluated based on data that was not used for clustering and often created by humans (i.e., experts). Such data is also called external benchmarks. Examples of such methods include: Rand index, F-measure, Jaccard index, mutual information, and confusion matrix [2, 30].

A popular external evaluation method, which is used in this study, is called classes-to-clusters method [31]. This method evaluates the clustering model based on labeled training datasets by mapping the clusters back onto classes, and then using the standard classification measures such as: accuracy, precision, recall, F-measure, confusion matrix, and many others. The definitions of these metrics are shown in Table 2.1; the assumption is there exist two classes A and B, for illustration. The equations in the table are based on the following definitions[29, 2]:

1. TP (True Positive): Number of of positive instances labeled as such.
2. FP (False Positive): Number of negative instances labeled as positive.
3. TN (True Negative): Number of negative instances labeled as such.
4. FN (False Negative): Number of negative instances, labeled as positive.

Table 2.1: The definition list of external evaluation metrics used in this thesis.

Evaluation Metric	**Equation**
Precision (P)	$\frac{TP}{TP+FP}$
Recall (R)	$\frac{TP}{TP+FN}$
Accuracy (A)	$\frac{TP+TN}{TP+TN+FP+FN}$
F-measure (F)	$\frac{2*P*R}{P+R}$
Weighted Precision	$\frac{ClassSize(A)*P(A)+ClassSize(B)*P(B)}{TotalSize}$
Weighted Recall	$\frac{ClassSize(A)*R(A)+ClassSize(B)*R(B)}{TotalSize}$
Weighted Macro F-measure	$\frac{ClassSize(A)*F(A)+ClassSize(B)*F(B)}{TotalSize}$
Unweighted Macro F-measure	$\frac{F(A)+F(B)}{2}$
Micro F-measure	$\frac{2*P*R}{P+R}$ but for aggregate TP, FP, FN

2.4 Features in Machine Learning

In machine learning, a feature is an individual measurable attribute of a phenomenon being observed. One of the important aspects for effective machine learning algorithms is to choose informative and independent features. Features can be classified into relevant, irrelevant, and redundant [32]. There are two similar but different concepts related to features: *feature selection* and *feature extraction*.

The main goal of feature selection is to select only those input dimensions (i.e., features) that contain relevant information for solving a particular problem, usually resulting in dimensionality reduction [33, 32]. As an example, removing words such as "*the*" might be very useful in improving the clustering results. Feature extraction is a more general idea with the goal of transforming the input space onto low dimensional subspace that preserves most of relevant information [34]. Note that feature extraction and feature selection methods can be used in combination or isolated [32]. Several methods have been proposed to reduce data complexity to a simpler form of information [27]. These include: independent component analysis (ICA) [35] and principal component analysis (PCA) [36].

Since this work is concerned with text clustering, the focus of next discussion is on text feature extraction. Several machine learning algorithms like clustering algorithms work with features with a specific format (e.g., numerical). Thus, this often requires transforming arbitrary data, such as text, into numerical features to be usable for the machine learning algorithm [37]. A popular way to extract numerical features from text is to represent this text in a *vector-space representation* [38]. This process is called sometimes *vectorization* [37], where a collection of text documents is converted into numerical feature vectors. The main idea behind such process is that frequencies of words (or sometimes referred to as terms) are used for analysis. These frequencies are typically normalized with statistical values such as the length of the document. So, a text collection with n documents and "d" terms can be seen as $n \times d$ matrix [38]. This representation strategy is also called *bag-of-words* or *bag-of-n-grams* because the precise ordering of the words is lost in this representation [37, 38].

Chapter 3

Literature Review

The problem of search results disambiguation has been widely studied by the research communities, and many different solutions have been proposed. This chapter introduces the most important work for each approach of search results grouping, starting with search results clustering, which will be the focus of this study. Then it discusses the related work dealing with Web categorization and directories, results diversification, word sense induction and disambiguation, and finally aspect identification. In addition, a brief related work for text feature selection is presented. It then concludes with related work of search results clustering and how text feature extraction is performed for the Arabic language in the last section.

3.1 Search Results Disambiguation

3.1.1 Search Results Clustering

One of the popular solutions to query ambiguity is search result clustering (SRC). The motivation behind using clustering is that search results with same meaning of the posed query are expected to be similar, whereas search results with different meanings are expected to belong to different clusters. The basis for this motivation is the well-known *cluster hypothesis*, as stated by van Rijsbergen (1979): “Closely related documents tend to be relevant to the same requests” [21].

In general, SRC approaches can be divided into two groups, namely data-centric or description-centric [21]. The reasons behind this division are twofold. First, the resulting clusters should be accurate. Therefore, there should be a focus on clustering algorithms that produce high-quality clusters. Second, understandable, comprehensive, and compact cluster labels are considerably important; therefore, in search results clustering, description comes first (A very important conclusion credited to Vivisimo[1]) [21]. These two approaches are reviewed next.

[1] A private technology company specializing in the development of computer search engines

3.1.1.1 Data-centric approaches

The idea of data-centric is to focus on the problem of clustering search results rather than presenting results to users, by using a conventional clustering algorithm (partitional, hierarchical, or other) [21]. These algorithms dealing with search results are often slightly modified to produce more effective results for end users in this context [21]. A more popular approach called Scatter/Gather, developed by Cutting et al. [5], performs an initial clustering of a collection of documents into *k* clusters and, after user selects groups of interest, reclusters the selected groups dynamically. A classic method for this approach is called Buckshot algorithm [5]. Improved methods of this approach, in terms of cluster quality and retrieval performance, have been proposed later such as LAIR2 algorithm [6].

There are a lot of other data-centric approaches out there. Some use agglomerative hierarchical clustering, such as LASSI [22], where a traditional agglomerative hierarchical clustering algorithm is used, with an improved feature selection phase. Other approaches use rough sets model [7] or exploit link information [39].

Despite the power and strengths of data-centric text clustering algorithms in grouping search results, the only drawback is the problem of cluster labeling [21]. In particular, it is difficult to recover the description of a cluster from its feature vectors, where text is likely represented by a *vector space* representation. A keyword-based representation of descriptions is insufficient from the user perspectives. This was the main motivation of making algorithms aware of labeling results, yielding results interpretable to users [21].

3.1.1.2 Description-centric approaches

Other researchers tend to specifically design search results clustering methods that take into account both cluster quality and cluster descriptions. The quality of the latter often comes first, meaning that if a cluster cannot be described, it is probably of no value to users and should be removed [21].

Among these approaches are ones based on suffix trees, which are root directed trees that contain all the suffixes of a string. One of the earliest algorithms, called suffix tree clustering (STC) algorithm, that uses suffix trees, is that developed by Zamir and Etzioni [8] and implemented in a system called Grouper [9]. An edge of a suffix tree represents the label of a non-empty sub-string s and the label of each vertex v is formed by concatenating the edge labels on the path from the root r to v. If the set of strings (i.e., the bag of words) for search result snippets to be clustered are represented using suffix tree, we can consider each vertex as a set of documents that share its phrase (i.e., the label of the vertex). Therefore, one can see that the vertices represent the initial cluster set C_0. The algorithm, with the aim of returning the top k clusters, produces the final clustering by merging similar clusters in C_0 according to a scoring function, defined based on the number of documents in the initial cluster and the length of common phrase [8].

The suffix trees approach received later some improvements. Branson and Greenberg [40] improve the performance and overcome the low scalability of the original approach by using document-to-document similarity scores. Others like Crabtree et al. [41] found out an issue in the original scoring function and proposed the extended suffix tree clustering algorithm (ESTC) with a novel scoring function as well as a new method for selecting the top *k* clusters. Other work based on suffix trees, in order to choose meaningful labels for the clusters, attempts to extract relevant keyphrases from generalized suffix trees [42].

Lingo [10] is one of the descriptive-centric algorithms, especially designed for search results clustering. It was developed as a part of Carrot² framework. In a nutshell, Lingo has four main phases: snippets preprocessing, frequent phrases extraction, label induction, and content allocation. In the first three phases, it attempts to process search results to identify certain dominating topics, based on singular value decomposition. It is worth noting that if a certain vector has no frequent phrase, it would be simply discarded, following the idea introduced at the beginning of this section. The final phase involves that for each frequent phrase, the algorithm allocates search results which contain that frequent phrase. Lingo is monothetic clustering algorithm, i.e., label containment determines document-cluster relationship, where topics are generated by singular value decomposition, and the cluster descriptions are comprehensible because they are extracted directly from search results. However, singular value decomposition is rather computationally expensive [21].

There are many related approaches for search results clustering in the literature. These approaches are based on formal concept analysis [43], spectral clustering [44], spectral geometry [45], link analysis [46], or graph connectivity measure [47]. It is worth noting that most of the work in this area is done without explicit use of lexical semantics.

3.1.2 Web Categorization

One of the earliest approaches to grouping the results appeared in Web 1.0, where the Web is almost based on static web pages, is attempting to manually organize and categorize Web sites [13]. The resulting repositories are often referred to as *Web directories*, whereby the Web sites are listed by category or even by subcategories if possible. A property of these categories is that they are organized as taxonomies. Moreover, anyone can search for some information therein even though they are not real search engines [13]. A popular Web directory was called Open Directory Project (ODP), and it is now called DMOZ[2].

The idea of organizing results into a predefined set of categories is sometimes called *faceted classification*, or simply, *facets* [48]. This idea seems to be useful, especially for specific-domain Web sites (like e-commerce sites). However, building general Web directories suffers from: *(1)* the manual updates to cover new pages and new meanings. *(2)* covering a small portion of the Web. *(3)* classifying Web pages using coarse categories, which makes it difficult to distinguish between instances of the same type.

[2] `https://www.dmoz.org`

There are many methods and techniques developed for classifying Web documents automatically [49, 50, 11]; however, these are usually based on supervised-learning and suffer from reliance on a set of predefined classes. Moreover, Bruza et al. [51] reported that directory-based systems are among the most ineffective solution to information retrieval systems.

3.1.3 Results Diversification

Another approach that deals with query ambiguity and tries to disambiguate search results is called *diversification*. The idea is to rerank top search results based on criteria that maximizes their diversity so that top search results are as different as possible in terms of query senses/meanings. Nowadays, popular search engines, such as Google and Bing, apply such techniques to their top search results.

One of the early diversification algorithms was proposed by Carbonell and Goldstein [52]. It is mainly based on similarity functions to measure the diversity among documents themselves and between document and query. Some researchers tend to use techniques to specify which document is most different from top ranked ones. For example, Chen and Karger [53] proposed a technique that uses conditional probabilities, whereas Zhang et al. [54] ranks using affinity graph, which is based on topic variance and coverage.

While most search results ranking algorithms using heuristics such as global link analysis, user behavior, and content relevance, suffer from information redundancy in returned results, an interesting algorithm called Essential Pages [18] done by Microsoft, addresses this problem by returning a set of essential pages that maximizes the information covered over the total knowledge that exists on the Web about a given query. Another interesting work provides a systematic approach to diversifying results, yet aims to minimize the risk of dissatisfaction of the average user [19]. The authors developed a greedy algorithm that keeps relevance and diversity of the search results balanced.

Other work based on vector space representations have been proposed. Santamaria et al. [55] improved diversity in search results by representing Web page results as vectors and comparing them against Wikipedia pages using cosine similarity, thus adopting Wikipedia as sense inventory.

Other researchers tend to exploit the structure of the Web to diversify search results. Ma et al. [12] proposed approach that leverages Markov random walk and hitting time analysis on the query-URL bipartite graph. Chandar and Cartertte [56] diversify search results by proposing a graph-based method that exploits the links in the Web pages to find a result set of documents that cover various subtopics for a given query.

3.1.4 Word Sense Induction and Disambiguation

Other researchers have proposed to exploit two related but different ideas in attempt to address the query ambiguity issue: *word sense induction* (WSI) and *word sense disambiguation* (WSD) . In particular, in WSI, the key idea is to dynamically discover an inventory of senses of the input query and then use these senses to cluster Web search results returned by a search engine. One of the first attempts was done by Schutze and Perdersen [57], where they showed that vector-based WSI can improve bag-of-words ad hoc information retrieval. Some studies done by Udani et al. [58] and others [14] proved that WSI can benefit Web search result disambiguation. Nguyen et al. [59] provided an interesting work that tries to identify query senses. It makes use of topic analysis models, such as probabilistic latent semantic analysis (PLSA) and latent Dirichlet allocation (LDA) to discover a set of hidden topics, thus improving the clustering quality. These topics are usually estimated from large (universal) data sets, meaning that they are query independent. Marco and Navigli [13], on the other hand, introduced a novel WSI-based approach that clusters Web search snippets after inducing various senses of the ambiguous query dynamically. They leveraged a finding of an exploratory study [60], stating that the majority of relevant uses of ten query words can be identified using graph-based WSI. As a result, Marco and Navigli [13] studied the impact of several graph-based WSI algorithm and integrated them in their clustering framework.

Some other studies used word sense disambiguation in their approaches. In contrast to WSI, the core idea is to use existing word senses (which are usually edited by humans) to cluster Web search results. Even though very little work on this idea exists, one recent work done by Huang et al. [15] is based on Wikipedia disambiguation pages. They improved clustering result by filtering semantically unrelated concepts and assigning search results to relevant topics based on the similarities between concepts in results and topics.

3.1.5 Query Logs Mining

Looking at query logs, one can see that it is a rich source that contains a large number of queries posed by users as well as click-through information. Wang and Zhai [20] proposed that query aspects can be identified by mining those queries that are similar or close to the current input queries. This line of research, so-called aspect identification, has been developed in this field over recent years, attempting to solve query ambiguity.

As a different approach, Wang et al. [16] presented a work that extracts broad latent aspects of a given query from query reformulations found in historical search session logs. Each broad latent aspect represents a set of keywords that convey one sense or one information need. Another interesting work [17] presented named entity topic modeling approach, which aims to discover generic topics for a category of named entities using query logs and click data.

Marco and Navigli [13] pointed out that even though this line of research has commonalities with word sense induction, there are also some differences. Most important, aspect identification discriminates between very fine-grained facets of a given query. WSI, in contrast, induces different meanings or senses of a given query. In addition, the data of query logs and click-through for search engines are often not available to public, thus making it hard to replicate and evaluate experiments, in comparison with other systems.

3.2 Feature Selection in Text Clustering

It is well known that feature selection is a very essential topic in text clustering [27]. This is because it significantly affects performance of clustering in terms of both *effectiveness* (i.e., the degree at which the algorithm produce correct clusters) and *efficiency* (i.e., the speed at which the algorithm produce the clusters) [27].

In the context of text categorization, it is common and easy to apply feature selection process under the assumption that supervision exists for that process [61]. On the other hand, a number of simple unsupervised methods can also be used in text categorization for feature selection process. This section reviews some examples of such methods.

3.2.1 Document Frequency

One of the simplest method that can be used in feature selection process is the use of document frequency (DF) to filter out irrelevant features. Simply put, the idea is based on the assumption that very frequent words are often not discriminative. Therefore, words that are too frequent in the corpus can be removed because they are often common words. Examples of words that are not discriminative include "a", "an" and "the". These words are also often known as *stop words*.

There are many methods available in the literature for stop-word removal [33]. Lists of about 350 to 400 stop words for different languages are available to be used in the retrieval process [33]. In some cases, it happens that words may be mistyped in document; therefore, words that occur extremely infrequently can be removed as well. Such words can more likely occur in Web pages like blogs or social networks [33]. The reason why these words can be removed is that they do not add anything to similarity calculations which are used in most clustering algorithms [33].

There is a weighting method, called **T**erm **F**requency - **I**nverse **D**ocument **F**requency (TF-IDF), that can also be used to filter out the very common words in a soft way [62]. The TF-IDF is the product of two measures, term frequency (TF) and inverse document frequency (IDF). The TF, in the simplest form, represents the raw count of a term in a document. On the other hand, the IDF represents whether the term is common across all documents in the corpus [33]. There are many various ways to determine the exact values of both measures. Log normalization is one of the popular weighting schemes of TF, as shown in Formula 3.1 [33]. Also, one of the popular forms for IDF is shown in Formula 3.2 [33].

$$tf(t,d) = \log\left(1 + f_{t,d}\right) \tag{3.1}$$

$$idf(t,D) = \log\frac{N}{n_t} \tag{3.2}$$

where t is the term that occurs in document d, $f_{t,d}$ is the raw count of the term t in the document d, N is the total number of documents in the corpus D, and n_t is the number of documents where the term t occurs.

3.2.2 Term Strength

Term strength method was originally proposed as a method that can be used in stop word reduction process in information retrieval by Wilbur and Sirotkin [63]. However, term strength was later applied in text categorization by Yang [64]. The idea is simply the more the term is strong the more the term is relatively informative and shared by related documents. One drawback of term strength: it may have high computation complexity due to the case of high number of documents, leading to difficulty with parameter tuning [27]. This method extends techniques used in supervised learning to the unsupervised learning [33].

The term strength is basically used to measure how good a word is for identifying two related documents. If there exist two related documents x and y, the term strength $s(t)$ of the term t can be defined using the following probability [33]:

$$s(t) = P(t \in y | t \in x) \tag{3.3}$$

One main issue here is that how one can define the documents x and y as related. One solution to define when a pair of documents are related is to use manual user feedback, which is rather equivalent to utilizing supervision in the feature selection process [33]. However, this is not practical when manually creating related pairs in large collections [33]. As result, there should be an automated and unsupervised way to define the notion of when two documents are related. To define the relatedness of two documents automatically, automated similarity functions can be used such as cosine function [33]. Therefore, two documents are defined to be related if their cosine similarity is above a predefined threshold. In this case, the term strength s(t) can be defined by sampling a number of pairs randomly as in Formula 3.4 [33].

$$s(t) = \frac{\text{Number of pairs in which t occurs in both documents}}{\text{Number of pairs in which t occurs in the first document}}$$

(3.4)

where the first document can simply be picked randomly.

Now to prune features using the term strength method, the term strength $s(t)$ can be compared to expected term strength that is randomly distributed in the training documents [33]. The term t is removed from the collection if $s(t)$ is more than two standard deviations greater than that of the random word [33].

3.2.3 Term Contribution

The idea behind term contribution is based on the fact that results of document clustering are highly dependent on document similarity. The contribution of a term can be seen as its contribution to document similarity. As an example, the contribution of a term to the similarity of two documents is the dot product of normalized frequencies in these two documents [62]. A major drawback of this method is that it tends to favor highly frequent words without looking at the discriminative power in the clustering process. Moreover, this method consumes $\mathcal{O}(n^2)$ time for each term. Therefore, it may require sampling methods to speed up the contribution [62].

In most of these methods, there is a concern that the term selection might be biased to the similarity function (e.g., cosine) used in the process. This means that if a different similarity function is used, the method may end up having different results for term selection [62]. Therefore, the decision of selecting an appropriate similarity function is very important for these methods [62].

3.3 Search Result Clustering and the Arabic Language

Most of the previous related work is done for the English language. Consequently, many algorithms that are developed for the English language perform poorly when applied to other languages like Arabic, which is highly inflectional and morphologically rich [65, 66]. Some proposed methods and approaches are language-specific like Russian [67], Chinese [68], and Turkish [69]. However, a little published research has been found for the Arabic language like in [23, 24, 25, 26].

3.3.1 Arabic Search Results Clustering

Sahmoudi and Lachkar [23] claimed that the suffix tree clustering algorithm (STC) has never applied for Arabic Web search snippets. Thus, they studied how STC algorithm can be applied to Arabic snippets. Even though they gained promising results, the evaluation they performed is subjective and not objective. The reason behind using subjective evaluation is the lack of standard labeled test collection for the Arabic language [23]. To the best of our knowledge, there is no standard test collection of the Arabic language for evaluating search results clustering. Indeed, this is the main motivation behind building labeled data collection of search results for the Arabic language in this thesis. In another work [24], Sahmoudi and Lachkar proposed an interactive system for Arabic Web search results clustering (ISAWSRC) for Arabic query reformulation. This system enables users, after the systems shows the cluster labels, to click on produced cluster label so that the system can then retrieve more relevant results. A very recent work [70] done by the same authors, studies how to integrate and adapt formal concept analysis (FCA) for Arabic web search results clustering. They performed an experimental study to show that FCA is better than suffix tree clustering and Lingo [10] in terms of both clustering and label quality. They used a dataset of Arabic documents from the Open Directory Project as a benchmark.

Most of the related work in this and previous sections are based on experimental research, which mainly tries to evaluate clustering experiments and give some recommendations for different stages of clustering, especially in the preprocessing stage. This is discussed further next.

3.3.2 Text Feature Selection and Extraction

Feature selection methods have been widely used in many real applications. This includes pattern recognition applications like Object-Based Land Cover Mapping of Unmanned Aerial Vehicle Imagery [71], text categorization, image processing, and many others [27].

One can observe that the main challenge in terms of clustering quality in SRC approach, is how to represent and extract features from search results. Each work related to Web search clustering suggests a particular way of extracting features. For example, the authors of Grouper system [9] suggested to use single words and ordered sequences of words as text features. Lassi authors [22], on the other hand, used lexical affinities, i.e., pairs of words with strong correlation of appearance in the input text (such pairs are said to share a lexical affinity). Lingo authors [10] used flat clustering with frequent phrases (i.e., 2-gram words) as text features. The authors of SRC [72] (a previous add-on to MSN search) suggested to use n-gram of words as text features in their clustering framework.

A recent master thesis [73] evaluated the effect of preprocessing in Arabic documents clustering in the general context, not in the search results context. Its goal was to find the best combinations of these techniques when using clustering algorithm. They used two clustering algorithms K-means and expectation maximization (EM). Their results confirmed that: *(1)* K-means is suitable for Arabic text clustering and gives better evaluation than EM algorithm. *(2)* Euclidean distance is more appropriate than Manhattan distance for Arabic text clustering. *(3)* Applying term pruning with small value for TF enhances the evaluation (minimum TF of 3 gave the best value of evaluation). *(4)* Term weighting (TF-IDF) enhances the evaluation. *(5)* Regarding morphological analysis, light stemming is found to be more appropriate than root-based stemming and raw text. *(6)* Using normalization enhances the evaluation too. These results are almost consistent with other comparative studies [74, 75].

Other studies have been performed to investigate the effectiveness (i.e., clustering quality) of different stemming approaches on Arabic text clustering quality. For example, Ghanem and Ashour [76] showed that light stemming achieved best results of clustering quality in terms of recall, precision and F-measure when compared with others (i.e., root-based and without stemming). Alomari [77], on the other hand, achieved the best results of clustering quality without stemming. From such studies, one can conclude that stemming more likely decreases the effectiveness.

With all these studies attempting to address the clustering of Arabic content on Web, a recent review study done by Alghamdi and Selamat [27] confirms the challenges that face Arabic language. In particular, this includes:

1. How to identify significant features?
2. How to build a suitable model that results in high-performance clustering model?

This thesis helps fill the first gap by thoroughly studying *feature sources* and *vector space representations*, which gives insights of the best combinations that result in a high-quality clustering. Moreover, this work mitigates the second challenge by proposing a novel approach to build a supervised clustering model that competes with the popular traditional approach that is purely dependent on the unsupervised learning. This supervised clustering model is based primarily on leveraging search results of clear queries. Additionally, in the light of the lack of published benchmarks of search results clustering for the Arabic, this work builds a complete benchmark that contains different datasets designated for the Arabic.

Chapter 4

Data Collection: AMBIGArabic

One of the important aspects of any experimental research is understanding and collecting the data required for experiments. In order to perform and evaluate experiments of search results clustering, real data from web search engines must be collected. Typically, any data for search results clustering include:

1. A set of ambiguous queries.
2. A set of meanings for each ambiguous query.
3. A set of search results for each ambiguous query.
4. A corresponding meaning for each search result (i.e., the labels).

Unfortunately, when we looked for SRC benchmarks, there was no benchmark for Arabic published and publicly available, intended for experimenting with search results clustering. In contrast, English has a number of benchmarks intended for that purpose. The most popular and widely used benchmarks are AMBIguous ENTries (AMBIENT)[1] [78] and MORE Sense-tagged QUEries (MORESQUE)[2] [79]. Both benchmarks were collected and investigated separately. A statistical analysis was performed for the two benchmarks to show how the results are distributed across the meanings of all queries. Figure 4.1 shows how search results are distributed over the meanings of the two ambiguous queries, "Aida" query from AMBIENT and "Stephen king" from MORESQUE[3]. It was observed that the queries along with their meanings in both benchmarks are selected from Wikipedia disambiguation pages. As shown in Figure 4.1, there are also some queries where their search results do not cover all meanings, thus leaving a lot of search results unlabeled.

The main challenges of building benchmarks of SRC are:

[1] Description of AMBIENT dataset and the download link can be found at `http://search.fub.it/ambient/`

[2] Description of MORESQUE dataset and the download link can be found at `http://lcl.uniroma1.it/moresque`

[3] The remaining queries can be found here: `https://goo.gl/mu2FJM` and `https://goo.gl/BYy3je` for AMBIENT and MORESQUE, respectively

1. The feasibility and the limits of fetching search results from web search engines.
2. Specifying the meanings for each ambiguous query, given that the nature of meanings or senses is dynamic and varies over time.
3. Labeling search results with the collected meanings.

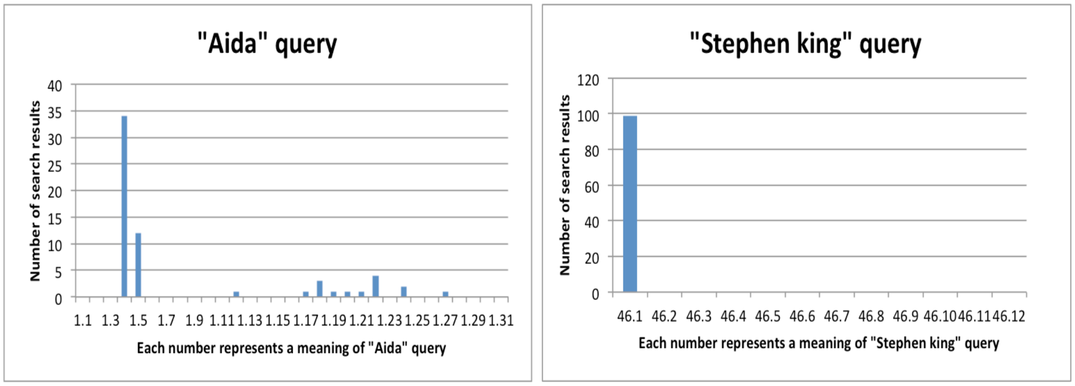


Figure 4.1: Distribution of search results over meanings.

This chapter thoroughly presents the work of building datasets intended for performing search result disambiguation experiments for Arabic language. The feasibility, data design, and search results acquisition are discussed in section 4.1. In this study, meanings were collected with the help of Wikipedia disambiguation pages. Section 4.2 proposes two novel approaches for labeling search results instead of the manual labeling approach. Finally, three core benchmarks are proposed for Arabic: two benchmarks with a gold standard and one benchmark based on blind relevance feedback. The proposed benchmarks for Arabic language (i.e., AMBIGArabic) are described in Section 4.3.

4.1 Toward Building an SRC Benchmark for Arabic

4.1.1 Feasibility Study: Fetching Search Results

One of the early steps in this study was to check the feasibility and limits of fetching search results from the popular search engines like Google, Bing, and Yahoo. Even though the experiments of this study were performed on two engines, Google and Bing, the three engines were investigated in the feasibility study stage. In fact, investigating the additional engine (i.e., Yahoo!) was important because any search engine can block us from fetching search results. Therefore, Yahoo engine was considered as a backup in this study.

4.1.1.1 Google

Google provides a service called Google Custom Search Engine (CSE) [80]. The purpose of this service is that it lets you include a search engine on your web site to help visitors find the information they are looking for. Moreover, it has the option to set it to search the entire web. We had three options to search Google:

1. Using the official Google API (e.g., Java client library) with custom search domain.
2. Exploiting the URL of custom search engine, requesting a query through a plain HTTP GET method, and then parsing the returned results (HTML scrapping).
3. Exploiting the URL of public Google search engine (HTML scrapping).

The first option has some limits like 1000-10,000 requests/day with a free quota of 100 requests only. Additional requests cost $5 per 1000 requests, up to 10,000 requests per day. The second option seemed to be fine. When trying the third option, Google blocked us as they are able to detect any robot generating unusual traffic from the same network. In the first two options, we can fetch 100 search results per query only though. Therefore, the final decision was to go with the option two i.e., fetching results from Google custom search URL. This required us to create our own domain[4].

4.1.1.2 Bing

To interact with Bing search engine, Bing was offering a service called Bing Search API [81] with a free quota of 5000 requests/month. Unfortunately, as of December 15, 2016 Microsoft announced that this API will no longer be supported, and they will offer a new cognitive service through their cloud platform[5] [81].

As another choice, we were able to exploit the public URL of Bing search engine (HTML scrapping). After several trials of fetching from their public URL, it seemed to be fine. We were able to fetch up to 200 results per query.

4.1.1.3 Yahoo!

Over that past years, Yahoo was providing a service called BOSS Search API. After March 31, 2016, they stopped BOSS search API and have provided a service similar to Google CSE, called Yahoo Partner Ads. Its main purpose to make a custom search engine for your site. As with Bing, we were able to exploit Yahoo search URL by building HTML scrapper. It worked fine, and no issues reported.

4.1.2 Data Design

In this section we show what the nature of data is and how data is modeled. The base model designed is a relational model. We have seven core relations. Four core entities

[4] `https://cse.google.com/cse/publicurl?cx=011305709239177939329:h3wb8k8xtky`

[5] `http://www.azure.com/`

are related to search results: *query*, *meaning*, *search result*, and *search engine*, and two core entities are related to user labeling: *user* and *user labeling*. In addition, we have the *full document* (we call it inner page) as a property of search result.

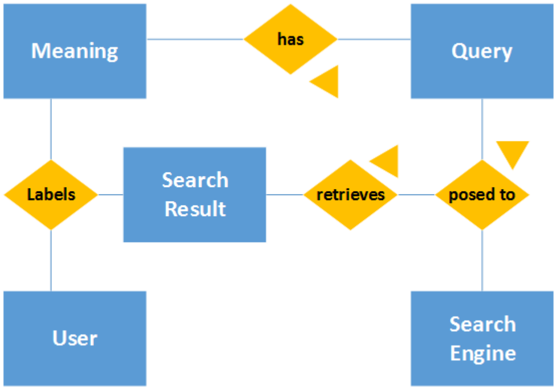


Figure 4.2: A high level entity relationship diagram of data.

Figure. 4.2 abstracts the core entities combined with the main relationships. The *query* table is used to store queries. Since the query could be either ambiguous or clear, the *query* table has a flag *is_ambiguous* to indicate whether the query is ambiguous. The ambiguous query can have multiple meanings. The *meaning* table is used to store the meanings. When posing a query to a search engine, search results are retrieved and stored in the *search result* table. The *user* table is used to store users who can label search results with meanings.

Some of the other entities and relationships (i.e., link tables) are not shown here. For more detailed scheme, please refer to Figure A.1 in Appendix A.

4.1.3 Search Results Acquisition: The Fetcher System

After studying the feasibility of fetching search results from Google, Bing, and Yahoo, the next step was implementing the fetcher system[6]. Based on a given query, this system is responsible for delivering search results, each of which contains a title, a snippet (a short summary), a URL pointing to the full document, and the full document itself[7].

The fetcher system was implemented, and all the implementation details as well as the source code are available at Spread repository[8]. Moreover, user interfaces were implemented to be able to test the fetcher system.

The high-level components of the data acquisition system (i.e., the fetcher system) are represented in Figure 4.3. The collected queries are loaded by the *data loader* component. The *crawler* component delivers the loaded queries into the *fetcher* components and then passes the search results to *persistence* component for storing the results. The

[6] Some call it data acquisition system.

[7] Not all sites allow you to fetch their inner pages. That's why there are some fetched search results without inner pages

[8] `https://github.com/haytham-salhi/Spread`

controller component gets the loaded queries from the *data loader* component and passes them to the *crawler* component to start fetching the search results.

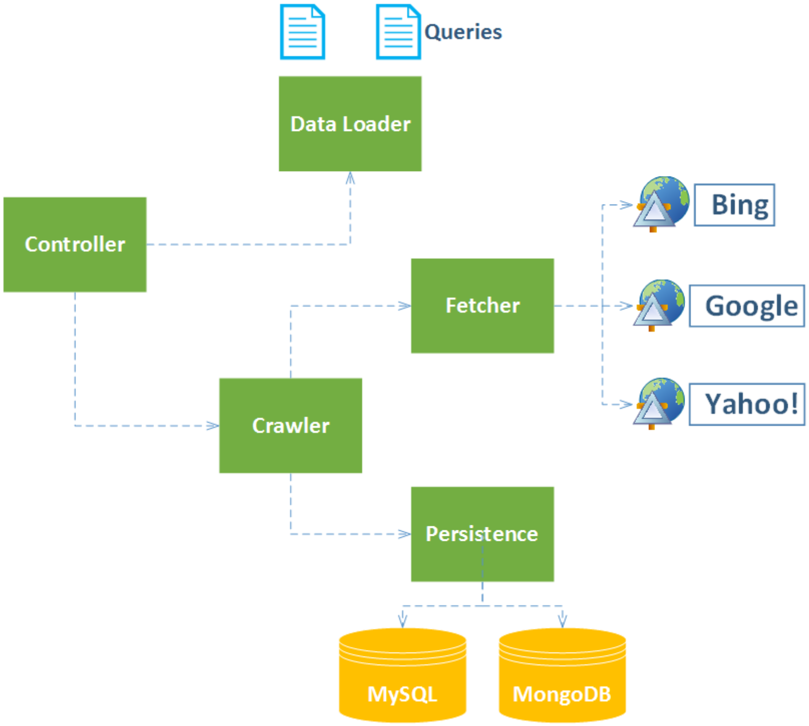


Figure 4.3: A high level component diagram for data acquisition system.

4.1.4 Characteristics of Search Results

After collecting an initial set of queries, we then looked into search results returned by the search engines, Google and Bing, and the following characteristics/issues were found:

- A query can refer to instances of the same entity and/or instances of different entities. As an example, the query آرمسترونج can refer to different entities of person class and even different entities of location class.

- A search engine can return results for specific subset of meanings or senses (not all senses). For example, if we query python on Google, most likely it would return the first 100 results about python as a programming language. So the assumption that might come to mind at first that a search engine would always return results (Say top 100) covering all meanings, is not correct.

- When searching for something in a web search engine, the search engine might return documents that have the whole query words or a portion. As an example, the query الجامعة العربية might return results that contain العربية only.

- We might think that if some query in Wikipedia disambiguation pages is disambiguated for persons only, then the search engine would return results for those persons only. That is not correct, a search engine might return results for other entities as well. The query سعود بن عبد العزيز, for example, can refer to persons or universities, or something we do not know about currently. In other words, new meanings or senses can show up over time. This is the reason at some point why you should specify the meanings explicitly for evaluation purposes.

4.2 Search Results Labeling

One important aspect of the benchmark is the data labeling. In fact, it represents one of the core challenges in this study where no Arabic datasets available for search results clustering. What researchers usually do is assessing the results by the help of humans. More often, however, the idea of hiring assessors is expensive and time consuming. This section defines the proposed labeling approaches and shows how the manual labeling was performed in this study.

4.2.1 The Proposed Approaches

This subsection presents two novel approaches for data labeling: *intersecting* and *mixing* approaches. Exploiting the search engine judgment and querying the meanings (i.e., the formulated clear queries) are the main motivation behind these two approaches.

4.2.1.1 Intersecting Approach

Formal Definition To disambiguate an ambiguous query q, which has different meanings/senses $S = \{s_1, s_2, ..., s_n\}$, using a search engine SE:

1. Fetch the results R_q for q, so we have $R_q = \{r1, r2, ..., r_m\}$ where m is the maximum number of search results that could be fetched, and r_i is the ith search result.

2. Fetch the results for each clear query (that is formed by combining q and s_i)[9] where $1 \leq i \leq n$, so we have: $R_{q(s1)} = \{r_{s11}, r_{s12}, ..., r_{s1j}\}$ ... until we reach $R_{q(sn)} = \{r_{sn1}, r_{sn2}, ..., r_{snk}\}$, where $0 < j, k \leq m$, and $R_{q(si)}$ is the list of search results of the clear query formed by q and s_i.

3. Annotate the search items as follows: $\forall$ item $\in (R_q \cap Rq_{(s_i)})$, annotate it with s_i.

This approach was looked into by developing an interface for testing it for different search engines. A snapshot of this interface is available here[10]. Moreover, some statistics and charts were generated to show some insights of intersections between search results of

[9] For example, say we have $q =$ أمازون and $s1 =$ شركة, then the resulting clear query is شركة أمازون.

[10] `https://goo.gl/DTEszH`

meaning and search results for an ambiguous query. Table 4.1 shows these statistics for some queries. The full sheet can be found here[11].

One drawback of this approach is that not all search results will be labeled. Take the query المالكي from Table 4.1 as an example. In Google, only 22 (i.e., 2 + 15 + 2 + 3) search results out of 100 were labeled. In Bing, only 55 (i.e., 24 + 24 + 1 +6) search results out of 200 were labeled.

Table 4.1: The intersections between results of ambiguous query and results clear queries.

Query	**Meaning**	**Formulated Query**	**Google Retrieved Results**	**Google Intersections**	**Bing Retrieved Results**	**Bing Intersections**
المالكي		المالكي	100		200	
	المذهب	المذهب المالكي	100	2	200	24
	نوري المالكي	نوري المالكي	100	15	200	24
	مراد المالكي	مراد المالكي	100	2	200	1
	فايز المالكي	فايز المالكي	100	3	200	6
عمان		عمان	100		200	
	سلطنة	سلطنة عمان	100	33	200	51
	مدينة	مدينة عمان	100	6	200	26
أمازون		أمازون	100		200	
	نهر	نهر أمازون	100	4	200	5
	شركة	شركة أمازون	100	23	200	49
البقرة		البقرة	100		200	
	حيوان	حيوان البقرة	100	4	200	6
	سورة	سورة البقرة	100	74	200	93

4.2.1.2 Mixing Approach

The idea of mixing approach is slightly different. It is based on collecting the search results for the meanings (i.e., formulated clear queries) first.

Formal Definition For an ambiguous query q, which has different meanings/senses $S = \{s_1, s_2, ..., s_n\}$, and using a search engine SE:

1. Fetch the results for each clear query (that is formed by combining of q and s_i) where $1 \leq i \leq n$.

2. Label the top N results[12] for each $R_{q(si)} = \{r_{si1}, r_{si2}, ..., r_{sij}\}$ with s_i, where $1 \leq i \leq n$, and $R_{q(si)}$ is the list of search results of the clear query formed by q and s_i.

3. Mix the labeled results from above together with the query q.

[11] https://goo.gl/8VW8eM

[12] The value of N depends on the desired size of the dataset you want to make.

4.2.1.3 Discussion of Approaches

Starting with the *intersecting approach*, even though a search engine tries its best to retrieve relevant results as a response to a clear query, some (usually few) items can be non-relevant.

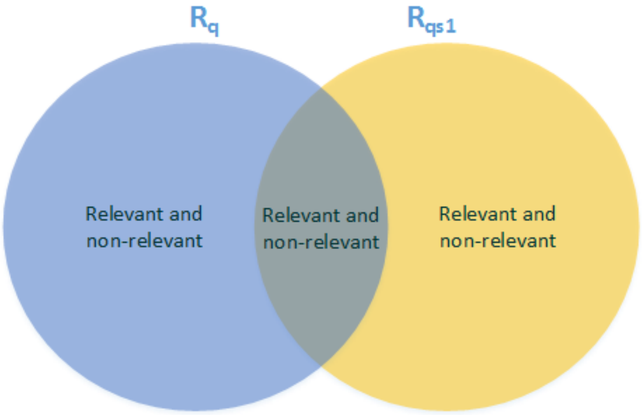

R_q
R_{qs1}
Relevant and non-relevant
Relevant and non-relevant
Relevant and non-relevant

Figure 4.4: A Venn diagram showing the two sets: R_q, R_{qs1}.

Looking at Figure 4.4, let us try to cover all possibilities which might happen when intersecting R_q with $R_{q(s1)}$:

- R_q (i.e., the set of search results of ambiguous query) can contain relevant and non-relevant results to s_1.
- $R_{q(s1)}$ (i.e., the set of search results of of the meaning s_1) most likely contains relevant results to s_1. However, it might contain non-relevant results to s_1 (usually *few* if any).
- When intersecting the two sets R_q and $R_{q(s1)}$, the output of intersection could be:
  - Relevant to s1 (as a result of relevant and relevant), which is fine.
  - Non-relevant to s1 would incorrectly be annotated as s_1 (as a result of non-relevant and non-relevant). They should be very few though. This is an **issue**! (One could try to add additional conditions along with the intersection to avoid this case as much as possible).
  - The cases (relevant and non-relevant) and (non-relevant and relevant) are impossible of course and would not happen (because we are taking the intersection).
  - Relevant items to s1 exist in R_q and do not exist in $R_{q(s1)}$ would not be labeled as s1. This is an **issue**! (They should be few; the items that are not annotated must not affect the evaluation!)

Although the *intersecting approach* is interesting for different purposes, it cannot be adopted alone in the labeling process. This is because not every search result will be labeled as justified above in 4.2.1.1.

In the *mixing approach*, on the other hand, we get rid of the drawback of the intersecting approach by ensuring that each query has all search results labeled. Furthermore, the results are now more representative and we ensure that the search results of a query cover all meanings.

As a first step, we started collecting ambiguous queries. Then the next job was to look into data and start building the benchmark.

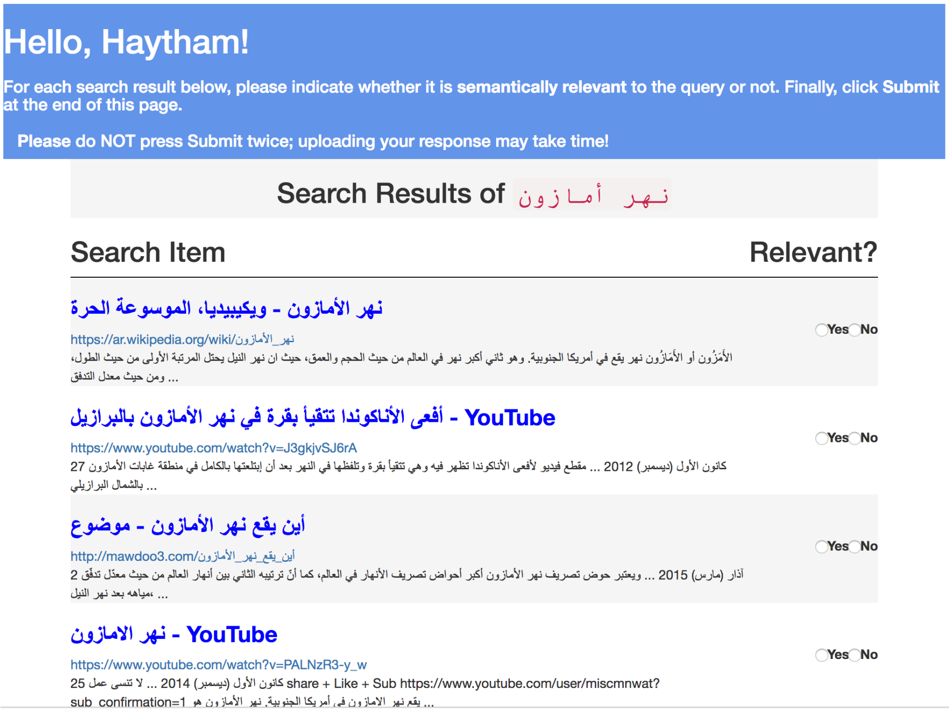
4.2.2 Human Relevance Assessment

As mentioned earlier, the assumption that all search results for some query are always relevant is not true. That is why using mixing approach alone is not sufficient for building a ground truth. In order to build benchmarks with a gold standard judgment of relevance, a human relevance assessment interface was developed for enabling a group of users to label search results manually, thus helping the process of building the gold standard or ground truth.

This interface supports two types of manual labeling:

- Yes/no annotation.
- Choices-based annotation.

In *yes/no annotation* strategy, an assessor indicates, for a specific information need expressed by a query, whether a search result is relevant to the information need. If the search result is judged relevant, the user selects yes; otherwise, no would be selected. Figure 4.5 shows the yes/no annotation interface for a specific query, along with its search items and choices.



Hello, Haytham!

For each search result below, please indicate whether it is **semantically relevant** to the query or not. Finally, click **Submit** at the end of this page.

Please do NOT press Submit twice; uploading your response may take time!

Search Results of نـهـر أمـازون

Search Item — Relevant?

نهر الأمازون - ويكيبيديا، الموسوعة الحرة ○Yes○No

https://ar.wikipedia.org/wiki/نهر_الأمازون

الأَمَزُون أو الأَمَازُون نهر يقع في أمريكا الجنوبية. وهو ثاني أكبر نهر في العالم من حيث الحجم والعمق، حيث ان نهر النيل يحتل المرتبة الأولى من حيث الطول،
ومن حيث معدل التدفق ...

أفعى الأناكوندا تتقيأ بقرة في نهر الأمازون بالبرازيل - YouTube ○Yes○No

https://www.youtube.com/watch?v=J3gkjvSJ6rA

كانون الأول (ديسمبر) 2012 ... مقطع فيديو لأفعى الأناكوندا تظهر فيه وهي تتقيأ بقرة وتلفظها في النهر بعد أن إبتلعتها بالكامل في منطقة غابات الأمازون 27
بالشمال البرازيلي ...

أين يقع نهر الأمازون - موضوع ○Yes○No

http://mawdoo3.com/أين_يقع_نهر_الأمازون

آذار (مارس) 2015 ... ويعتبر حوض تصريف نهر الأمازون أكبر أحواض تصريف الأنهار في العالم، كما أنّ ترتيبه الثاني بين أنهار العالم من حيث معدّل تدفّق 2
مياهه، بعد نهر النيل ...

نهر الامازون - YouTube ○Yes○No

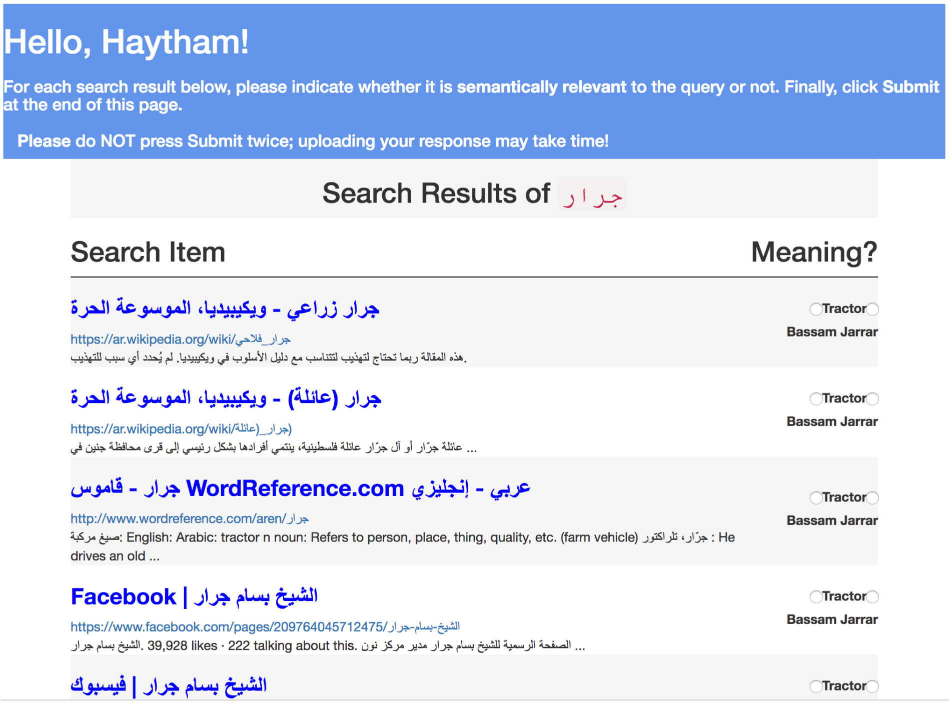
https://www.youtube.com/watch?v=PALNzR3-y_w

25 لا تنسى عمل ... 2014 (ديسمبر) كانون الأول share + Like + Sub https://www.youtube.com/user/miscmnwat?
sub_confirmation=1 يقع نهر الامازون في امريكا الجنوبية. نهر الأمازون هو ...

Figure 4.5: An example snapshot of yes/no interface.

Yes/no annotation strategy was used to build a gold standard of the mixing-based benchmark. In particular, this type of benchmark is built primarily using clear queries. The mixing-based benchmark is discussed further in Section 4.3.

The second type of the labeling used in this manual process is *choices-based annotation*. This strategy was used to label search results of a query with a predefined set of meanings. In other words, the user selects, for each search result, one meaning only from the predefined set of meanings, indicating the sense of that search result. If the search result has sense other than the predefined ones, no selection will be made. Figure 4.6 shows the choices-based annotation interface for a specific query, along with its search items and its choices that represent its meanings.



Hello, Haytham!

For each search result below, please indicate whether it is **semantically relevant** to the query or not. Finally, click **Submit** at the end of this page.

Please do NOT press Submit twice; uploading your response may take time!

Search Results of جرار

Search Item

Meaning?

جرار زراعي - ويكيبيديا، الموسوعة الحرة
https://ar.wikipedia.org/wiki/جرار_فلاحي
هذه المقالة ربما تحتاج لتهذيب لتتناسب مع دليل الأسلوب في ويكيبيديا. لم يُحدد أي سبب للتهذيب.
○Tractor○
Bassam Jarrar

جرار (عائلة) - ويكيبيديا، الموسوعة الحرة
https://ar.wikipedia.org/wiki/جرار_(عائلة)
عائلة جرّار أو آل جرّار عائلة فلسطينية، ينتمي أفرادها بشكل رئيسي إلى قرى محافظة جنين في ...
○Tractor○
Bassam Jarrar

عربي - إنجليزي WordReference.com جرار - قاموس
http://www.wordreference.com/aren/جرار
صيغ مركبة: English: Arabic: tractor n noun: Refers to person, place, thing, quality, etc. (farm vehicle) جرّار، تلراكتور : He drives an old ...
○Tractor○
Bassam Jarrar

الشيخ بسام جرار | Facebook
https://www.facebook.com/pages/209764045712475/الشيخ-بسام-جرار
الشيخ بسام جرار. 39,928 likes · 222 talking about this. الصفحة الرسمية للشيخ بسام جرار مدير مركز نون ...
○Tractor○
Bassam Jarrar

الشيخ بسام جرار | فيسبوك
○Tractor○

Figure 4.6: An example snapshot of choices-based interface.

Choices-based strategy was used to build a gold standard of the plain human-annotated (HA) benchmark, which basically contains real search results of ambiguous queries. The plain benchmark is discussed further in Section 4.3.

4.3 AMBIGArabic Benchmarks

This section presents the three benchmarks that were built during the thesis work. They are called AMBIGArabic Benchmarks (which is a shortcut for Ambiguous Arabic Benchmarks). Two of these benchmarks (i.e., mixing-based and plain benchmarks) are made with a gold standard, and one benchmark is based on the blind relevance feedback.

4.3.1 Mixing-based Human-annotated Benchmarks

As the name suggests, this benchmark is based on the mixing labeling approach discussed in Section 4.2.1.2. Leveraging the search results of clear queries to build datasets is very useful for experiments.

Since it is important for some types of experiments and metrics to have balanced datasets, the desired goal of this benchmark is to have datasets containing a set of queries, each with *balanced* search results of more than one meaning. That is, if a query has two meanings, then the search results of the first meaning and the search results of the second meaning will be combined together. In other words, all meanings of a query will be equally represented.

To build this benchmark, the following steps were followed:

1. 30 ambiguous queries were collected with their meanings. From the meanings, 63 clear queries were formed. The full list of queries and meanings are shown in Table C.1 in Appendix C and can be found online here[13].

2. After forming the clear queries, search results were fetched for both engines, Google and Bing. In particular, 100 search results were fetched for Google, and 200 search results were fetched for Bing.

3. After that, yes/no annotation labeling was performed by humans on the clear queries in order to produce datasets (i.e., queries) with equal sizes of relevant search results belonging to different meanings. The final judged results can be found one here for Google[14] and Bing[15].

The mixing-based benchmark consists of the following:

1. 30 ambiguous queries along with their meanings (between two and three for each query).

2. The top 30 human-judged relevant search results for each clear query (formed using the meaning), along with their titles, snippets, and inner pages. That means, if an ambiguous query has two meanings, it would be 60 (30 + 30) search results. The lowest common value among the queries is 30; that's why that number is selected.

3. The human labels for all search results.

Some statistics about human judgment giving information like number of judges were calculated. This is shown in Table C.3 in Appendix C.2 for Google as an example. They are also available online for Google[16] and Bing[17].

4.3.2 Mixing-based BRF-annotated Benchmarks

Blind Relevance Feedback (BRF), also referred to as pseudo relevance feedback, is one of the types of relevance feedback used in information retrieval systems. The idea behind blind relevance feedback is to assume that the top k ranked results are relevant to the query, without user intervention [82]. In this study, this was seen as an opportunity to check how useful the blind relevance feedback could be in search results clustering, when using it for labeling search results.

To build this benchmark, the same procedure as above benchmark was followed except in the third step. In the third step, the top 50 search results were assumed to

[13] https://goo.gl/UcSkkE

[14] https://goo.gl/KRBvsB

[15] https://goo.gl/epg2Ct

[16] https://goo.gl/jjcR2J

[17] https://goo.gl/v4xadq

be relevant to their query, producing datasets with equal sizes of relevant search results belonging to different meanings.

The BRF-based benchmark basically contains the same elements as above but with different size of search results because the top 50 ranked search results (assumed as relevant) were collected for each clear query. That means, if an ambiguous query has two meanings, it would have 100 (50 + 50) search results.

4.3.3 Plain Human-annotated Benchmarks

In fact, a search results disambiguation system would work on real search results of an ambiguous query. Out of this need, benchmarks of search results of ambiguous queries were built.

To build this benchmark, the following procedure were followed:

1. A subset composed of 11 and 15 queries[18] were chosen for Google and Bing, respectively, on the basis that they have reasonable search results belonging to more than one meaning of predefined meanings. Each subset, for Google and Bing, has three queries with three meaning. The remaining queries has two meanings for each.

2. For each query, 100 search results for Google and 200 search results for Bing were fetched. Regarding inner pages, not all sites allow you to crawl their web pages. However, inner pages for most search results were successfully fetched. The fetcher component was capable of fetching inner page for most search results. For Bing, the fetcher was able to crawl 2822 inner pages out of 3000 search results. For Google, the fetcher was able to crawl 1033 out of 1100.

3. After that, all search results were labeled manually using the developed assessment interface. The annotation process is choices-based selection. As an example, *sakhr* has three senses: company, sakhr bn amro, or stone, as shown below.

The plain benchmark now consists of the following:

1. 11 ambiguous query for Google and 15 queries for Bing along with their predefined meanings (between two and three for each query).

2. The top 100 ranked search results for Google and the top 200 ranked search results for Bing.

3. The human labels for search results.

The final judged results can be found here for Google[19] and Bing[20]. Some statistics about human judgment are shown in Table C.4 for Google and in Table C.5 for Bing.

[18] They can be found here: `https://goo.gl/SrBBWf`

[19] `https://goo.gl/VFebSk`

[20] `https://goo.gl/dZm3jk`

They can be found online for Google[21] and Bing[22] as well.

Table 4.2 and Table 4.3 summarize the three benchmarks in terms of size of queries, size of search results, and labeling method used. All the benchmarks above can be downloaded for use from this online location[23].

Table 4.2: Size of queries and labeling method for each benchmark.

Benchmark	**Google queries**	**Bing queries**	**Google clear queries**	**Bing clear queries**	**Labeling method**
Human-annotated mixing-based	30	30	63	63	Mixing-based approach with human
BRF mixing-based	30	30	63	63	Mixing-based approach with relevance feedback
Plain	11	15	25	33	Human

Table 4.3: Size of search results per query for each benchmark.

Benchmark	**Search results per query/Google**		**Search results per query/Bing**	
	Queries with 2 meaning	**Queries with 3 meanings**	**Queries with 2 meanings**	**Queries with 3 meanings**
Human-annotated mixing-based	60	90	60	90
BRF mixing-based	100	150	100	150
Plain	100		200	

[21] `https://goo.gl/dsp4zh`

[22] `https://goo.gl/658ftd`

[23] `https://github.com/haytham-salhi/Spread/tree/master/datasets`

Chapter 5

Experimental Design and Methodology

This chapter presents the complete design and mechanism of the experiments that were performed. The first section shows the experiments that investigate the influence of feature sources and feature representations on the effectiveness of search results clustering. It starts with the research hypothesis. Then it shows the independent variables and their treatments as well as the other neutralized variables. Finally it discusses how these experiments are executed and evaluated.

The second section presents the design of how an supervised approach could be utilized in unsupervised learning and how such approach could be executed and evaluated.

5.1 Influence of Features

The first type of experiments is concerned with the study of the influence of feature source and feature extraction on the effectiveness of search result clustering. Particularly, it explores whether there is a significant difference between feature sources (i.e., title, snippet, title with snippet, and inner page) and feature space representations (i.e., single words, phrases, single words with 2-grams, single words with 2-grams and 3-grams) on the effectiveness by performing several experiments on our benchmarks such as human-annotated, BRF-annotated, and plain benchmarks. Moreover, it investigates how useful blind relevance feedback could be in search results clustering. The main goal of using the blind relevance feedback is to see whether it supports results in other experiments (i.e., human annotated).

5.1.1 Research Hypotheses

This study has two main hypotheses. The goal is to support the alternative hypothesis by rejecting the null hypothesis.

- **Null hypothesis:** There is no influence of different feature sources and feature space representations on the effectiveness of Arabic search results clustering measured by the F-measure of clustering.

- **Alternative Hypothesis:** There is a significant influence of different feature sources and feature space representations on the effectiveness of Arabic search results clustering measured by the F-measure of clustering.

5.1.2 Dependent Variable

As we focus on one of the important requirements of search results clustering, *effectiveness* is considered as the dependent variable of this study and is measured by the *F-measure* of search results clustering.

5.1.3 Independent Variables of Interest

The main independent variables that were investigated through this study are related to text feature engineering:

- **Feature Source**: This variable is very important as it represents the source where the features are extracted from. This variable has the following conditions:
  1. Title only.
  2. Snippet only.
  3. Title with snippet.
  4. Inner page (Full document text).
- **Feature Space Representation**: This is another important variable that represents how the text features are represented in a vector space model. This variable has the following conditions:
  1. Single words.
  2. Phrases of 2-grams.
  3. Single words with phrases of 2-grams.
  4. Single words with phrases of 2-grams and 3-grams.

5.1.4 Neutralized Variables

There are many other variables that might affect the performance of search results clustering but which are not factors of interest in this study. To make our experiments valid, the other variables were neutralized as much as possible. Most other variables can be categorized into three categories: data (feature) preprocessing, feature representation, and clustering algorithm. Furthermore, we have a variable related to the raw data, which is the size of search results per query.

5.1.4.1 Data Preprocessing Variables

The variables related to data preprocessing with their values as well as the rationales are discussed below.

1. - *Variable*: Stemming.
   - *Value*: Light Stemming.
   - *Rationale*: Generally, stemming is a good practice for the variations of the same word. When performing stemming, more than one word with different meanings can be mapped to one stemmed word. Thereby, while stemming can improve recall in some cases, it can hurt precision slightly. Moreover, studies [76, 73] for Arabic language have shown that the light stemming is better than the root-based stemming or null stemming (i.e., without stemming) in text clustering.

2. - *Variable*: Ambiguous query words.
   - *Value*: To be removed.
   - *Rationale*: In dry runs of some experiments, we noticed that the ambiguous query terms can considerably affect the quality of search results clustering. The reason behind that is that the similarity of documents largely depends on word weights. Therefore, an ambiguous word with a high weight in two documents can result in high similarity between these two documents, while in fact they are not similar because that word has different meanings in each document. This result has been noted as well by Caruso et al. [83] but in a different context.

3. - *Variable*: Letter normalization.
   - *Value*: To be normalized.
   - *Rationale*: We believe that a few letters in Arabic are rather equivalent (including alif normalization, taa marbouta normalization, alif maqsoura normalization, and tatweel normalization like أمــازون). Normalizing such letters in Arabic is important where the same word can be written with a slight modification. As an example, the word مكتبة can be found written as مكتبه. The two words are exactly the same meaning.

4. - *Variable*: Diacritics.
   - *Value*: To be removed.
   - *Rationale*: Diacritics feature is one of the additional dimensions that increases the complexity of Arabic language. While diacritics can affect the word meaning in Arabic (like عَمَّان and عُمَان), Arabic words are rich of inflectional suffixes and prefixes so that one word with the same meaning can have many inflectional forms. Therefore, diacritics are often removed.

5. - *Variable*: Punctuation marks.

- *Value*: To be removed.
- *Rationale*: Anything related to punctuation marks such as periods, commas, and semicolon, are removed. They have no value and are useless in the bag-of-words model.

6. - *Variable*: Non Arabic words.
   - *Value*: To be removed.
   - *Rationale*: Since this study focuses on Arabic text, non-Arabic words are removed.

7. - *Variable*: Numbers (numeric digits).
   - *Value*: To be removed.
   - *Rationale*: Including numbers and digits as features increases the space dimensionality. Furthermore, since we deal with queries of words, they are useless to keep the numeric digits.

8. - *Variable*: Non alphabetic words.
   - *Value*: To be removed.
   - *Rationale*: Non alphabetic words might present noise in the analysis, and it is prudent to remove them.

9. - *Variable*: Stop words.
   - *Value*: To be removed.
   - *Rationale*: Stop words are those words occurring frequently in any language. Since they are not very discriminative features for information retrieval and mining applications, it is often recommended to remove them [33].

The preprocessing steps are performed in the following order:

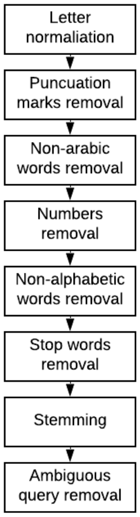


Figure 5.1: The order of the preprocessing steps.

Additionally, before removing the ambiguous query, it is entered through the letter normalization and then the stemming stages only.

5.1.4.2 Feature Representation Variables

The other variables that are related to features representation, with their values as well as the rationales, are discussed below.

1. - *Variable*: Word frequency.
   - *Value*: To be used.
   - *Rationale*: Word frequency is important as it indicates how relevant this document is to that word. Therefore, it will significantly affect the similarity computation between search results.

2. - *Variable*: Words to keep: the top most common words in all the string attribute values to keep, plus any words that are as common as the least common word amongst the top ones.
   - *Value*: Here the value depends on the case as follows:
     - In case of inner page (i.e., full document): 300 words
     - In case of title and snippet: Since both engines, Google and Bing, generate different lengths of title and/or snippet and to make this neutralized as much as possible, this factor was calculated with same equation for both engines, based on statistics related to the number of words and terms as follows:

       $$wordsToKeep(engine) = \frac{T}{R} \tag{5.1}$$

       where T: total number of terms detected in title and snippet for all results of all ambiguous and clear queries.
       R: number of search results that have terms for all ambiguous and clear queries.
       Based on the above equation, the calculated value is **25** for Google and **17** for Bing. Notably, Bing snippets are shorter than Google ones. As for the mix of Google and Bing, the average of the two was taken, which is **21**.
   - *Rationale*: The idea of words to keep is to attempt to keep the top-N most common words among the lexicon (i.e., the dictionary). Otherwise, all words will be kept. Since the title and snippet of search results are small text, we choose a small number of words to keep. For inner pages, in contrast, we choose a larger number since it is more likely to contain more words. The values above were calculated based on some statistics related to number of words and terms of the search results of search engine. The detailed calculations and statistics can be found in Appendix B.

3. - *Variable*: Words frequency damping.
   - *Value*: Not used.
   - *Rationale*: As previously mentioned, repetition of the same word in the document considerably affects the similarity computation. However, to provide more stability to the similarity computation, a damping function is often applied. However, Aggarwal and Zhai [62] pointed out that clustering have shown better performance without damping, especially if the underlying data sets are relatively clean and contain no or few spam documents.

4. - *Variable*: Inverse document frequency.
   - *Value*: To be used.
   - *Rationale*: The informative words are those words occurring frequently in a document but infrequently across documents. This is the idea behind using this type of normalization.

5. - *Variable*: Minimum word frequency to keep.
   - *Value*: 1.
   - *Rationale*: Since the preprocessing of search results removes the noises, there is no plausible reason to remove the words whose frequencies are less than a specific value. In other words, all words with frequencies larger than 1 will be kept.

6. - *Variable*: Feature vector length normalization.
   - *Value*: To be used.
   - *Rationale*: It is important to normalize the documents so that similarity computations are correct and reasonable. The reason is discussed in the rationale of distance function variable in the next subsection.

5.1.4.3 Clustering Algorithm Variables

The variables related to the machine learning clustering task, which are used to cluster search results, are discussed below.

1. - *Variable*: Clustering algorithm.
   - *Value*: K-means algorithm.
   - *Rationale*: K-means[1] is one of K-family clustering algorithms, which form the basis for other types of clustering algorithms. K-means is a good choice for a wide range of information retrieval tasks [1]. It has been shown that such algorithms (like K-means and K-medoids) are very appropriate for text clustering [74, 73]. Moreover, K-means tends to converge very quickly in

[1] Detailed description of the K-means algorithm can be found in the Background chapter (Chapter 2).

practice [1] and this was observed while running some dry runs. K-means is also more efficient than hierarchical clustering algorithms. In particular, implementations of K-means require $O(KN)$ time complexity, while hierarchical algorithms require $O(n^2)$.

2. - *Variable*: Number of clusters (K).
   - *Value*: It depends on the experiment itself as follows: for most experiments it is set to the *number of predefined meanings* of an ambiguous query. Only for experiments that are performed on plain benchmarks to study the influence of the desired factors it is set to best value according to Calinski-Harabasz criterion [84].
   - *Rationale*: As we deal with queries with predefined meanings, the number of such meanings are known, thus the best value of K can be determined early. However, this study includes experiments that deal with dynamic determination of K, by using Calinski-Harabasz method [84]. A study [85] shows that silhouette and Calinski-Harabasz methods are amongst the best for determining K dynamically. Moreover, Calinski-Harabasz is widely used in cluster validity and is supported by Weka.

3. - *Variable*: Distance measure function.
   - *Value*: Euclidean distance.
   - *Rationale*: As search results clustering deals with text, the most appropriate choice for the similarity (distance) function is the cosine similarity [33]. Therefore, the Euclidean distance function is used for the normalized document lengths. It is worth noting that an interesting property of Euclidean distance is that, if applied to normalized vectors, it will give you the same ranking of similarity as the cosine does [86].

4. - *Variable*: Initialization strategy for choosing seeds (centroids).
   - *Value*: Kmeans++.
   - *Rationale*: K-means method aims at finding the clustering that minimizes the intracluster variance of instances (i.e., the sum of squared distances between each data point being clustered and its cluster centroid). However, it is NP-hard problem to find the globally optimal solution. To avoid possible poor clustering, Kmeans++ was proposed as an approximation algorithm for this problem (i.e., optimization problem) [87].

5. - *Variable*: Seed.
   - *Value*: 10.
   - *Rationale*: This number is used for randomization when the random mode is selected for the initialization strategy. However, this value has no direct effect because the initialization strategy is Kmeans++ and not at random. The default value is kept.

6. - *Variable*: Replacing missing values with mean/mode.
   - *Value*: Not used.
   - *Rationale*: We deal with highly sparse data; therefore, it would be impractical to replace every missing value with the mean/mode.

7. - *Variable*: Max number of iterations.
   - *Value*: 500.
   - *Rationale*: The default value is kept. From the preliminary results, the algorithm converged with a few number iterations (between 2 and 10 iterations). Thereby, it has no direct effect, and it is safe to keep the default value as is.

5.1.5 Mechanism

The subjects in terms of experimental research in this study (i.e., the objects to which we apply the experiment conditions) are the ambiguous queries along with their collected search results. The flowchart in Figure 5.2 abstracts the main procedure of experiments, which starts from line #8 on the left side of the same figure.

Given the factors of interest and the neutralized variables, each of these ambiguous queries along with its search results and its meanings were used as input to that experiment procedure, depending on type of benchmark. In this part of study (i.e., concerned with the influence of feature sources and feature space representations), all type of benchmarks were used.

The left side of Figure 5.2 shows the main steps that were followed when running the experiments. This procedure was applied first on mixing-based human-annotated benchmarks including Google, Bing, and mix of Google and Bing. For each of these benchmarks, the number of ambiguous queries is 30. Each of these queries has a number of search results dependent on the number of its predefined meanings. The base number of search results per meaning is 30 as well[2].

Then it was applied on BRF-based benchmarks including Google, Bing, and mix of both. In BRF-based experiments, the base number of search results (i.e., per meaning number) was changed just to check whether changing data size would affect the performance. This was achieved by running the experiments against 10, 20, and 30 data sizes.

Finally, experiments of this procedure were run for plain benchmarks including Google and Bing, separately. The core difference in procedure between the runs of these experiments and the runs of previous experiments is in determining number of clusters (K). In previous experiments, K is determined based on the number of meanings. In these experiments, K is dynamically determined using Calinski-Harabasz criterion [84] instead.

[2] That is, if we have a query with two meanings, then this query would have 30 search results for meaning1 plus 30 search results for meaning2, which is equal to 60 search results

```
set data processing variables
set vector-space related variables
set clustering-related variables
set type of benchmark
let selection_source_modes = [TITLE_ONLY, TITLE_WITH_SNIPPET,
  SNIPPET_ONLY, INNER_PAGE]
let space_representation_modes = [SINGLE_WORDS,
  PHRASES_OF_2_GRAMS, SINGLE_WORDS_WITH_2_GRAMS,
  SINGLE_WORDS_WITH_2_AND_3_GRAMS]
foreach query in benchmark do
main_exp_procedure:
  foreach source_mode in selection_source_modes do
    foreach representation_mode in space_representation_modes do
      Get the labeled raw search results
      Prepare the raw search results given source_mode and data
      processing variables
      Build the vectore space dataset given representation_mode and
      vector-space related variables
      Perform clustering given clustering-related variables
      Evaluate and output evaluation metrics
```

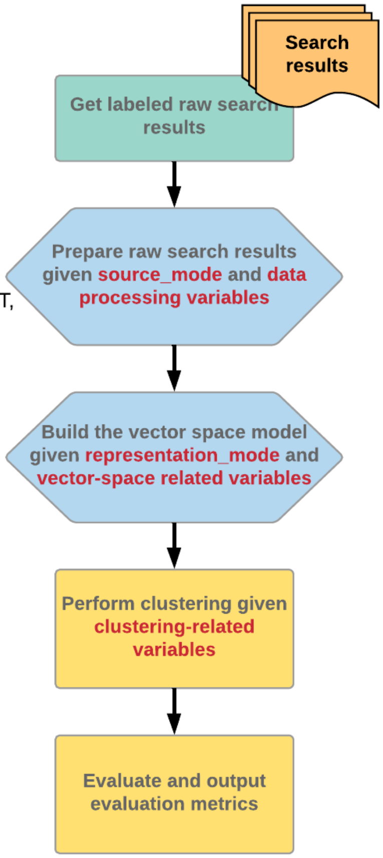

Search results
Get labeled raw search results
Prepare raw search results given source_mode and data processing variables
Build the vector space model given representation_mode and vector-space related variables
Perform clustering given clustering-related variables
Evaluate and output evaluation metrics

Figure 5.2: A pseudocode with a flowchart of the main steps.

5.1.6 Evaluation Methodology

Cluster evaluation is generally a challenging task. In this study, human assessors participated in building ground truth benchmarks by manually labeling search results. This point was the main motivation of evaluating experiments objectively. Thereby, this study followed an objective external evaluation method to leverage the manual process of labeling that was done when building benchmarks.

The external evaluation method followed in this study is the classes-to-clusters method [31]. In a nutshell, the idea is to find the minimum error assignment class labels to clusters, with the constraint that a class label can be assigned to one cluster only.

So for each single dataset (i.e., within a benchmark) that includes an ambiguous query, predefined meanings, and search results, the classes in this case are represented by the predefined meanings. As an example, for the query Amazon (أمازون), the classes in this case are river (نهر) and company (شركة).

After mapping classes to clusters, the popular evaluation metrics were calculated by first calculating the confusion matrix. These metrics[3] include:

- Accuracy.
- Weighted precision.
- Weighted recall.
- Weighted macro F-measure.
- Averaged macro F-measure.
- Averaged micro F-measure.

As shown in the pseudocode in Figure 5.2 and after executing the main procedure for all queries, each query would have the values for all above evaluation metrics. However, because the dependent variable of these experiments is the effectiveness of search results clustering, the weighted macro F-measure, which is a harmonic mean of precision and recall, is used to measure the effectiveness of search results clustering.

After having all values of effectiveness for all subjects (i.e., queries) and for all independent variables (i.e., feature source and feature representation mode) and to judge the hypotheses, we need to check whether there is a significant difference between the conditions and try to find the best combination that results in best effectiveness.

Since the design of experiments is two-way with two within-subjects factors (i.e., within-group design and number of independent variables are two), this was done as follows:

- The data of conditions were subjected to normality test, knowing that it is very hard to expect that any sampled data would have precise properties of a probability distribution such as normal distribution.
- Then the data were subjected to a significance test. In particular, the significance test used is the repeated measures ANOVA test (F-test).
- A boxplot analysis was performed to depict the data distribution for each condition and to calculate F-measure values including mean, 25th percentile, 50th percentile (median), 75th percentile, minimum, and maximum.
- PostHoc, a pairwise comparisons test, was used to determine which particular conditions differ from which other conditions test.
- The best combination of conditions was found by using the mean and median together as well as the significance test.

[3] The definitions of the evaluation metrics can be found in the Background chapter (Chapter 2).

5.2 Supervised Approach to Unsupervised Learning (SAUL): A Proposed Approach for Search Results Clustering

This section shows the design of experiments comparing between the traditional approach and the proposed approach for disamiguating real search results of an ambiguous query. While the traditional approach is unsupervised and dynamically clusters search results, the proposed approach leverages the power of supervised datasets in unsupervised learning, with the help of the notion of blind relevance feedback. It is worth noting that since these experiments work with real search results of ambiguous queries, plain benchmarks are the main theme of the datasets used in this work. Moreover, the *evaluation* of this type of experiments is still challenging. This is mainly because we deal with datasets that might contain search results that do not belong to any of predefined meanings, leaving some search results unlabeled.

5.2.1 Traditional Approach: Dynamic Clustering

This approach represents the traditional work of disambiguating search results of an ambiguous query using clustering. For the sake of comparison, a baseline approach is implemented to be compared with the proposed approach. This baseline approach is challenged by two factors:

- Determining the number of meanings; thus, number of clusters (i.e., the K value of the algorithm).
- Evaluating against training datasets that are manually labeled using predefined meanings. Moreover, these datasets contain results that do not belong to any predefined meanings, or equivalently have no labels.

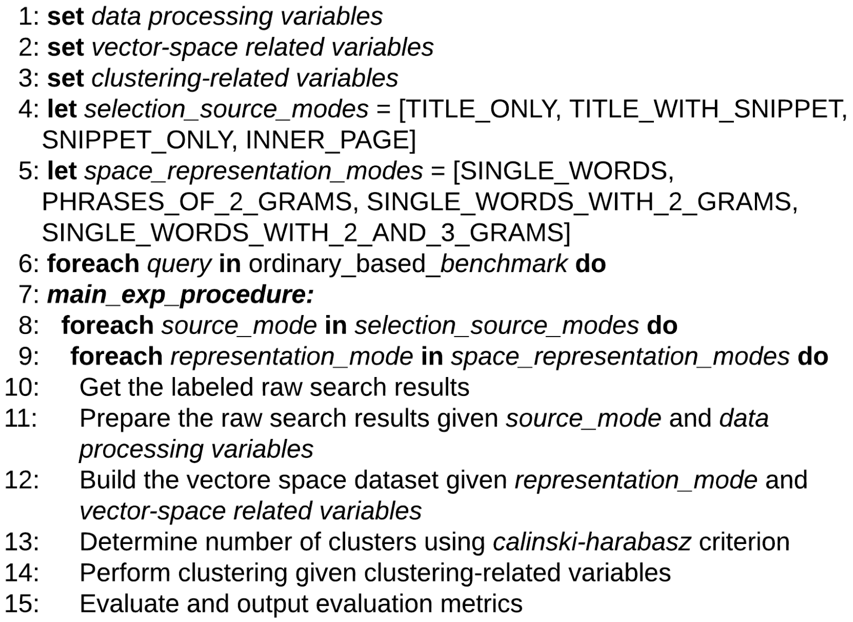
The first point was addressed by using a popular method for determining K dynamically, called Calinski-Harabasz criterion [84]. A comparative study [85] shows that silhouette and Calinski-Harabasz methods are amongst the best. Moreover, the Calinski-Harabasz method is widely used in cluster validation and is supported by Weka.

All other parameters such as those related to data preprocessing, feature representation, and clustering algorithm were set to the same ones as described previously in Section 5.1.4.

The details of how the evaluation was performed for this approach are discussed later in Section 5.2.3.

5.2.1.1 Mechanism

The plain benchmarks were used for this approach, including Google and Bing. These benchmarks are composed of 11 and 15 ambiguous queries for Google and Bing, respectively. For each search engine, the steps shown in Figure 5.3 were performed.



```
set data processing variables
set vector-space related variables
set clustering-related variables
let selection_source_modes = [TITLE_ONLY, TITLE_WITH_SNIPPET,
SNIPPET_ONLY, INNER_PAGE]
let space_representation_modes = [SINGLE_WORDS,
PHRASES_OF_2_GRAMS, SINGLE_WORDS_WITH_2_GRAMS,
SINGLE_WORDS_WITH_2_AND_3_GRAMS]
foreach query in ordinary_based_benchmark do
main_exp_procedure:
  foreach source_mode in selection_source_modes do
    foreach representation_mode in space_representation_modes do
      Get the labeled raw search results
      Prepare the raw search results given source_mode and data
      processing variables
      Build the vectore space dataset given representation_mode and
      vector-space related variables
      Determine number of clusters using calinski-harabasz criterion
      Perform clustering given clustering-related variables
      Evaluate and output evaluation metrics
```

Figure 5.3: A pseudocode describing the main steps of the plain experiments.

After performing the clustering step for each ambiguous query, the evaluation metrics were computed and stored for later analysis.

5.2.2 A Supervised Approach to Unsupervised Learning (SAUL)

This thesis proposes an approach that takes advantage of the models that were built using clear queries. More specifically, the idea is to treat these models as supervised learning models and use them to classify new unseen search results of ambiguous queries. This approach was applied separately on both Google and Bing search engines. What makes this approach more challenging is how it will be evaluated. This approach was evaluated using the same methodology used for the traditional approach. The details of how the evaluation was performed are discussed later in Section 5.2.3.

As the initial clustering model is built using the training dataset (i.e., search results of clear queries), the number of clusters (K) needs to initially be determined. For example, if Amazon query (أمازون) has two meanings: river (نهر) and company (شركة), then the clear queries are amazon company (شركة أمازون) and amazon river (نهر أمازون). Therefore, for the initial clustering model built for Amazon, K will be set to 2. All other parameters such as those related to data preprocessing, feature representation, and clustering algorithm were set to the same ones as described previously in Section 5.1.4.

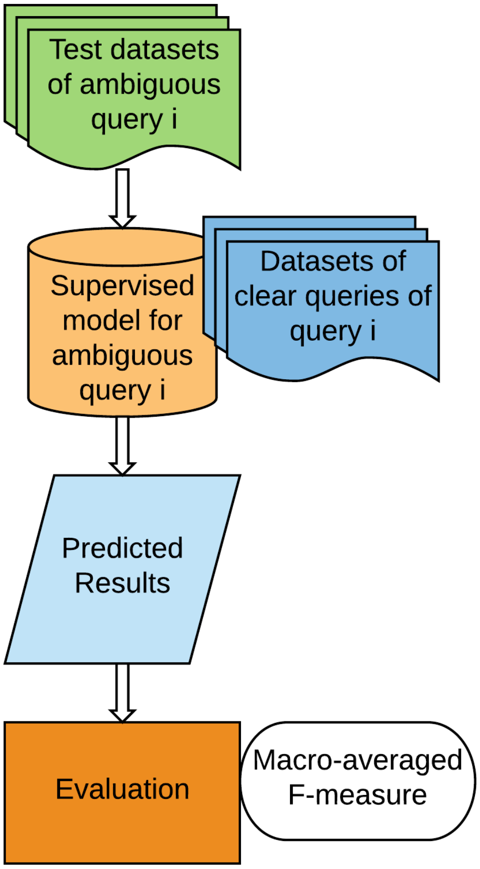
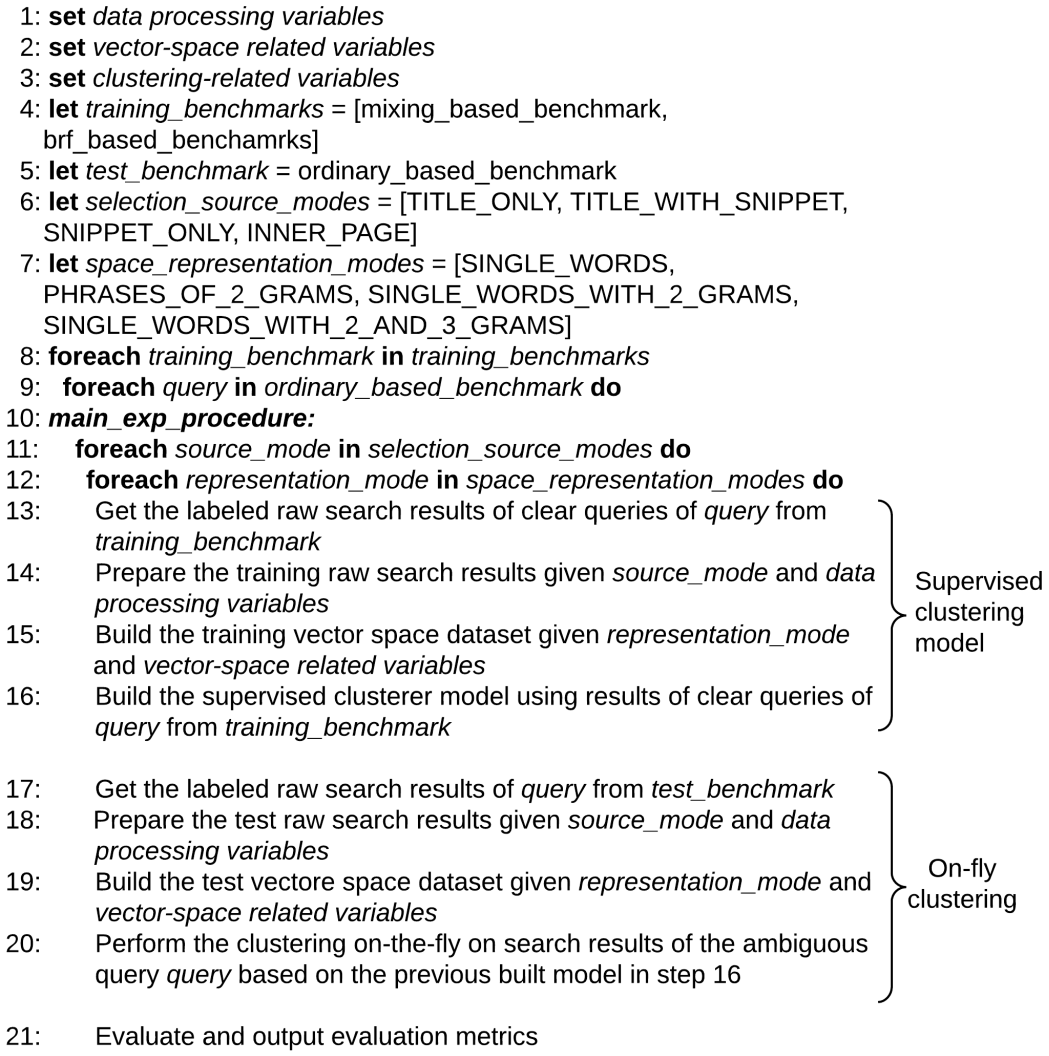

Test datasets of ambiguous query i
Supervised model for ambiguous query i
Datasets of clear queries of query i
Predicted Results
Evaluation
Macro-averaged F-measure

Figure 5.4: A diagram depicting the design model of the proposed approach.

5.2.2.1 Mechanism

The experiments here were performed in two stages in order to be compared with the traditional approach. In the first stage, the datasets of mixing-based human annotated benchmarks (i.e., manually annotated) were considered as training datasets, and the datasets of plain benchmarks were considered as test datasets. The second stage is similar; the only difference is that the training datasets were taken from BRF-based benchmarks instead.

As shown in Figure 5.4, a supervised model was built for every ambiguous query based on datasets of clear queries. Then, these models were tested and evaluated against test datasets that contain search results of those ambiguous queries, by clustering the unseen search results on the fly. As in the baseline approach, this approach was applied on Google and Bing, separately. Figure 5.5 abstracts the mechanism in pseudocode style.



```
 1: set data processing variables
 2: set vector-space related variables
 3: set clustering-related variables
 4: let training_benchmarks = [mixing_based_benchmark,
    brf_based_benchamrks]
 5: let test_benchmark = ordinary_based_benchmark
 6: let selection_source_modes = [TITLE_ONLY, TITLE_WITH_SNIPPET,
    SNIPPET_ONLY, INNER_PAGE]
 7: let space_representation_modes = [SINGLE_WORDS,
    PHRASES_OF_2_GRAMS, SINGLE_WORDS_WITH_2_GRAMS,
    SINGLE_WORDS_WITH_2_AND_3_GRAMS]
 8: foreach training_benchmark in training_benchmarks
 9:   foreach query in ordinary_based_benchmark do
10: main_exp_procedure:
11:    foreach source_mode in selection_source_modes do
12:      foreach representation_mode in space_representation_modes do
13:        Get the labeled raw search results of clear queries of query from      ⎫
           training_benchmark                                                   ⎪
14:        Prepare the training raw search results given source_mode and data    ⎪ Supervised
           processing variables                                                 ⎬ clustering
15:        Build the training vector space dataset given representation_mode    ⎪ model
           and vector-space related variables                                   ⎪
16:        Build the supervised clusterer model using results of clear queries of ⎪
           query from training_benchmark                                        ⎭

17:        Get the labeled raw search results of query from test_benchmark      ⎫
18:        Prepare the test raw search results given source_mode and data       ⎪
           processing variables                                                 ⎪ On-fly
19:        Build the test vectore space dataset given representation_mode and   ⎬ clustering
           vector-space related variables                                       ⎪
20:        Perform the clustering on-the-fly on search results of the ambiguous ⎪
           query query based on the previous built model in step 16             ⎭

21:        Evaluate and output evaluation metrics
```

Figure 5.5: A pseudocode describing the main steps of SAUL experiments.

5.2.3 Evaluation methodology

As the plain datasets (i.e., search results of ambiguous queries) might contain search results with no labels, the classes-to-clusters evaluation method[4] (the last stage in Figure 5.4) in Weka [31] was modified as follows:

1. The resulting clusters can now be larger than or equal to the number of meanings (i.e., K >= C, where K is the number of clusters, and C is the number of actual

[4] You can know more about classes-to-cluster method in the Background chapter (Chapter 2)

classes, which are the meanings of the ambiguous query). Previously, this was prohibited in Weka; K must be equal to C.

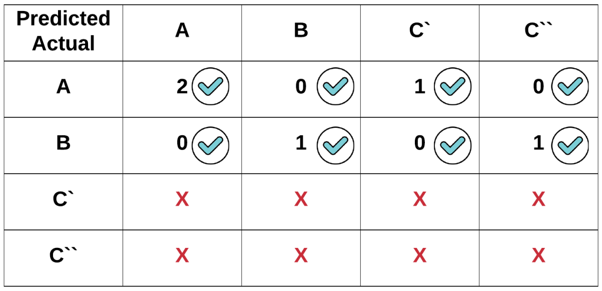
2. The missing labels must not affect the core logic of classes-to-clusters mechanism. This is because in such external validation, we care much more about how successful the machine learning algorithm is able to separate the labeled instances into the appropriate clusters.

3. The clusters, which have all their instances unlabeled, would be ignored. This means that these unlabeled instances will not be considered in the confusion matrix.

4. The clusters that have all their instanced labeled, would be taken into account, and their instances would be included in the confusion matrix.

5. The clusters that contains unlabeled instances, these unlabeled instances would be ignored in the confusion matrix.

6. Based on the above and after executing the core logic of classes-to-clusters mechanism, there could be clusters that are not assigned to classes (i.e., the case where the resulting clusters >= the actual classes). These clusters can have labeled instances. These labeled instances should not be excluded from the evaluation and should be taken into account. In fact, these labeled instances affect the true positive rate only (i.e., recall), thus the F-measure and accuracy. So, these instances will be counted when calculating the true positive rate. These instances, though, do not affect the precision. Figure 5.6 depicts an example of confusion matrix of ambiguous query with two meanings A and B as well as two resulting unknown clusters C' and C''. The ✓mark means that the value is taken into account, whereas the ✗ mark means that the value will be neglected. This confusion matrix shows that the raws of C' and C'' are always neglected and their values will be 0 because the assumption is there are no actual instances from them. However, the columns of C' and C'' are taken into account because actual instances of A and/or B could fall into any of these two unknown clusters.



Predicted Actual	A	B	C`	C``
A	2 ✓	0 ✓	1 ✓	0 ✓
B	0 ✓	1 ✓	0 ✓	1 ✓
C`	X	X	X	X
C``	X	X	X	X

Figure 5.6: An example confusion matrix for an ambiguous query with two meanings.

After performing the evaluation method above, the popular evaluation metrics were calculated by first calculating the confusion matrix[5]. These metrics are the same metrics mentioned for the previous experiments (Section 5.1.6). The same significance test procedure is followed as well.

We took the best combination of the traditional approach and compared it with the best combination of the second approach. After performing the significance tests, the mean and median of the best combination of the baseline approach were compared with the mean and median of the best combination of the proposed approach. This was done for both search engines, Google and Bing. Given such data, the increase/decrease percentages were computed to see which is better (i.e., this proposed approach or the baseline approach).

[5] The confusion matrix contains true positive, false positive, true negative, and false negative values. The definitions of these terms can be found in the Background chapter (Chapter 2).

Chapter 6

Spread Framework

In this thesis, the `Spread` framework was developed. This framework aims at providing application programming interfaces (APIs) to run search results clustering experiments. As it is intended for research purposes, it is open to the public and ready to use. Spread framework was built to be extensible so that any interested developer can extend the framework to support different types of algorithms and data that can be used in SRC experiments. Its name is inspired by the idea of spreading search results over clusters and an acronym of Search REsults Disambiguation. In this section, we describe its main components and how it is built.

The goals of Spread framework can be summarized as follows:

1. Providing a facility to load the queries with their meanings and then fetching the search results for ambiguous and clear queries from different Web search engine like Google, Bing, and Yahoo.

2. Providing an interface for human relevance assessment for labeling search results.

3. Conducting SRC experiments with the ability of changing the variables described in Chapter 5.

4. Generating nice graphs to evaluate the clusters of search results..

5. Demonstrating the SAUL approach in action.

6.1 Implementation Technologies

This framework is built using various technologies and libraries. It is developed using *Java language*, along with *Java EE*, *Tomcat*, *Spring*, and *Hibernate*. Its code is version-controlled and managed on a public *GitHub* repository[1] with a companion site developed[2].

[1] `https://github.com/haytham-salhi/Spread`

[2] `https://haytham-salhi.github.io/Spread/`

MySQL and *MongoDB* are used for storing Web search results. Particularly, MySQL is used to store all data described in the Data collection chapter. The complete MySQL scheme can be found in the Appendix A. On the other hand, MongoDB is used optionally to store the HTML content of fetched web pages. The reason behind this is that the HTML content of web pages is considered semi-structured and can be processed later to extract structured data.

For Java machine learning APIs, we integrated *Weka* [31] into Spread to use different machine learning algorithms.

The Web interfaces that are used for investigation throughout this thesis are deployed on our own public servers on *Amazon Cloud*.

6.2 Architectural Design

To achieve the goals of Spread framework, many components were implemented. Figure 6.1 present a high level overview of the main components.

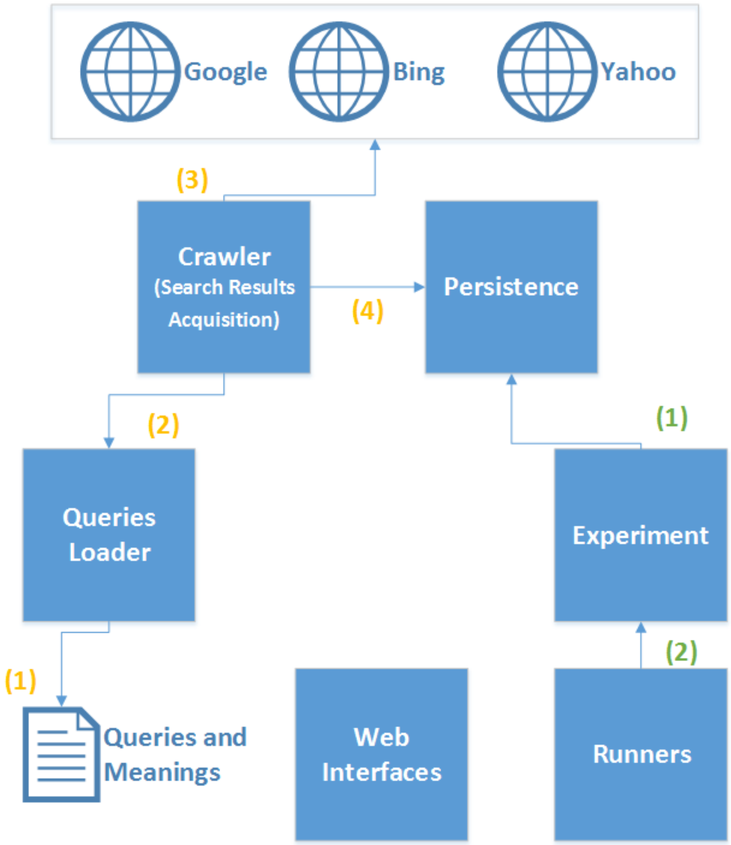


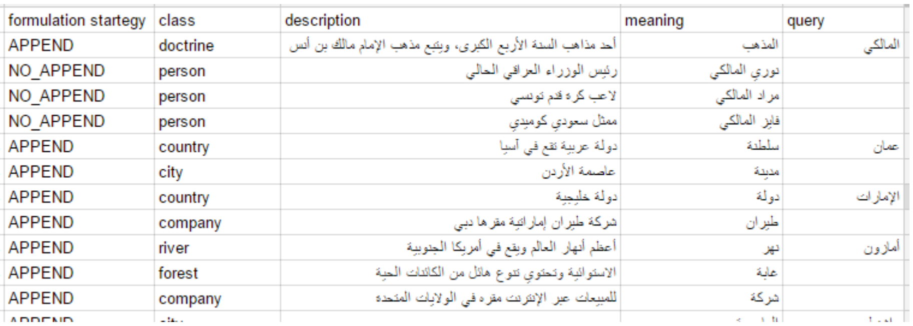
Figure 6.1: The big picture of the high-level components of Spread.

This section gives a brief overview of logical high level components, which are *search* *results acquisition* (represented by left-side components (1), (2), (3), and (4) in Figure 6.1) and *experiment* (represented by right-side components (1) and (2)).

6.2.1 Web Search Results Acquisition Components

Following the flow numbered from 1 to 4 on the left side of Figure 6.1, the stage (1) indicates that the queries along with their meanings are collected in CSV files, like the ones we collected.

Basically, the file has 5 columns: *query*, *meaning*, *description*, *class*, *formulation strategy*. Figure 6.2 shows a snapshot of the file.



formulation startegy	class	description	meaning	query
APPEND	doctrine	أحد مذاهب السنة الأربع الكبرى، ويتبع مذهب الإمام مالك بن أنس	المذهب	المالكي
NO_APPEND	person	رئيس الوزراء العراقي الحالي	نوري المالكي	
NO_APPEND	person	لاعب كرة قدم تونسي	مراد المالكي	
NO_APPEND	person	ممثل سعودي كوميدي	فايز المالكي	
APPEND	country	دولة عربية تقع في آسيا	سلطنة	عمان
APPEND	city	عاصمة الأردن	مدينة	
APPEND	country	دولة خليجية	دولة	الإمارات
APPEND	company	شركة طيران إماراتية مقرها دبي	طيران	
APPEND	river	أعظم أنهار العالم ويقع في أمريكا الجنوبية	نهر	أمازون
APPEND	forest	الاستوائية وتحتوي تنوع هائل من الكائنات الحية	غابة	
APPEND	company	للمبيعات عبر الإنترنت مقره في الولايات المتحده	شركة	

Figure 6.2: A snapshot of the structure of CSV queries file.

The *query* column represents the ambiguous query name (i.e., the query keywords). The *meaning* column represents the meanings of that ambiguous query. These meanings were collected with the help of Wikipedia. The next two columns represent a *description* for the clear query and a *class* for the clear query, respectively. The last column is a *formulation strategy* to specify either the clear query needs to be formulated by the ambiguous query plus the meaning, or the meaning only. This is determined by specifying either APPEND for the ambiguous query plus the meaning formulation or NO_APPEND for the meaning only formulation.

After preparing these files, the queries loader in the stage (2) parses the CSV and passes the list of ambiguous queries to the crawler component. The crawler component in the stage (3) triggers the fetcher component for fetching from a specific search engine. Afterwards, in the stage (4) the fetcher component communicates with the persistence component to persist the search results for all queries loaded.

6.2.2 Conducting Experiments

The second important part of the framework is concerned with running search result clustering experiments. Figure 6.3 depicts the subcomponents of the high-level *experiment* component (stage (1)) on the right side of Figure 6.1. The *experiment* component has four main sub-components as follows:

- *Data component*: this component provides the required API for reading search results datasets by communicating with the persistence component. It provides datasets of ambiguous query and datasets for clear queries.

- *Search item preparation component*: this component represents the core of data preprocessing and feature generation. This includes methods for:
  - Providing the required APIs to set the feature sources (i.e., title, snippet, title with snippet, or inner page) and feature space representations (e.g., single-words or phrases).
  - Arabic text preprocessor which includes the methods for: stemming, letter normalization, stop words removal, diacritics removal, non-Arabic words removal, non-alphabetic words removal, tokenization, punctuation marks removal, and so on.
  - Converting text data from string representation into vector-space representation. This process is referred to as vectorization.
  - Preparing multi-dimensional feature vectors so that they can be input to the clustering algorithm.
- *Clusterer*: this component is responsible for building the clustering model and provides the required APIs to set clustering algorithm variables such as the K variable, the initialization method, and the max number of iterations.
- *Evaluation*: this component is used after generating the clusters to evaluate the clustering model using the *classes to cluster* evaluation method.

In addition, the *runners* component (stage (2)) defines the experiments to run, generates some useful charts, and outputs the results into a structure of directories. This is achieved by orchestrating the APIs of the four subcomponents.

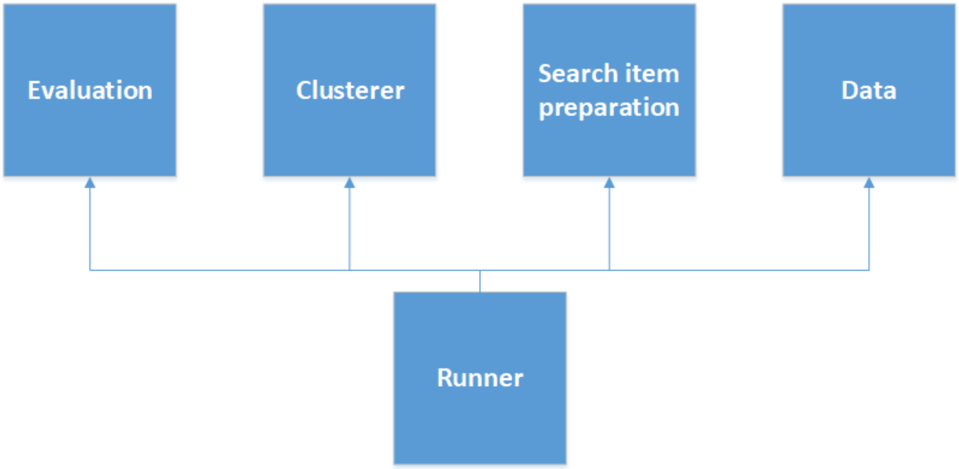


Figure 6.3: The subcomponents of *experiment*.

6.3 Experiment Pipeline

After discussing the architectural design of the framework, this section presents the big picture of experiment pipeline after integrating all experiment components. Figure 6.4 shows how the components communicate and are integrated together, combined with the set of variables, parameters, and outputs. Custom runners can be developed to run experiments with different configurations.

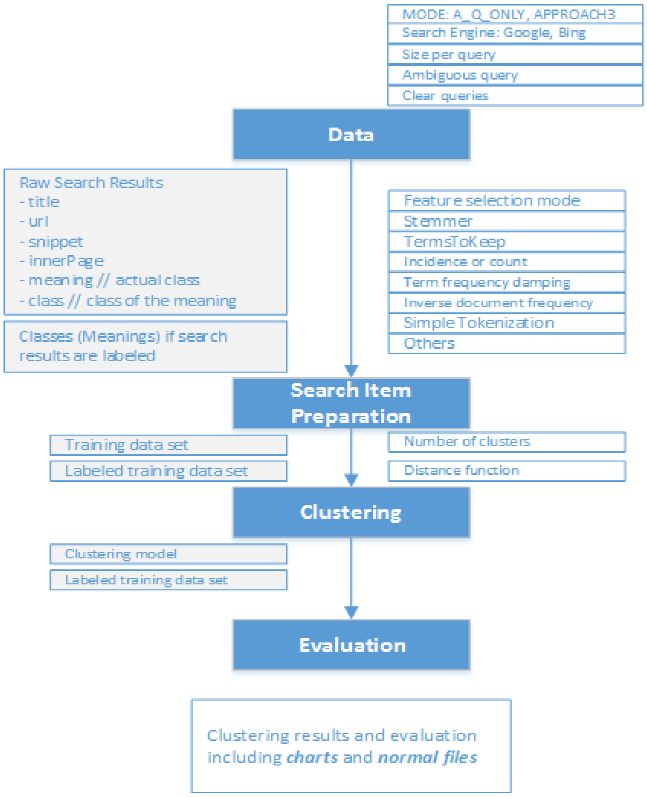


Figure 6.4: A high-level diagram of experiment pipeline.

6.4 Experiment Demonstration

Let's run an SRC experiment that uses K-means algorithm, with a dynamic determination of K using the framework;

1. First, you need to checkout the framework from the public Github repository: `https://github.com/haytham-salhi/Spread`.

   The framework currently supports four runners:

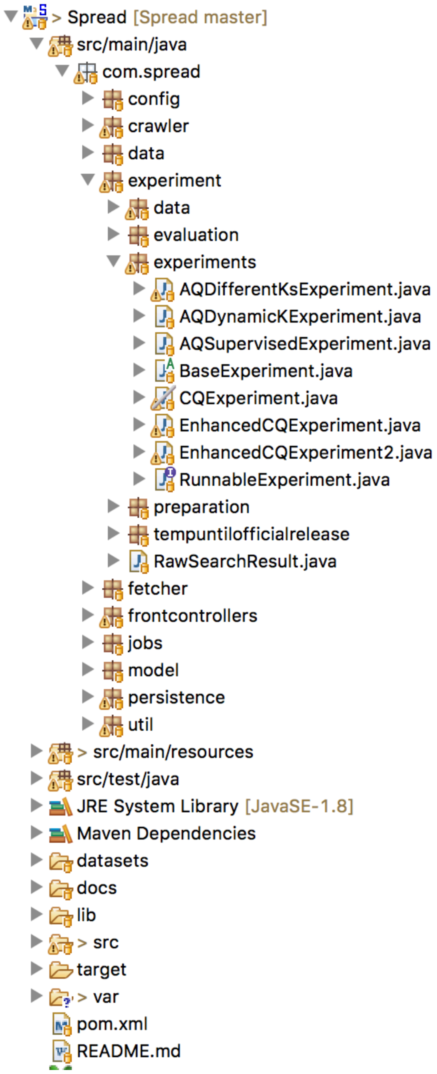
   (a) A runner that runs K-means on results of ambiguous queries with different K values.

(b) A runner that runs K-means on results of ambiguous queries with a dynamic determination of K (based Calinski-Harabasz method [84]).

(c) A runner that runs the proposed approach of this thesis (i.e., SAUL approach).

(d) A runner that runs K-means of results that are made up of results of clear queries.

2. After importing the project into IDE like eclipse, you will have the following structure:

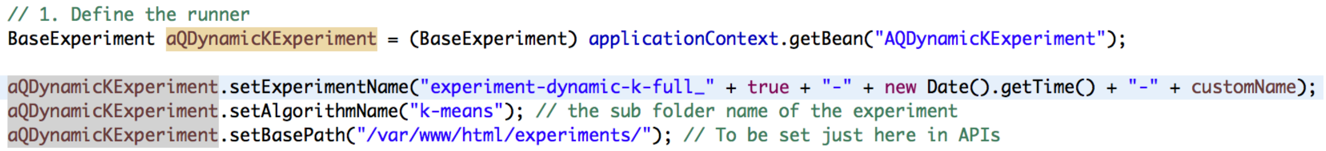


```
▼ > Spread [Spread master]
  ▼ src/main/java
    ▼ com.spread
      ▶ config
      ▶ crawler
      ▶ data
      ▼ experiment
        ▶ data
        ▶ evaluation
        ▼ experiments
          ▶ AQDifferentKsExperiment.java
          ▶ AQDynamicKExperiment.java
          ▶ AQSupervisedExperiment.java
          ▶ BaseExperiment.java
          ▶ CQExperiment.java
          ▶ EnhancedCQExperiment.java
          ▶ EnhancedCQExperiment2.java
          ▶ RunnableExperiment.java
        ▶ preparation
        ▶ tempuntilofficialrelease
        ▶ RawSearchResult.java
      ▶ fetcher
      ▶ frontcontrollers
      ▶ jobs
      ▶ model
      ▶ persistence
      ▶ util
  ▶ > src/main/resources
  ▶ src/test/java
  ▶ JRE System Library [JavaSE-1.8]
  ▶ Maven Dependencies
  ▶ datasets
  ▶ docs
  ▶ lib
  ▶ > src
  ▶ target
  ▶ > var
    pom.xml
    README.md
```

Figure 6.5: The structure of Spread framework.

3. Open a new driver class and start defining an object instance of the experiment,

as shown in Figure 6.6.

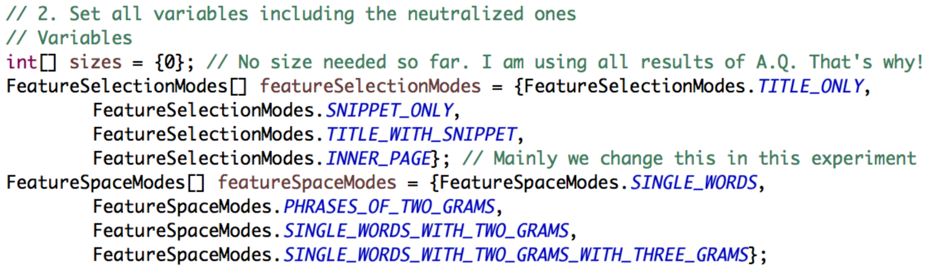


```
// 1. Define the runner
BaseExperiment aQDynamicKExperiment = (BaseExperiment) applicationContext.getBean("AQDynamicKExperiment");

aQDynamicKExperiment.setExperimentName("experiment-dynamic-k-full_" + true + "-" + new Date().getTime() + "-" + customName);
aQDynamicKExperiment.setAlgorithmName("k-means"); // the sub folder name of the experiment
aQDynamicKExperiment.setBasePath("/var/www/html/experiments/"); // To be set just here in APIs
```

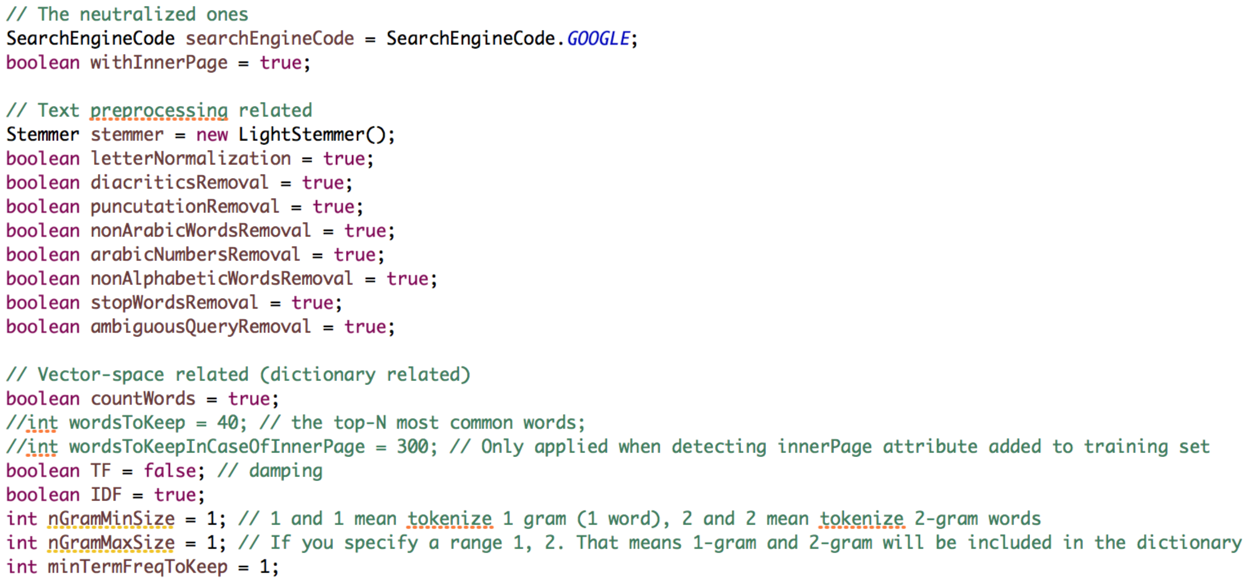
Figure 6.6: Object instance from the runner.

4. Then, you need to specify the values of the variables that will be changed during the execution of the experiment as shown in Figure 6.7 as well as the neutralized variables as shown in and Figure 6.8. Currently, this runner supports changing two variables: feature source mode and feature space mode.



```
// 2. Set all variables including the neutralized ones
// Variables
int[] sizes = {0}; // No size needed so far. I am using all results of A.Q. That's why!
FeatureSelectionModes[] featureSelectionModes = {FeatureSelectionModes.TITLE_ONLY,
        FeatureSelectionModes.SNIPPET_ONLY,
        FeatureSelectionModes.TITLE_WITH_SNIPPET,
        FeatureSelectionModes.INNER_PAGE}; // Mainly we change this in this experiment
FeatureSpaceModes[] featureSpaceModes = {FeatureSpaceModes.SINGLE_WORDS,
        FeatureSpaceModes.PHRASES_OF_TWO_GRAMS,
        FeatureSpaceModes.SINGLE_WORDS_WITH_TWO_GRAMS,
        FeatureSpaceModes.SINGLE_WORDS_WITH_TWO_GRAMS_WITH_THREE_GRAMS};
```

Figure 6.7: The values of the experiment variables.



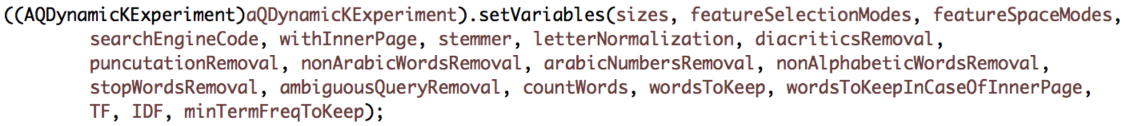
```
// The neutralized ones
SearchEngineCode searchEngineCode = SearchEngineCode.GOOGLE;
boolean withInnerPage = true;

// Text preprocessing related
Stemmer stemmer = new LightStemmer();
boolean letterNormalization = true;
boolean diacriticsRemoval = true;
boolean puncutationRemoval = true;
boolean nonArabicWordsRemoval = true;
boolean arabicNumbersRemoval = true;
boolean nonAlphabeticWordsRemoval = true;
boolean stopWordsRemoval = true;
boolean ambiguousQueryRemoval = true;

// Vector-space related (dictionary related)
boolean countWords = true;
//int wordsToKeep = 40; // the top-N most common words;
//int wordsToKeepInCaseOfInnerPage = 300; // Only applied when detecting innerPage attribute added to training set
boolean TF = false; // damping
boolean IDF = true;
int nGramMinSize = 1; // 1 and 1 mean tokenize 1 gram (1 word), 2 and 2 mean tokenize 2-gram words
int nGramMaxSize = 1; // If you specify a range 1, 2. That means 1-gram and 2-gram will be included in the dictionary
int minTermFreqToKeep = 1;
```

Figure 6.8: The values of the neutralized variables.

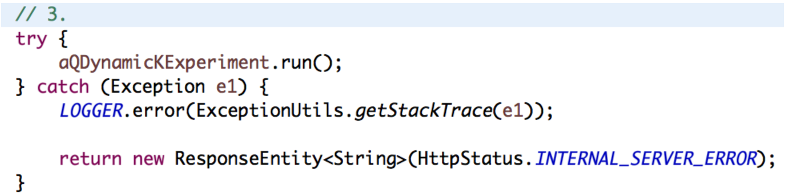
5. Then, you need to set the experiment variables as shown in Figure 6.9.



```
((AQDynamicKExperiment)aQDynamicKExperiment).setVariables(sizes, featureSelectionModes, featureSpaceModes,
        searchEngineCode, withInnerPage, stemmer, letterNormalization, diacriticsRemoval,
        puncutationRemoval, nonArabicWordsRemoval, arabicNumbersRemoval, nonAlphabeticWordsRemoval,
        stopWordsRemoval, ambiguousQueryRemoval, countWords, wordsToKeep, wordsToKeepInCaseOfInnerPage,
        TF, IDF, minTermFreqToKeep);
```

Figure 6.9: The experiment variables set API.

6. Finally, you need to call the run API, which is the last statement in the code, as shown in Figure 6.10.

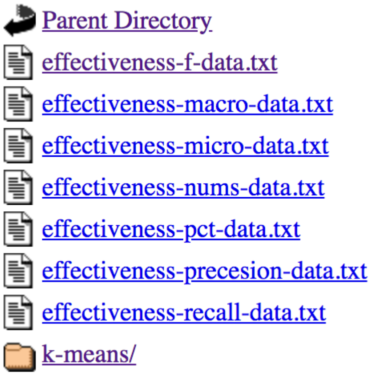


```
// 3.
try {
    aQDynamicKExperiment.run();
} catch (Exception e1) {
    LOGGER.error(ExceptionUtils.getStackTrace(e1));

    return new ResponseEntity<String>(HttpStatus.INTERNAL_SERVER_ERROR);
}
```

Figure 6.10: The experiment run API.

7. After you run the driver class using Java environment, the output will be in the path you specified as in Figure 6.6. It will look like the structure in Figure 6.11.



```
Parent Directory
effectiveness-f-data.txt
effectiveness-macro-data.txt
effectiveness-micro-data.txt
effectiveness-nums-data.txt
effectiveness-pct-data.txt
effectiveness-precesion-data.txt
effectiveness-recall-data.txt
k-means/
```

Figure 6.11: The structure of the output folder.

The output folder contains all detailed results including the evaluation metrics for all queries, the generated clusters, the detailed information about the model, and summary charts for evaluation. Figure 6.12 represents an example of the accuracy chart of Almalki (المالكي) query that shows the accuracy levels for different values of the experiment factors.

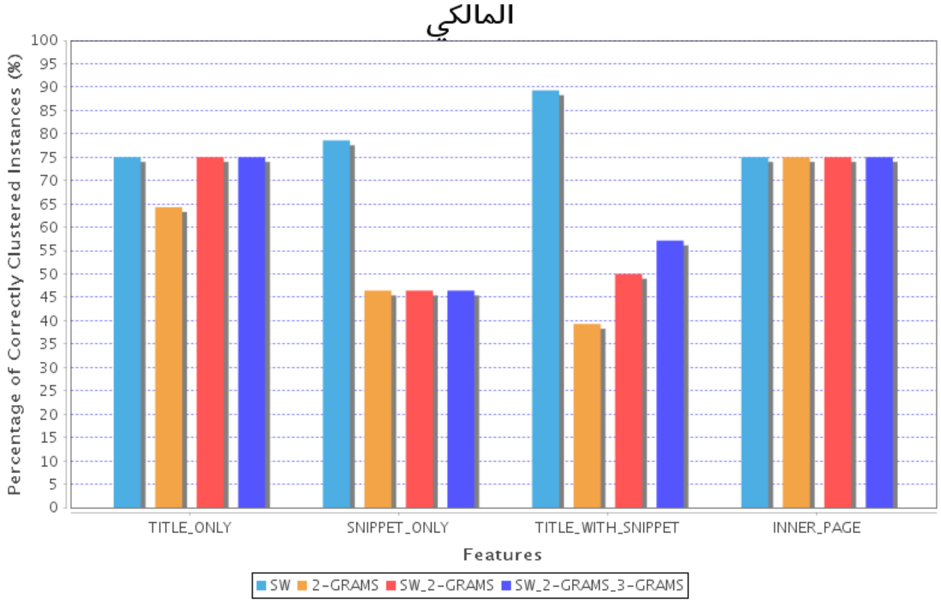
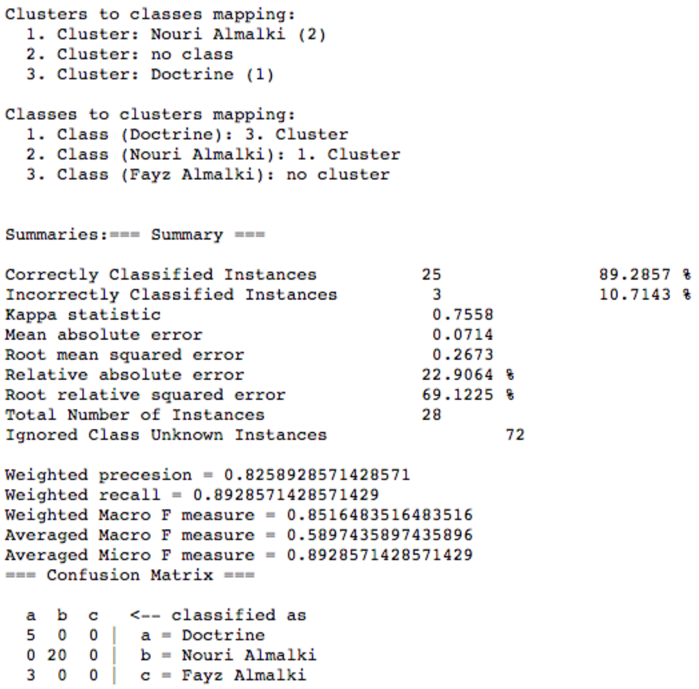


Figure 6.12: Summary of evaluation chart showing the accuracy.

Also, Figure 6.13 shows a snapshot of the evaluation text file. It shows the detailed evaluation along with the resulting clusters.



```
Clusters to classes mapping:
  1. Cluster: Nouri Almalki (2)
  2. Cluster: no class
  3. Cluster: Doctrine (1)

Classes to clusters mapping:
  1. Class (Doctrine): 3. Cluster
  2. Class (Nouri Almalki): 1. Cluster
  3. Class (Fayz Almalki): no cluster


Summaries:=== Summary ===

Correctly Classified Instances          25               89.2857 %
Incorrectly Classified Instances         3               10.7143 %
Kappa statistic                          0.7558
Mean absolute error                      0.0714
Root mean squared error                  0.2673
Relative absolute error                 22.9064 %
Root relative squared error             69.1225 %
Total Number of Instances               28
Ignored Class Unknown Instances                  72

Weighted precesion = 0.8258928571428571
Weighted recall = 0.8928571428571429
Weighted Macro F measure = 0.8516483516483516
Averaged Macro F measure = 0.5897435897435896
Averaged Micro F measure = 0.8928571428571429
=== Confusion Matrix ===

  a  b  c   <-- classified as
  5  0  0 |  a = Doctrine
  0 20  0 |  b = Nouri Almalki
  3  0  0 |  c = Fayz Almalki
```

Figure 6.13: A snapshot of evaluation text file.

6.5 SAUL Approach in Action: A Demonstration

To show that the SAUL approach can be used in real search engines without human intervention (i.e., using blind relevance feedback), we built a component that fetches the senses from wikipedia disambiguation pages. These senses will then be digested by the SAUL mechanism. This section shows the big picture of the whole solution that can be used to disambiguate search results.

In particular, Spread has an API that disambiguates search results based on the SAUL approach along with wikipedia disambiguation pages as a discovery source for senses. This API is an HTTP GET API and has the following syntax:

```
GET http://{server_name}/spread?query={query}&engine={engine}
```

Table 6.1: The Spread API parameters.

Parameter	Description
{server_name}	The server where Spread is running on.
{query}	The ambiguous query text you are looking for.
{engine}	The search engine. Possible values: **G** (for Google) and **B** (for Bing).

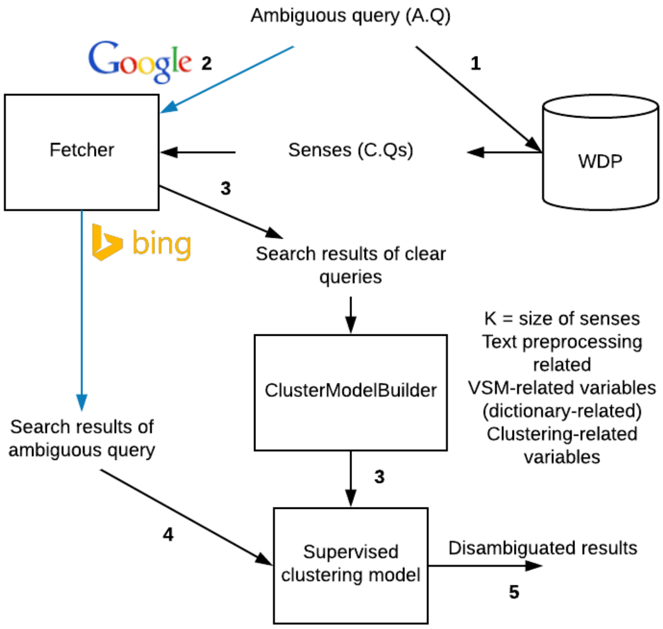

Ambiguous query (A.Q)
Google 2
1
Fetcher
Senses (C.Qs)
WDP
3
bing
Search results of clear queries
K = size of senses
Text preprocessing related
VSM-related variables (dictionary-related)
Clustering-related variables
ClusterModelBuilder
Search results of ambiguous query
3
4
Supervised clustering model
Disambiguated results
5

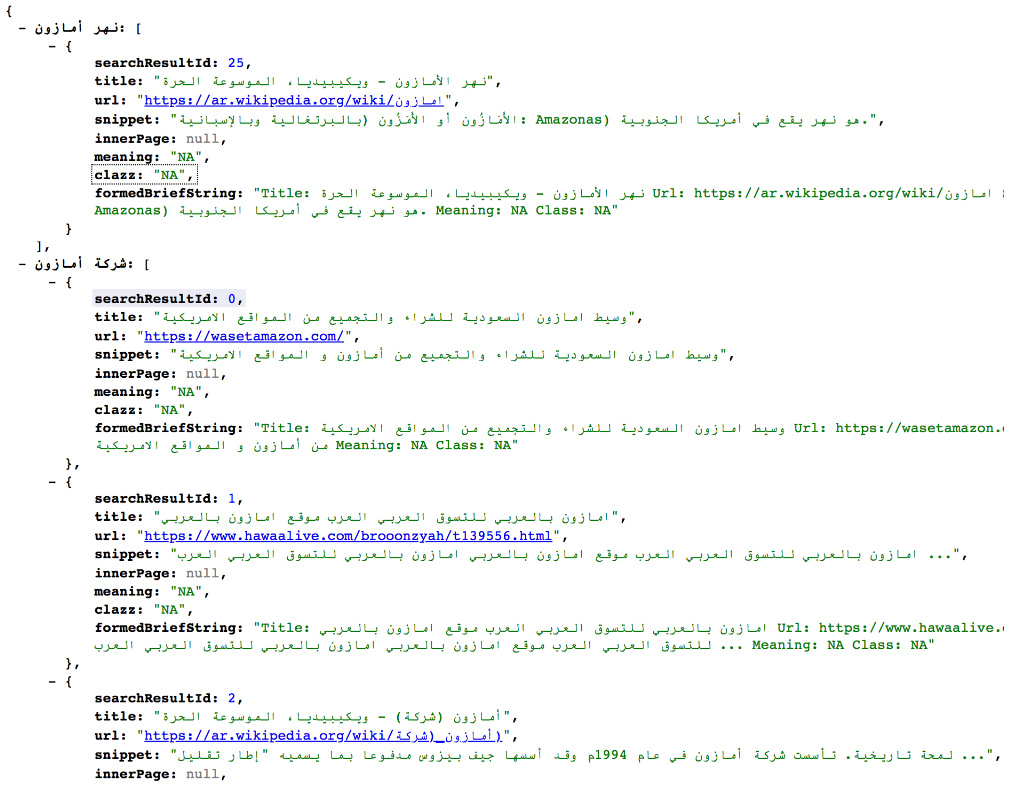
Figure 6.14: The big picture of the automated SAUL approach.

Table 6.1 describes the required parameters in the Spread API. This API consults first the WDP component that is in charge of fetching the senses from wikipedia disambiguation pages. These senses will be entered as clear queries into the fetcher component. The fetcher component fetches the search results from the specified search engine. Here is where the SAUL approach comes. The search results of clear queries are taken into the ClusterModelBuilder to build the supervised clustering model. Finally, the search results of the ambiguous query will be input to the supervised clustering model. This model produces groups of results based on the senses. Figure 6.14 briefly depicts the whole flow of the automated SAUL approach.

As an example, let us disambiguate the search results of the query Amazon (أمازون) from Bing. So the request will look like:

```
GET spread?query=%D8%A3%D9%85%D8%A7%D8%B2%D9%88%D9%86&engine=B
```

After issuing the request, the SAUL mechanism will be executed. Figure 6.15 shows the disambiguated search results that are returned from Bing. Note that there is 1 search result in the group نهر أمازون, and the remaining 199 search results are in the group شركة أمازون.



```
{
 - نهر أمازون: [
    - {
          searchResultId: 25,
          title: "نهر الأمازون - ويكيبيديا، الموسوعة الحرة",
          url: "https://ar.wikipedia.org/wiki/امازون",
          snippet: "الأمَازُون أو الأمَزُون (بالبرتغالية وبالإسبانية: Amazonas) هو نهر يقع في أمريكا الجنوبية.",
          innerPage: null,
          meaning: "NA",
          clazz: "NA",
          formedBriefString: "Title: نهر الأمازون - ويكيبيديا، الموسوعة الحرة Url: https://ar.wikipedia.org/wiki/امازون
          Amazonas) هو نهر يقع في أمريكا الجنوبية. Meaning: NA Class: NA"
      }
   ],
 - شركة أمازون: [
    - {
          searchResultId: 0,
          title: "وسيط امازون السعودية للشراء والتجميع من المواقع الامريكية",
          url: "https://wasetamazon.com/",
          snippet: "وسيط امازون السعودية للشراء والتجميع من أمازون و المواقع الامريكية",
          innerPage: null,
          meaning: "NA",
          clazz: "NA",
          formedBriefString: "Title: وسيط امازون السعودية للشراء والتجميع من المواقع الامريكية Url: https://wasetamazon.
          من أمازون و المواقع الامريكية Meaning: NA Class: NA"
      },
    - {
          searchResultId: 1,
          title: "امازون بالعربي للتسوق العربي العرب موقع امازون بالعربي",
          url: "https://www.hawaalive.com/brooonzyah/t139556.html",
          snippet: "امازون بالعربي للتسوق العربي العرب موقع امازون بالعربي امازون بالعربي للتسوق العربي العرب ...",
          innerPage: null,
          meaning: "NA",
          clazz: "NA",
          formedBriefString: "Title: امازون بالعربي للتسوق العربي العرب موقع امازون بالعربي Url: https://www.hawaalive.
          للتسوق العربي العرب موقع امازون بالعربي امازون بالعربي للتسوق العربي العرب ... Meaning: NA Class: NA"
      },
    - {
          searchResultId: 2,
          title: "أمازون (شركة) - ويكيبيديا، الموسوعة الحرة",
          url: "https://ar.wikipedia.org/wiki/أمازون_(شركة)",
          snippet: "لمحة تاريخية. تأسست شركة أمازون في عام 1994م وقد أسسها جيف بيزوس مدفوعا بما يسميه "إطار تقليل ...",
          innerPage: null,
```

Figure 6.15: Disambiguated search results of Amazon query from Bing.

Table 6.2 shows the classes to clusters mapping for the trained clustering model.

Consequently, the cluster C0 is labeled as شركة أمازون and the cluster C1 is labeled as نهر أمازون.

Table 6.2: The classes to clusters mapping of the trained clustering model.

Assigned to cluster –>	**C0**	**C1**
نهر أمازون	64	136
شركة أمازون	200	0

Chapter 7

Evaluation and Statistics

The experimental design was improved and enhanced over a number of iterations of dry runs. Afterward, a wet-run of all experiments, as designed in the previous chapter, was performed to get the complete and final results.

This chapter presents the evaluation of experiments, the statistical analysis of results, and the main findings. It reports the evaluation metrics including precision, recall, and F-measure. The statistical analysis was conducted on the F-measure as an evaluation metric. The first section is concerned with the results of the influence of feature source and feature space representation on effectiveness of search results clustering. This section is divided into three subsections according to which benchmarks are used as follows: *mixing-based human-annotated benchmarks*, *BRF-based benchmarks*, and *plain benchmarks*. The second section of this chapter shows results with respect to the proposed approach (i.e., the SAUL approach) and compares them with results of the traditional approach. Finally, different statistics were used in result analysis and inference including: mean, median, boxplot analysis, normality test, Anova test (F-test), and PostHoc test.

7.1 Influence of Features

The goal here is to find whether there is a significant difference between feature sources (i.e., title, snippet, title with snippet, and inner page) and feature space representations (i.e., single words, phrases, single words with 2-grams, single words with 2-grams and 3-grams) on the effectiveness of search results clustering.

7.1.1 Mixing-based Human-annotated Benchmarks

As a quick reminder for this type of benchmarks, a collection of search results is formed for $A.Q$ (i.e., an ambiguous query) where these results are composed of results belonging to its clear queries $C.Q_1$, $C.Q_2$, $\cdots$, evenly.

7.1.1.1 Google

For Google search engine, Figure 7.1 shows the boxplot of the weighted macro F-measure. Table 7.1 shows the mean value of F-measure for all conditions.

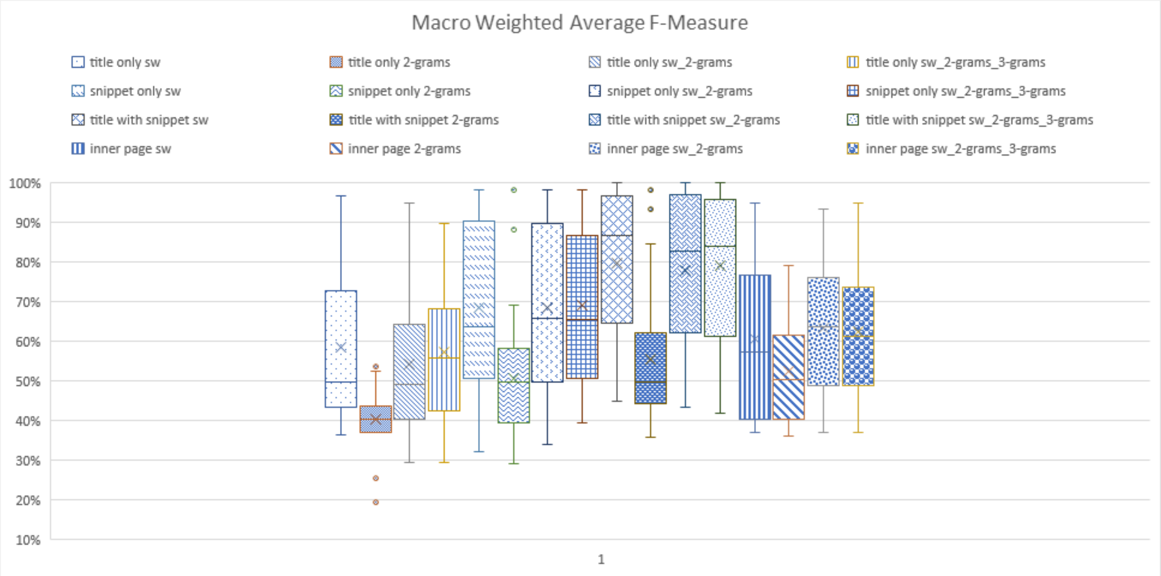


Figure 7.1: A boxplot diagram of F-measure when using MBHA benchmarks for Google.

Table C.6 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[1]. The details of raw results of both disambiguation and evaluation can be found online[2]. Table 7.1 summarizes by showing the mean of each evaluation metric for all sources and spaces. In this and other tables displaying the evaluation metrics, each table has three values: P (precision), R (recall), and F (F-measure).

Table 7.1: The mean of each metric when using MBHA benchmarks for Google.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	72%	65%	59%	77%	72%	69%	85%	82%	80%	79%	67%	61%
2-grams	73%	52%	40%	73%	59%	51%	73%	62%	55%	71%	59%	52%
sw__2-grams	74%	61%	45%	79%	72%	69%	84%	80%	78%	79%	68%	63%
sw__2-grams __3-grams	73%	63%	57%	79%	73%	69%	83%	81%	79%	79%	67%	62%

[1] https://goo.gl/Q8hthB
[2] https://goo.gl/eHg4Gu

7.1.1.2 Bing

As for Bing search engine, Figure 7.2 shows the boxplot of the weighted F-measure. Table 7.2 shows the mean value of F-measure for all conditions.

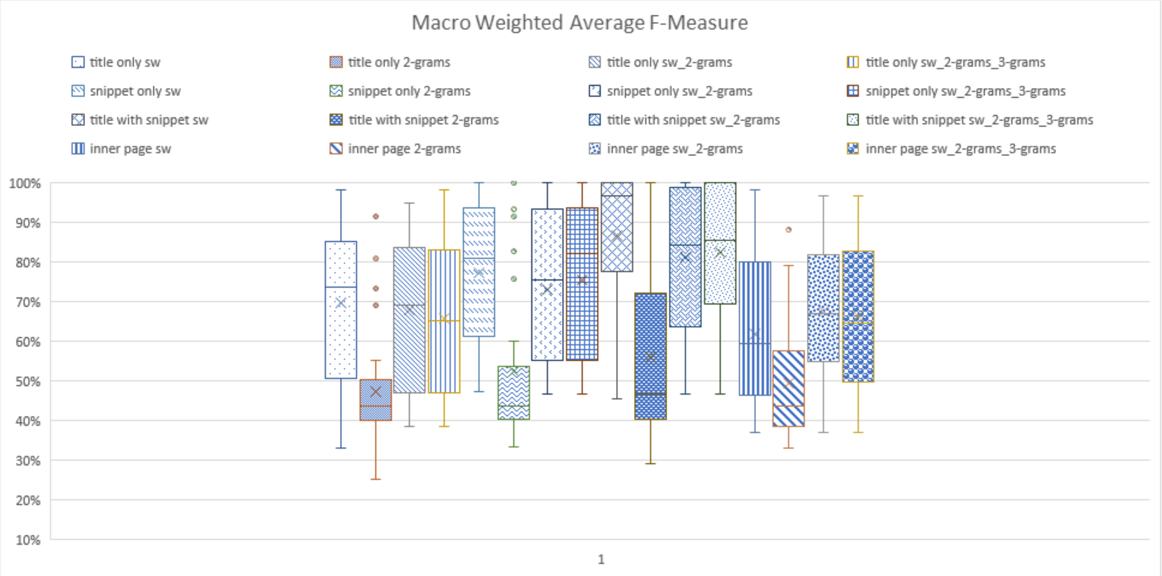


Figure 7.2: A boxplot diagram of F-measure when using MBHA benchmarks for Bing.

Table C.9 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online here[3]. The details of raw results of both disambiguation and evaluation can be found here[4]. Table 7.2 summarizes by showing the mean of each evaluation metric for all sources and spaces.

Table 7.2: The mean of each metric when using MBHA benchmarks for Bing.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	79%	73%	70%	84%	80%	77%	92%	88%	87%	77%	67%	62%
2-grams	72%	57%	47%	79%	61%	52%	78%	64%	56%	75%	58%	49%
sw__2-grams	80%	72%	68%	85%	76%	73%	89%	83%	81%	80%	71%	68%
sw__2-grams __3-grams	79%	70%	66%	85%	78%	76%	89%	84%	82%	79%	70%	66%

[3] https://goo.gl/3p6tA5
[4] https://goo.gl/HxzfwY

7.1.1.3 Mix of Google and Bing

As for mix of both engines, the mixing-based human-annotated benchmarks also contains datasets that are composed of search results from both engines. For example, *amazon* query (أمازون) has 30 search results from Google and 30 search results from Bing as *a river* plus 30 search results from Google and 30 search results from Bing as *a company*, having a total of 120 search items. Figure 7.3 shows the boxplot of the weighted F-measure.

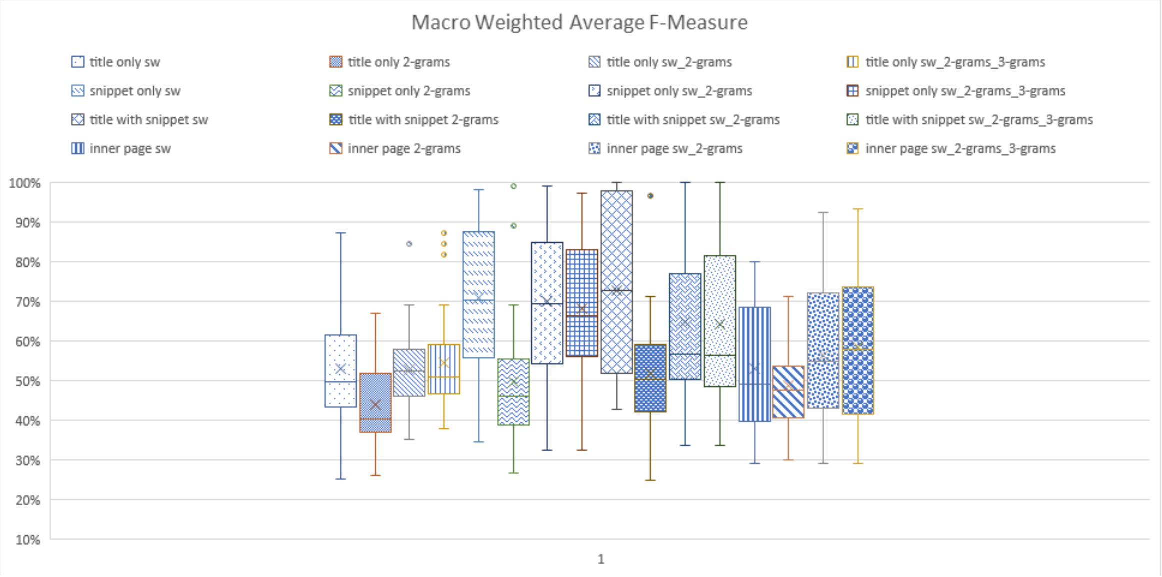


Figure 7.3: A boxplot diagram F-measure when using MBHA benchmarks for mixed data.

Table C.12 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[5]. The details of raw results of both disambiguation and evaluation can be found online[6]. Table 7.3 summarizes by showing the mean of each evaluation metric for all sources and spaces.

[5] `https://goo.gl/HxzfwY`

[6] `https://goo.gl/bointz`

Table 7.3: The mean of each metric when using MBHA benchmarks for mixed data.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	70%	60%	53%	79%	74%	71%	80%	76%	73%	69%	60%	53%
2-grams	68%	54%	44%	72%	58%	50%	70%	59%	52%	69%	57%	49%
sw__2-grams	70%	59%	53%	80%	73%	70%	76%	69%	65%	74%	63%	56%
sw__2-grams __3-grams	72%	61%	55%	79%	72%	68%	76%	69%	64%	76%	65%	59%

7.1.1.4 Discussion

The data of levels (i.e., conditions) are normally distributed. Based on the F-test (repeated measures ANOVA test) and adopting the macro weighted F-measure, the following items are concluded:

- There is a significant difference between the feature sources (i.e, **title**, **snippet**, **title-with-snippet**, and **inner page**) on clustering effectiveness (measured by F-measure). In other words, the effect of the feature sources on the effectiveness is significant.

- **Title-with snippet** is the best amongst the other sources. However, when mixing the Google and Bing results, **Snippet only** and then **Title-with snippet** are the best.

- There is a significant difference between the feature space representations (i.e, **sw**, **sw__2-grams**, **sw__2-grams**, and **sw__2-grams__3-grams**) on clustering effectiveness (measured by F-measure). In other words, the effect of the space representation on the effectiveness is significant.

- **Single words** dimensions, (**single words with 2-grams and 3-grams**), and (**single words with 2-grams**) dimensions are the best amongst the others.

- The **Source * Space** interaction effect on the quality of clusters (measured by F-measure) achieves a statistical significance.

- (**Title with snippet * sw**) is the best.

Therefore, the null hypothesis can be rejected, and the results above support the alternative hypothesis.

Notably, (**title with snippet * sw**), then (**title with snippet * sw__2-grams__3-grams**), and then (**title with snippet * sw__2-grams**) outperform the other combinations.

PostHoc tests support the claim that (**title with snippet * sw**) significantly differs from the other combinations.

In mix of both, we have small differences here as follows:

- **(Snippet only * sw)**, **(snippet only * sw__2-grams)**, and **(snippet only * sw__2-grams__3-grams)** outperform to some extent **(title with snippet * sw__2-grams__3-grams)** and **(title with snippet * sw__2-grams)**.
- The reason for such difference is mainly due to the difference in lengths of title and/or snippet between Google and Bing. The wordsToKeep calculations in the Design chapter shows that Google tends to have longer title and snippet than Bing. Thus, mixing length-inconsistent titles and snippets and likely different strategies of generating snippets per engine are potential causes for such small difference.

Table 7.4 summarizes the discussion above by showing the best value for each condition and for each engine, along with ANOVA *p* value. In this and other tables summarizing the best values, the cell having more than one value means that these values are the best and listed in order.

Table 7.4: The best value for each variable along with the *p* statistics value.

Variable/ Engine	**Source**	**Space**	**Source * Space**
Google	title w/ snippet, (1.51×10^{-10})	sw__2__3, (1.33×10^{-19}) sw sw__2	title w/snippet * sw, (0.0463) title w/ snippet * sw__2__3 title w/ snippet * sw__2
Bing	title w/ snippet, (1.11×10^{-06})	sw, (5.27×10^{-22}) sw__2 sw__2__3	title w/snippet * sw, (0.0138) title w/ snippet * sw__2__3 title w/ snippet * sw__2
Mix of both	snippet, (7.78×10^{-07}) title w/ snippet	sw, (1.22×10^{-13}) sw__2__3 sw__2	title w/snippet * sw, (0.0005) snippet * sw snippet * sw__2

7.1.1.5 Main Findings

By looking into the above results of the experiments, which were applied on the mixing-based human-annotated benchmarks, we have the following main findings:

1. The selection source of features is important and significantly affects the performance of search results disambiguation.
2. Building the dimensions of document vectors is important and significantly affects the disambiguation process.
3. Using **(title with snippet * sw)**, **(title with snippet * sw__2-grams__3-grams)**, or **(title with snippet * sw__2-grams)** gives the best performance (in terms of effectiveness, measured by F-measure). Moreover, and most importantly, both engines, Google and Bing, align very well with this conclusion.

4. Mixing results from Google and Bing could lead to a bit unexpected results because each engine generates different lengths of titles and snippets, and perhaps different mechanism of generating snippets from original Web page. However, results show the same pattern and the same competing combination of **(title with snippet * sw)**.

5. The results show increasing the data size improves the results for Google, Bing, and mix of Google and Bing, as shown in Figure 7.4. This figure shows how the effectiveness of disambiguation changes over different data sizes. The y-axis represents the macro F-measure. The x-axis represents the 16 levels of the two factors: feature sources and space representations (4 × 4). Each curve indicates a particular data size. As concluded in the previous point that Google plus Bing are worse than the results for each separately, increasing the data size improves the results though. This indicates that the variations in snippet generation for each engine needs more data to learn the two engines in a combined model.

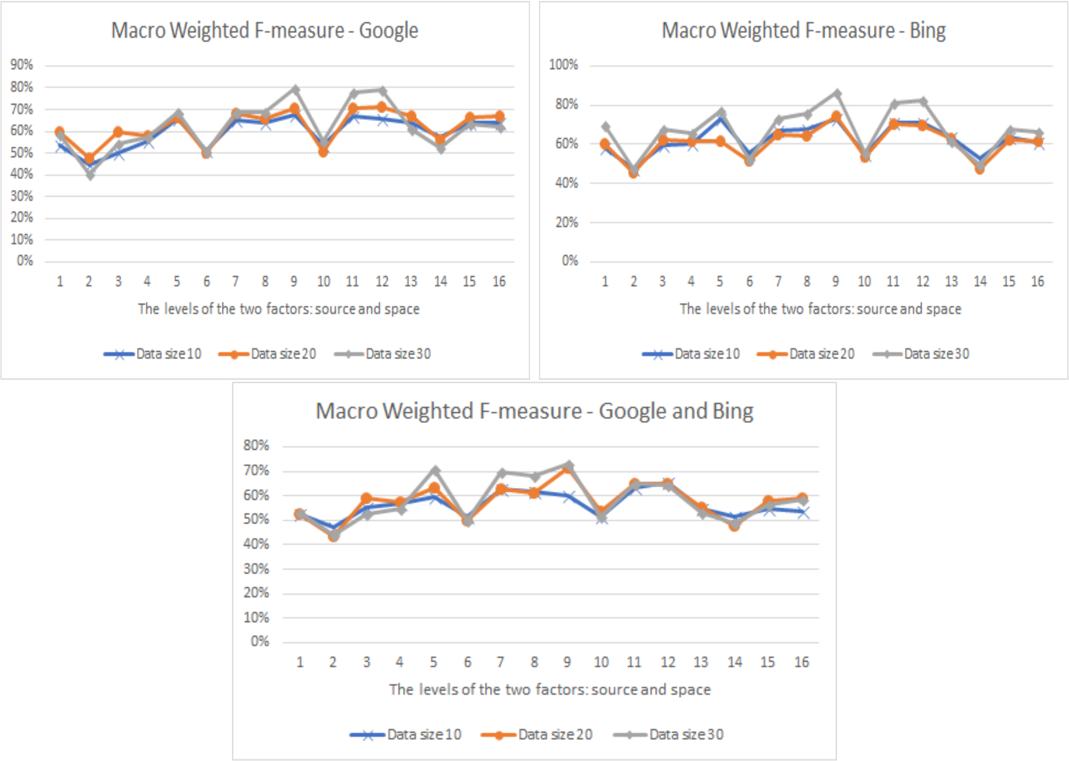


Figure 7.4: Macro weighted F-measure averaged across all queries over different data sizes.

7.1.2 Blind Relevance Feedback

As described in the Data Collection chapter, BRF-based benchmarks contain datasets that are pseudo-annotated. That is, assuming that the top 50% ranked documents are relevant to the sense of the query without manual check. The main goal is to see whether the pseudo relevance feedback supports the results in the previous experiments (i.e., human annotated).

7.1.2.1 Google

Figure 7.5 shows the boxplot of the weighted F-measure for Google search engine. Table 7.5 shows the mean value of F-measure for all conditions.

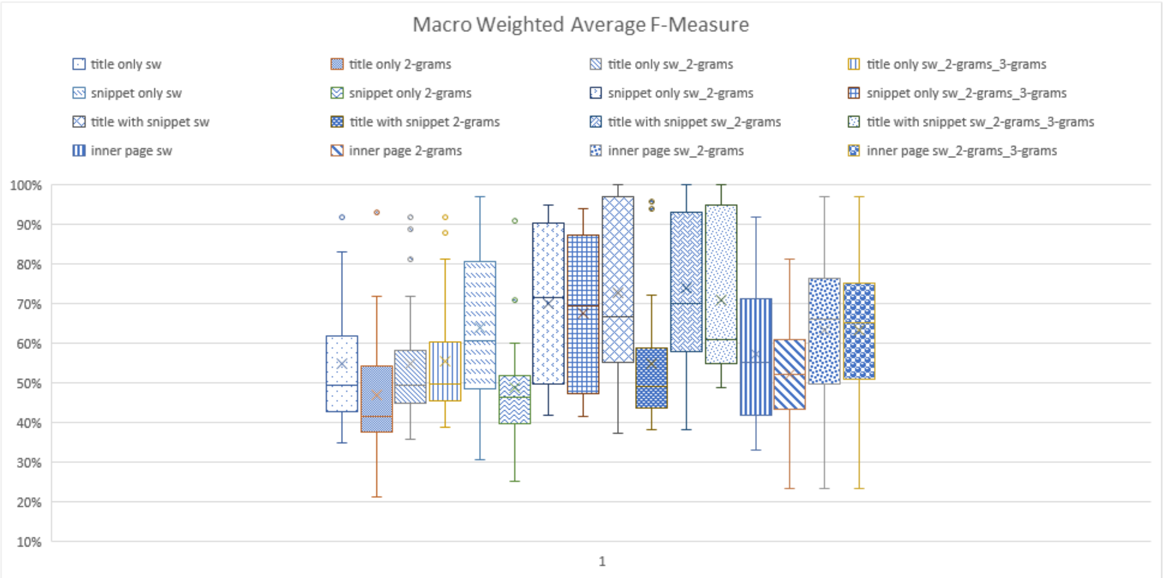


Figure 7.5: A boxplot diagram of F-measure when using BRF benchmarks for Google.

Table C.15 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[7]. The details of raw results of both disambiguation and evaluation can be found online[8]. Table 7.5 summarizes by showing the mean of each evaluation metric for all sources and spaces.

[7] https://goo.gl/Ew8h27

[8] https://goo.gl/TYURgF

Table 7.5: The mean of each metric when using BRF benchmarks for Google.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	71%	61%	55%	73%	68%	64%	80%	75%	73%	74%	64%	57%
2-grams	71%	56%	47%	73%	58%	49%	73%	61%	55%	72%	59%	52%
sw__2-grams	71%	61%	55%	78%	73%	70%	81%	76%	74%	74%	68%	63%
sw__2-grams __3-grams	73%	62%	55%	77%	71%	68%	79%	74%	71%	74%	68%	63%

7.1.2.2 Bing

As for Bing, Figure 7.6 shows the boxplot of the weighted F-measure. Table 7.6 shows the mean value of F-measure for all conditions.

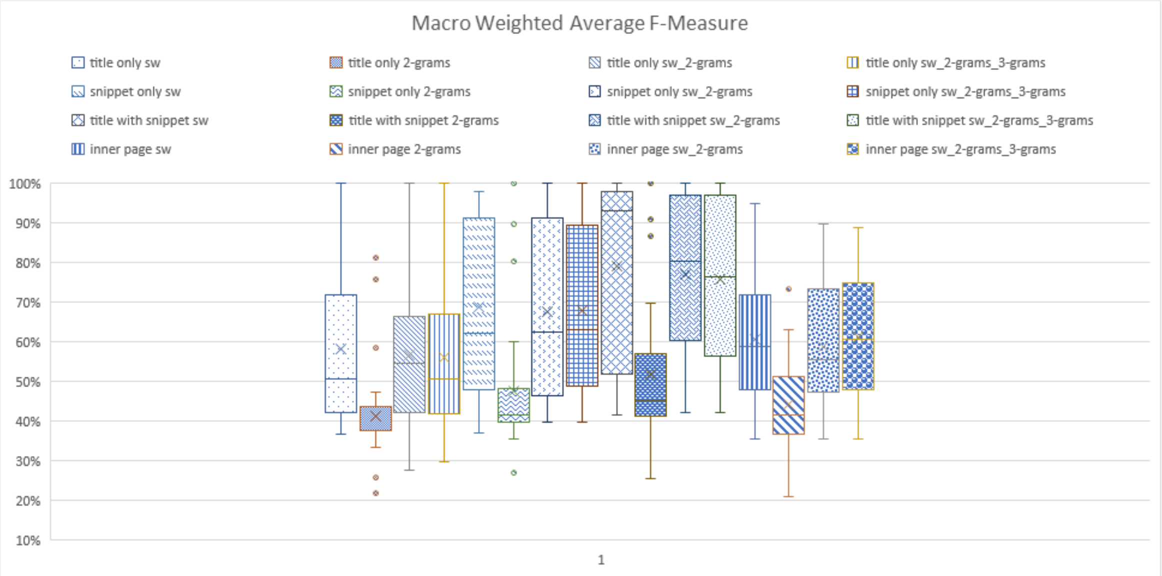


Figure 7.6: A boxplot diagram of F-measure when using BRF benchmarks for Bing.

Table C.18 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[9]. The details of raw results of both disambiguation and evaluation can be found online[10]. Table 7.6 summarizes by showing the mean of each evaluation metric for all sources and spaces.

[9] https://goo.gl/gNh4JS

[10] https://goo.gl/ehXmPy

Table 7.6: The mean of each metric when using BRF benchmarks for Bing.

Source/ Space	title			snippet			title w/ snippet			inner page		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	74%	65%	58%	78%	73%	69%	85%	82%	79%	72%	65%	61%
2-grams	70%	53%	41%	73%	58%	47%	74%	61%	52%	69%	55%	44%
sw__2-grams	72%	63%	57%	78%	72%	68%	85%	80%	77%	72%	64%	59%
sw__2-grams __3-grams	73%	63%	56%	80%	73%	68%	84%	79%	76%	74%	66%	61%

7.1.2.3 Mix of Google and Bing

The BRF-based benchmarks also contains datasets that are composed of search results from both engines. Figure 7.7 shows the boxplot of the weighted F-measure. Table 7.6 shows the mean value of F-measure for all conditions.

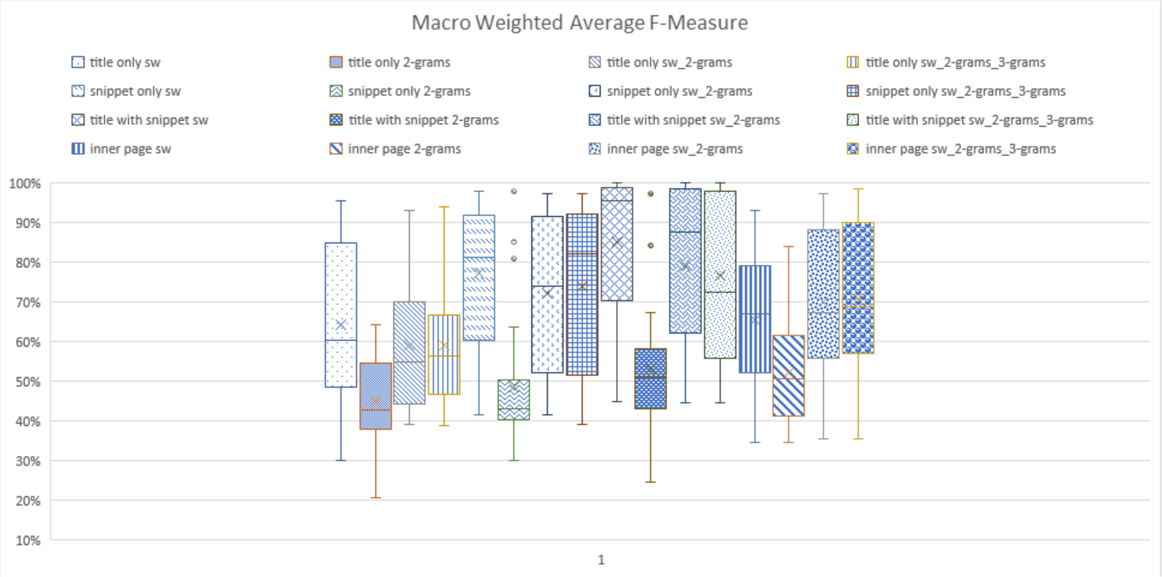


Figure 7.7: A boxplot diagram of F-measure when using BRF benchmarks for mixed data.

Table C.21 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[11]. The details of raw results of both disambiguation and evaluation can be found online[12]. Table 7.7 summarizes by showing the mean of each evaluation metric for all sources and spaces.

[11] https://goo.gl/fq3PKk
[12] https://goo.gl/DqgW7E

Table 7.7: The mean of each metric when using BRF benchmarks for mixed data.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	75%	69%	64%	85%	80%	77%	90%	87%	85%	77%	70%	65%
2-grams	69%	55%	45%	75%	58%	49%	72%	60%	53%	72%	59%	52%
sw__2-grams	72%	65%	59%	81%	76%	72%	86%	81%	79%	78%	73%	70%
sw__2-grams __3-grams	74%	65%	59%	82%	77%	74%	84%	79%	77%	79%	74%	71%

7.1.2.4 Discussion

Based on the above weighted F-measure charts, the following items are concluded:

- In Google, the majority of good results (i.e., have more quality) occur at **(title with snippet * sw__2-grams)**, **(title with snippet * sw)**, and **(title with snippet * sw__2-grams__3-grams)**.
- In Bing, the majority of good results occur at **(title with snippet * sw)**, **(title with snippet * sw__2-grams)**, and **(title with snippet * sw__2-grams__3-grams)**.
- In mix of both, the majority of good results occur at **(title with snippet * sw)**, **(title with snippet * sw__2-grams)**, **(snippet only * sw)** and **(title with snippet * sw__2-grams__3-grams)**.
- The pattern of effectiveness for all engines is very similar to the pattern of human-annotated mixing-based benchmarks.

Table 7.8 summarizes the discussion above by showing the best value for each condition and for each engine, along with ANOVA p value.

Table 7.8: The best value for each variable along with the p statistics value.

Variable/ Engine	**Source**	**Space**	**Source * Space**
Google	title w/ snippet, (1.69×10^{-06})	sw_2, (1.14×10^{-13}) sw_2_3 sw	title w/ snippet * sw_2, (0.0222) title w/snippet * sw title w/ snippet * sw_2_3
Bing	title w/ snippet, (1.36×10^-06)	sw, (3.30×10^{-21}) sw_2_3 sw_2	title w/snippet * sw, (0.1623) title w/ snippet * sw_2 title w/ snippet * sw_2_3
Mix of both	title w/ snippet, (6.17×10^{-07})	sw, (1.21×10^{-20}) sw_2 sw_2_3	title w/snippet * sw, (0.0003) title w/ snippet * sw_2 title w/ snippet * sw_2_3

7.1.2.5 Main Findings

By looking into the above results of the experiments, which were applied on the BRF-based benchmarks, we have the following main findings:

1. Interestingly, the box plot pattern of pseudo relevance feedback results is very close to the one of manually annotated results. Therefore, it supports the results and conclusions of the previous experiments (i.e., human annotated datasets).

2. Even though not every single result of the clear query is relevant to that query, pseudo relevance feedback succeeds and gives the same conclusions, and this is due to the assumption that the majority of results should be relevant to the clear query.

3. This considerably shows how useful pseudo relevance feedback concept can be in search results disambiguation. Particularly, this gives an initial hint that this new way of labeling by using blind relevance feedback can be relied on when labeling datasets (e.g., training datasets). Therefore, this eliminates the human efforts and saves time.

7.1.3 Plain Benchmarks

In this type of experiments, the disambiguation process was directly applied on search results of subset of ambiguous queries[13], evaluated against human-annotated datasets with predefined meanings.

7.1.3.1 Google

Figure 7.8 shows the boxplot of the weighted f-measure for Google search engine. Table 7.9 shows the mean value of F-measure for all conditions.

[13] The subsets can be found in Appendix C.3.7 for Google and Appendix C.3.7 for Bing.

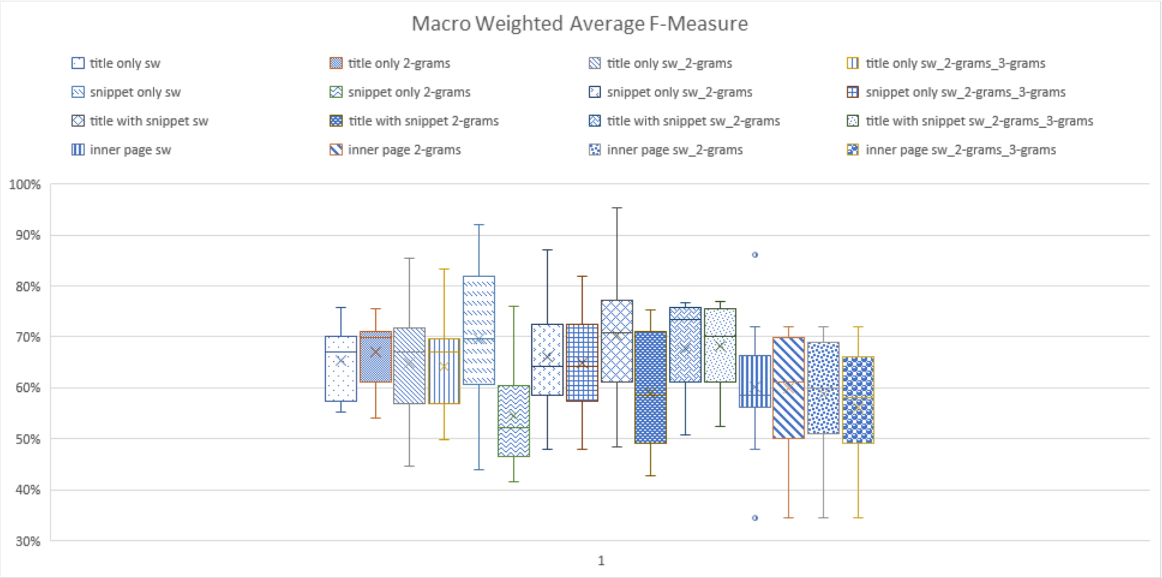


Figure 7.8: A boxplot diagram of F-measure when using plain benchmarks for Google.

Table C.24 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[14]. The details of raw results of both disambiguation and evaluation can be found online[15]. Table 7.9 summarizes by showing the mean of each evaluation metric for all sources and spaces.

Table 7.9: The mean of each metric when using plain benchmarks for Google.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	76%	65%	65%	84%	66%	70%	79%	69%	70%	70%	64%	60%
2-grams	79%	66%	67%	71%	52%	55%	74%	57%	59%	62%	65%	60%
sw__2-grams	80%	61%	65%	82%	62%	66%	77%	66%	68%	63%	65%	59%
sw__2-grams __3-grams	76%	61%	64%	86%	58%	65%	80%	66%	68%	60%	64%	56%

7.1.3.2 Bing

As for Bing, Figure 7.9 shows the boxplot of the weighted f-measure. Table 7.10 shows the mean value of F-measure for all conditions

[14] https://goo.gl/c9fiWF

[15] https://goo.gl/T7eVNt

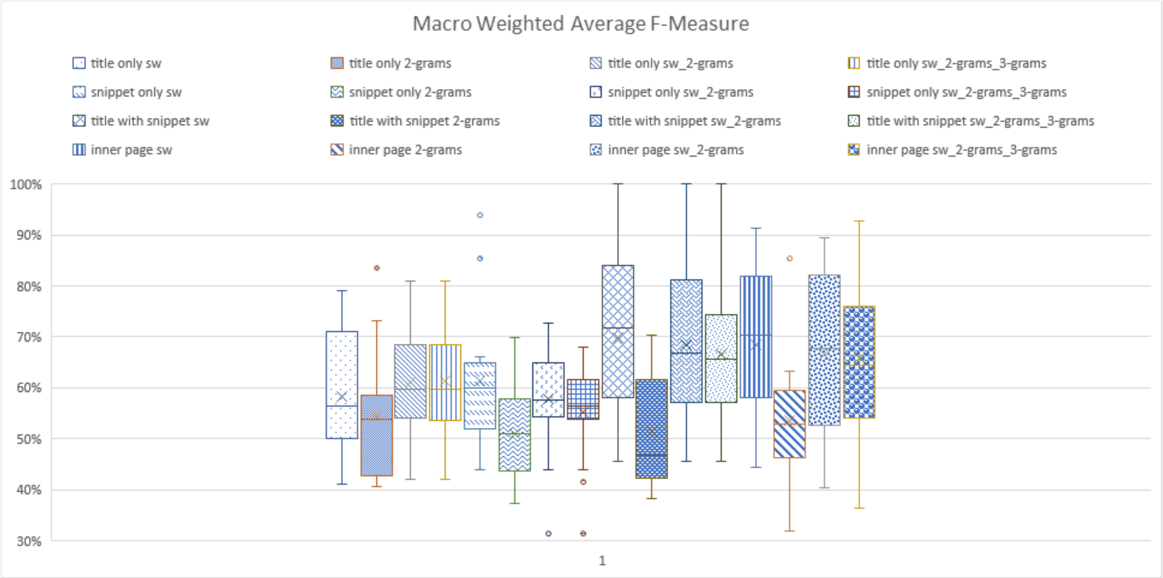


Figure 7.9: A boxplot diagram of F-measure when using plain benchmarks for Bing.

Table C.27 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[16]. The details of raw results of both disambiguation and evaluation can be found online[17]. Table 7.10 summarizes by showing the mean of each evaluation metric for all sources and spaces.

Table 7.10: The mean of each metric when using plain benchmarks for Bing.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	88%	48%	58%	90%	51%	61%	94%	60%	70%	72%	69%	68%
2-grams	79%	54%	55%	77%	50%	51%	81%	47%	51%	63%	59%	54%
sw__2-grams	88%	52%	61%	88%	48%	58%	94%	59%	69%	73%	67%	66%
sw__2-grams __3-grams	88%	53%	61%	87%	45%	55%	94%	57%	67%	73%	66%	66%

7.1.3.3 Discussion

Based on the F-test (repeated measures ANOVA test) and adopting the weighted F-measure, the following items are concluded:

- In Google, the majority of good results (i.e., have more quality) occur at **(title**

[16] https://goo.gl/SGgTcY
[17] https://goo.gl/k3dM59

with snippet * sw), **(snippet only * sw)**, **(title with snippet * sw__2-grams__3-grams)**, and **(title with snippet * sw__2-grams)**.

- In Bing, the majority of good results occur at **(title with snippet * sw)**, **(title with snippet * sw__2-grams)**, **(inner page * sw)**, and **(title with snippet * sw__2-grams__3-grams)**.
- In Bing, **inner page * sw** shows up for the first time among the combinations that have good quality compared to others. This happened because the number of inner pages fetched for Bing is more than Google[18].
- The **Source * Space** interaction effects on the quality of clusters (measured by F-measure) achieves a statistical significance.
- In Google, **(Title with snippet * sw)** is the best (with a weighted F-measure of 70% mean and 71% median).
- In Bing, **(Title with snippet * sw)** is the best (with a weighted F-measure of 70% mean and 72% median).

Therefore, the null hypothesis can be rejected, and the results above support the alternative hypothesis. Moreover, PostHoc tests show that **(title with snippet * sw)** significantly differs from the other combinations.

Table 7.11 summarizes the discussion above by showing the best value for each condition and for each engine, along with ANOVA *p* value.

Table 7.11: The best value for each variable along with the *p* statistics value.

Variable/ Engine	**Source**	**Space**	**Source * Space**
Google	title w/ snippet, (0.1106)	sw, (0.0088)	title w/ snippet * sw, (0.0454) snippet * sw title w/ snippet * sw__2__3
Bing	title w/ snippet, (0.0465)	sw, (1.24×10^{-07})	title w/ snippet * sw, (0.0243) title w/ snippet * sw__2 inner__page * sw title w/ snippet * sw__2__3

7.1.3.4 Main Findings

By looking into the above results of the experiments, which were applied on the plain benchmarks to see whether these benchmarks supports the claims concluded by mixing-based human-annotated benchmarks as well, we have the following main findings:

[18] For inner pages, not all sites allow you to crawl their web pages. However, we were capable of fetching inner page for most of search results. For Bing, we were able to crawl 2822 inner pages out of 3000 search results (labeled and unlabeled). For Google, we were able to crawl 1033 out of 1100 (labeled and unlabeled).

1. The results of experiments for Google and Bing still agree with the previous experiments on that the selection of source and how we build the dimensions considerably affect the performance of disambiguation.

2. Both engines agree on the best combination that gives the best performance: **(title with snippet * sw)**, **(title with snippet * sw__2-grams__3-grams)**, or **(title with snippet * sw__2-grams)**, even though Google shows a competitive combination: **(snippet only * sw)**; this makes total sense as snippets of Google tend to be longer than snippets of Bing.

3. Dynamically determining the number of clusters (i.e., the number of senses) based on a cluster validity criterion gives a reasonable performance. For the best combination, Google and Bing give 70% mean and 71% and 72% median, respectively. Now, are we getting better or worse results when we apply the proposed solution (i.e., SAUL)? This is discussed next.

7.2 The Proposed Approach: A Supervised Approach to Unsupervised Learning (SAUL)

This type of experiments augments supervised models into unsupervised learning by taking advantage of the models that are built using mixing-based human-annotated benchmarks or blind relevance feedback.

7.2.1 Models Built using Mixing-based Human-annotated Benchmarks

For each engine, a clustering model based on mixing-based human-annotated benchmarks is built. This model is then used to cluster new unseen search results of ambiguous queries.

7.2.1.1 Google

Figure 7.10 depicts the boxplot of the weighted F-measure for Google search engine. Table 7.12 shows the mean value of F-measure for all conditions.

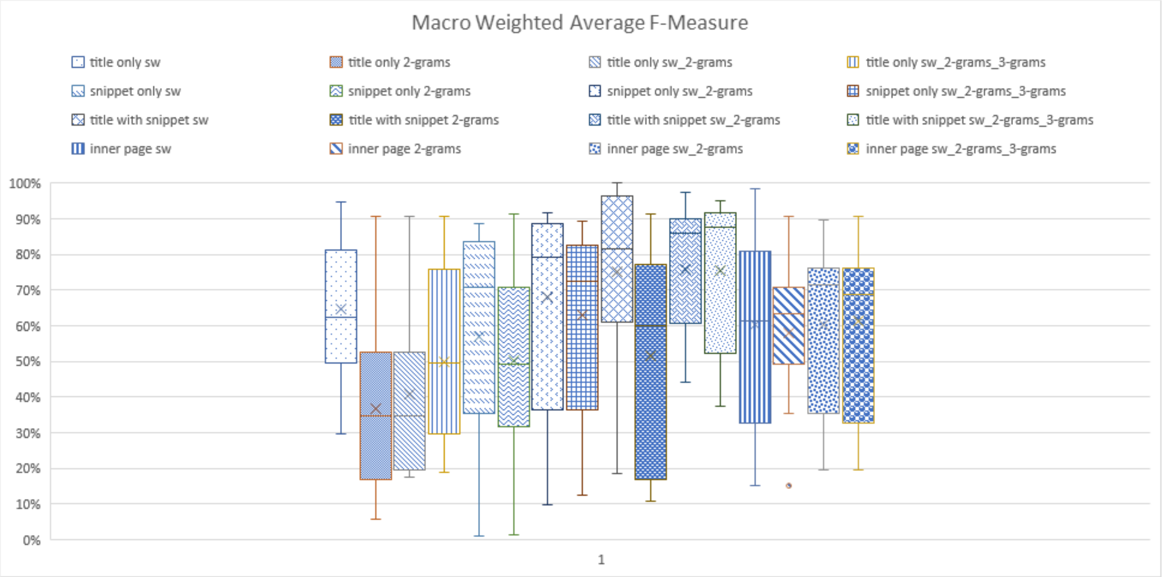


Figure 7.10: A boxplot diagram of F-measure when using MBHA supervised approach/Google.

Table C.30 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[19]. The details of raw results of both disambiguation and evaluation can be found online[20]. Table 7.12 summarizes by showing the mean of each evaluation metric for all sources and spaces.

Table 7.12: The mean of each metric when using MBHA supervised approach for Google.

Source/ Space	title			snippet			title w/ snippet			inner page		
	P	R	F	P	R	F	P	R	F	P	R	F
sw	72%	68%	65%	67%	61%	57%	80%	76%	75%	67%	65%	60%
2-grams	38%	49%	37%	66%	59%	50%	61%	57%	51%	74%	62%	58%
sw__2-grams	60%	48%	41%	77%	72%	68%	84%	76%	76%	78%	64%	60%
sw__2-grams __3-grams	64%	55%	50%	74%	65%	63%	82%	77%	76%	73%	65%	61%

7.2.1.2 Bing

As for Bing search engine, Figure 7.11 depicts the boxplot of the weighted F-measure. Table 7.13 shows the mean value of F-measure for all conditions.

[19] https://goo.gl/pZhK6G

[20] https://goo.gl/Rc7n34

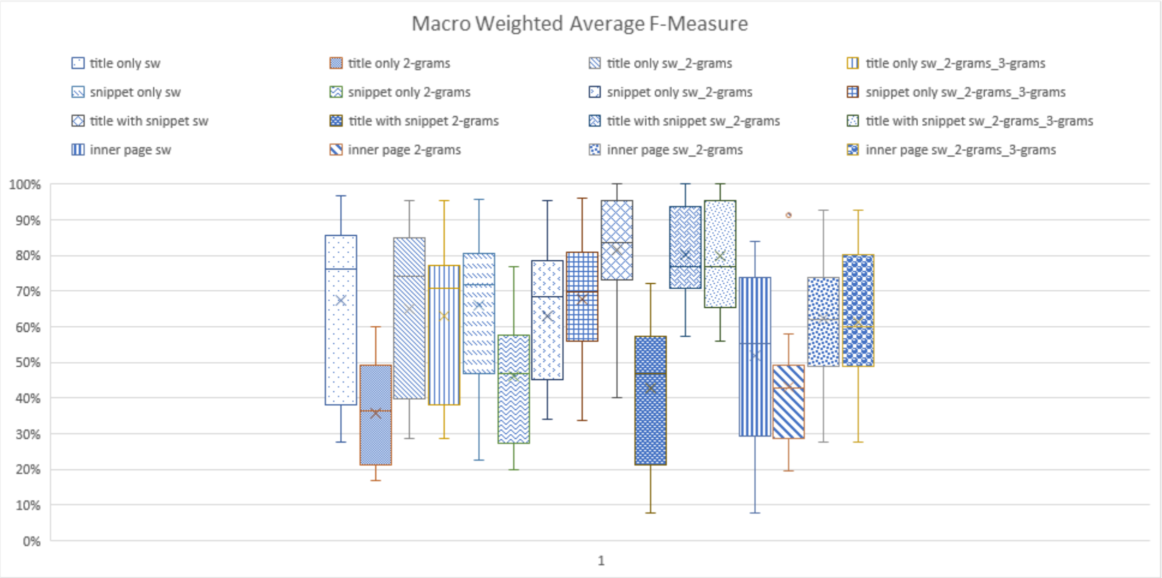


Figure 7.11: A boxplot diagram of F-measure when using MBHA supervised approach/Bing.

Table C.33 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[21]. The details of raw results of both disambiguation and evaluation can be found online[22]. Table 7.13 summarizes by showing the mean of each evaluation metric for all sources and spaces.

Table 7.13: The mean of each metric when using MBHA supervised approach for Bing.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	79%	70%	67%	77%	70%	66%	89%	83%	82%	64%	58%	52%
2-grams	65%	47%	36%	71%	56%	46%	68%	52%	43%	72%	50%	43%
sw__2-grams	83%	68%	65%	79%	68%	63%	87%	82%	80%	75%	65%	62%
sw__2-grams __3-grams	78%	66%	63%	80%	72%	68%	88%	82%	80%	73%	65%	61%

7.2.1.3 Discussion

Based on the weighted F-measure values for both Google and Bing, the following items are concluded:

[21] `https://goo.gl/3rCtDR`

[22] `https://goo.gl/YbQ1DN`

- In Google, the majority of good results (i.e., have more quality) occur at **(title with snippet * sw__2-grams)**, **(snippet * sw__2-grams__3-grams)**, and **(title with snippet * sw)**.
- In Bing, the majority of good results (i.e., have more quality) occur at **(title with snippet * sw)**, **(title with snippet * sw__2-grams)**, and **(snippet * sw__2-grams__3-grams)**.
- The best combinations in Google and Bing are found to be the same. Table 7.14 shows the combinations along with the mean and median values of F-measure.

Table 7.14: The mean and median of F-measure for best combinations in Google and Bing.

Engine/ Combination	**Google**		**Bing**	
	Mean	**Median**	**Mean**	**Median**
title w/ snippet * sw	75%	81%	82%	84%
title w/ snippet * sw__2	76%	86%	80%	77%
title w/ snippet * sw__2__3	76%	88%	80%	77%

7.2.2 Models Built using Mixing-based Blind Relevance Feedback

Instead of using mixing-based human-annotated approach of building the clustering models, these models are built using the datasets that are based on the concept of blind relevance feedback.

7.2.2.1 Google

Figure 7.12 depicts the boxplot of the weighted F-measure for Google search engine. Table 7.15 shows the mean value of F-measure for all conditions.

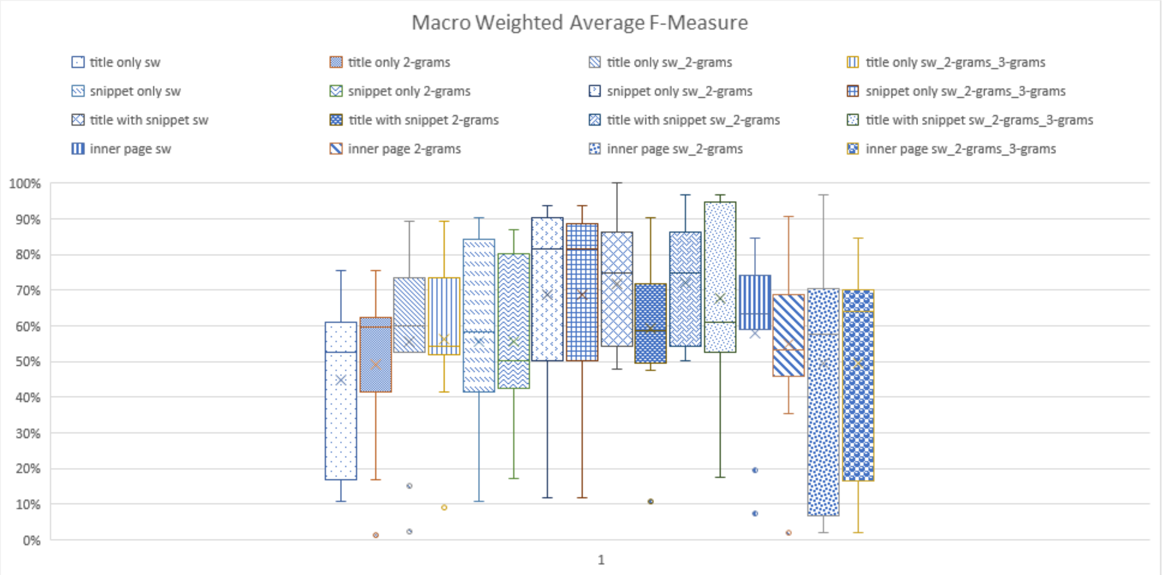


Figure 7.12: A boxplot diagram of F-measure when using BRF supervised approach/Google.

Table C.36 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[23]. The details of raw results of both disambiguation and evaluation can be found online[24]. Table 7.15 summarizes by showing the mean of each evaluation metric for all sources and spaces.

[23] https://goo.gl/66GFj8

[24] https://goo.gl/xRvASu

Table 7.15: The mean of each metric when using BRF supervised approach for Google.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	56%	50%	45%	60%	61%	56%	77%	74%	72%	70%	62%	58%
2-grams	52%	59%	49%	70%	64%	56%	64%	64%	59%	61%	63%	55%
sw__2-grams	69%	60%	56%	74%	72%	69%	78%	75%	72%	56%	52%	49%
sw__2-grams __3-grams	67%	63%	56%	73%	71%	69%	74%	70%	68%	56%	54%	50%

7.2.2.2 Bing

As for Bing search engine, Figure 7.13 depicts the boxplot of the weighted F-measure. Table 7.16 shows the mean value of F-measure for all conditions.

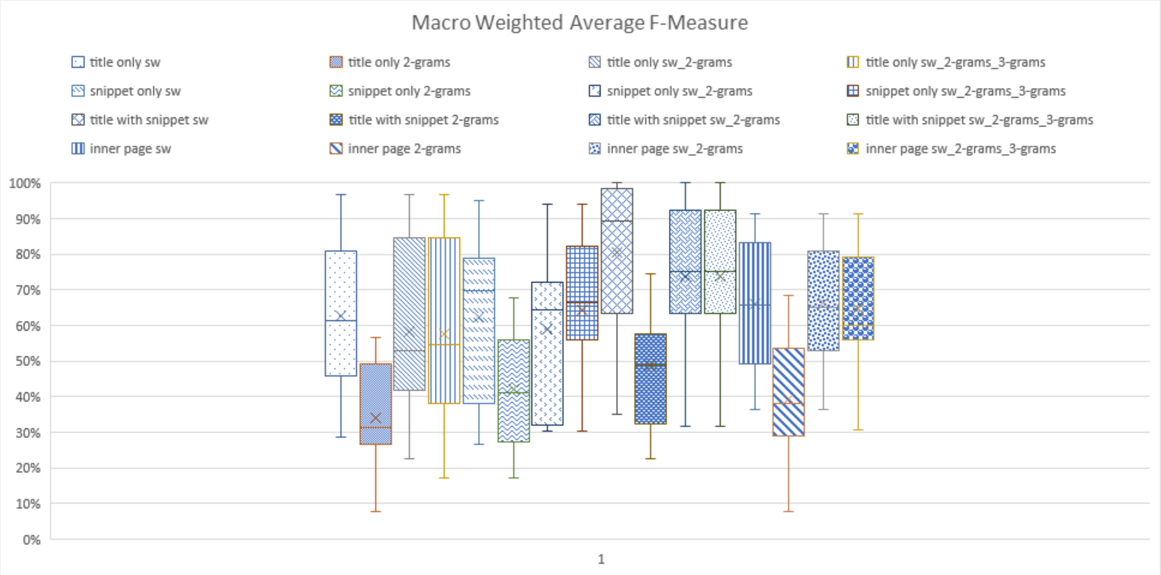


Figure 7.13: A boxplot diagram of F-measure when using BRF supervised approach/Bing.

Table C.39 in Appendix C.3 shows the F-measure values for all ambiguous queries. They can also be found online[25]. The details of raw results of both disambiguation and evaluation can be found online[26]. Table 7.16 summarizes by showing the mean of each evaluation metric for all sources and spaces.

[25] `https://goo.gl/YZgcmf`

[26] `https://goo.gl/SqYCHd`

Table 7.16: The mean of each metric when using BRF supervised approach for Bing.

Source/ Space	**title**			**snippet**			**title w/ snippet**			**inner page**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
sw	77%	67%	63%	77%	66%	62%	86%	82%	80%	75%	69%	66%
2-grams	56%	47%	34%	64%	53%	42%	71%	58%	49%	64%	51%	39%
sw__2-grams	76%	64%	58%	75%	64%	59%	85%	76%	74%	74%	70%	66%
sw__2-grams__3-grams	70%	64%	58%	80%	68%	64%	85%	77%	74%	74%	67%	65%

7.2.2.3 Discussion

Based on the weighted F-measure values for both Google and Bing, the following items are concluded:

- In both Google and Bing, the majority of good results (i.e., have more quality) occur at **(title with snippet * sw)**, **(title with snippet * sw__2-grams)**, and **(title with snippet * sw__2-grams__3-grams)**.
- The pattern of effectiveness is very similar to the pattern of human-annotated mixing-based benchmarks.
- The best combination found for Google, **(title with snippet * sw)**, achieves 72% mean and 75% median.
- The best combination found for Bing, **(title with snippet * sw)**, achieves 80% mean and 89% median.

7.2.3 Main Findings

By looking into the results of the proposed approach experiments, we have the following main findings:

1. The results of experiments for Google and Bing still agree with the previous experiments on that the *selection of source* and *how we build the dimensions* considerably affect the performance of disambiguation.
2. Both engines agree on the best combination that gives the best performance: **(title with snippet * sw)**, **(title with snippet * sw__2-grams__3-grams)**, or **(title with snippet * sw__2-grams)**.
3. Leveraging the models of clear queries for building supervised clustering models to disambiguate search results of ambiguous queries gives very good and interesting results in terms of effectiveness (measured by F-measure).

4. By comparing Google and Bing in the traditional approach with the proposed approach (i.e., SAUL approach) and by taking the best combination of source and space in each, the proposed approach (built on mixing-based human-annotated datasets) outperforms the traditional one by an *increase* of **8%** and **18%** in mean and **15%** and **17%** median, for both Google and Bing, respectively. Moreover, the proposed approach (built on blind relevance feedback datasets) outperforms the traditional one by an *increase* of **3%** and **15%** in mean and **6%** and **24%** in median, for both Google and Bing, respectively. Figure 7.14 visualizes these percentages and compares between the different approaches. Additionally, Table 7.17 summarizes the percentages for all approaches.

5. Most of previous experiments, with different approaches, agrees on building the clusterer for search results disambiguation using *title with snippet* as a features source and *sw* only or *sw_2-grams* or *sw_2-grams_3-grams* as a features representation gives the best effectiveness (measured by F-measure).

6. All above experiments show that using *phrases only* (i.e., 2-grams) as features *hurts* the clustering process in disambiguation regardless the source you take from. This is proven by the bad performance in terms of effectiveness (measured by F-measure) resulted by the 2-grams.

7. In all above experiments and with current settings, there are many reasons why innerpages show a bad performance compared to title with snippet. One reason is that web pages contain much noisy data (i.e., text not related to the topic). Moreover, another reason not directly related to the nature of web pages but to how web developers create them. Many websites hide some/all page contents behind JavaScript to protect it from crawling or stealing the content, which considerably affects the preprocessing stage; thus, the effectiveness of the disambiguator.

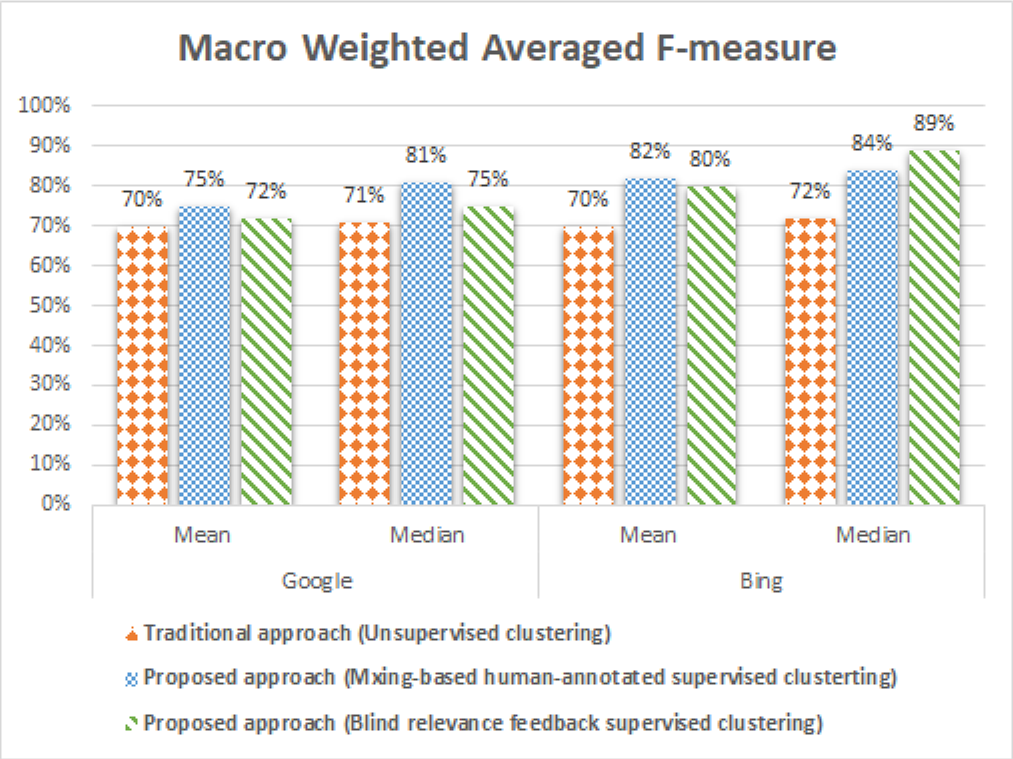


Figure 7.14: A comparison chart between the three approaches based on macro F-measure.

Table 7.17: The mean and median of macro F-measure for different approaches.

Engine/Approach	Traditional		Proposed (HA)		Proposed (BRF)	
	Mean	Median	Mean	Median	Mean	Median
Google	70%	71%	75%	81%	72%	75%
Bing	70%	72%	82%	84%	80%	89%

7.2.4 Comparison Using A Query Example

While the previous sections depict the big picture of results, this subsection compares between the traditional approach (i.e., baseline) and the proposed approaches: MBHA supervised clustering and BRF supervised clustering by showing specific query example. It summarizes the clustering results for the Bing search results of عمان query as an example. This query could refer to Oman (i.e., the country) or Amman (i.e., the city). Therefore, sultanate and capital are the meanings of that ambiguous query.

In the traditional approach, the resulting clusters (i.e., groups) are 6 clusters. The search results are distributed across the clusters as shown in Table 7.18. Each number in this table and the following tables is the number of search results that actually belongs to the meaning.

Table 7.18: The clusters of search results of عمان query using the baseline approach.

Assigned to cluster –>	C0	C1	C2	C3	C4	C5	C6
Sultanate	12	0	20	1	4	2	4
Capital	7	10	0	1	2	0	1

In the MBHA supervised clustering approach, the resulting clusters are 2 clusters. The search results are distributed across the clusters as shown in Table 7.19. This shows that this approach is better than the baseline approach.

Table 7.19: The clusters of search results of عمان query using the MBHA supervised approach.

Assigned to cluster –>	C0	C1
Sultanate	1	42
Capital	21	0

In the BRF supervised clustering approach, the resulting clusters are 2 clusters as well. The search results are distributed across the clusters as shown in Table 7.20. This shows that this approach is also better than the baseline approach.

Table 7.20: The clusters of search results of عمان query using the BRF supervised approach.

Assigned to cluster –>	C0	C1
Sultanate	38	5
Capital	0	21

Chapter 8

Conclusion and Outlook

This chapter draws the conclusion of the whole thesis and shows some work that can be done in future.

8.1 Conclusion

Using search results clustering as an approach to solve search results disambiguation problem is a good choice because inducing clusters from search results gives improved insights about query senses or meanings. This is very useful in helping users identify their information need easily and faster. Nevertheless, dealing with search results clustering is very challenging because for example the resulting clusters should be of high quality (i.e., the degree to which search results in a cluster belong to same meaning) and the cluster labels must be understandable.

This thesis helps mitigate some research gaps in search results clustering with focus on Arabic language. First, there has been no publicly available benchmarks for Arabic language, this thesis introduces such benchmarks (called AMBIGArabic) that can be used in any experiments involving search results clustering or even more generally search results disambiguation. In addition to human labeling that is usually performed to build such benchmarks, this study proposes two new labeling approaches that can be used along with the manual labeling: *mixing-based labeling* and *intersecting-based labeling.* This kind of benchmarks is very helpful for researchers who want, for example, to study and compare different methods or algorithms for search results clustering.

Second, as there has been no clear proof that shows what source of features one should take from or how one can represent such features in vector space model, and though different studies propose different ways of text feature representation such as n-grams or lexical affinities, or single words along with ordered sequences, this thesis shows that deciding which *feature source* to select (i.e., title only, snippet only, title with snippet, or inner page) and/or which *vector space representation* to use (i.e., single words (sw), 2-grams, sw with 2-grams, and sw with 2-grams and 3-grams), *statistically significantly* matters. In other words, the findings show that working with single words alone or even single words along with n-grams (like 2-grams and/or 3-grams) is the best.

Moreover, this thesis shows that extracting those features from title and snippet gives the best effectiveness in terms of F-measure. This thesis recommends adopting them (i.e., title with snippet as a feature source and single words or even single words along with 2-grams and/or 3-grams as a feature representation) for Arabic language in the preprocessing stage.

To conduct search result clustering experiments, there has been no frameworks that aid in implementing and running such experiments. This thesis offers a new kind of extensible framework, called Spread, that is primarily built upon Weka framework. In particular, this framework has the ability to perform SRC experiments and fetch search results from engines like Google, Bing, and Yahoo. In addition, this framework offers a human assessment interface, that can be used to label search results using different strategies like *yes/no annotation* or *multi-meanings choices*.

Third, this thesis proposes a new way for dealing with search results clustering. A popular traditional approach (which is considered as a baseline approach) of using SRC-based methods is to cluster the search results using K-means with a dynamic determination of K. However, despite the fact that they dynamically cluster search results, most of their methods have poor label indicators for clusters generated, and consequently generating such labels is not an easy task. Instead, the proposed idea is based primarily on augmenting a supervised approach leveraging training datasets of clear queries into clustering, which is an unsupervised learning. This process involves building a supervised clustering model for each ambiguous query, then this model is used to cluster/classify new unseen search results. Knowing that collecting clear queries and labeling them by experts is very time consuming, this thesis also shows how blind relevance feedback can be very useful when using it to build training datasets instead. When comparing this proposed approach (called SAUL) to the traditional clustering approach, the proposed approach outperforms the traditional approach by 8% and 18%, when using human labeled training datasets, and by 3% and 15%, when using blind relevance feedback datasets, for Google and Bing, respectively. Additionally, the cluster label generation problem becomes easy since each generated cluster will be labeled by the defined meaning, which is primarily used in the clear query formulation.

As a real use case, the proposed approach can be used as follows: a fully working disambiguation system can be built by forming clear queries of words/phrases by getting their meanings/senses from a lexical database or a Web taxonomy-such as ODP, WordNet, and WDP, then fetching search results for the clear queries, and then building a supervised model per word/phrase based on those results and using the blind relevance feedback. Afterwards, when a word/phrase is queried, their search results will be input to its supervised clustering model and reported accordingly. To prove the feasibility of this use case, this thesis introduces a fully working API that is based on the SAUL approach along with WDP as a discovery source for the senses. This API is a main part of the Spread framework and is demonstrated in Chapter 6.

As usual, no work without limitations. The scope of the experimental research of this study covered two factors only: feature sources and feature space representations. Other important factors having high impact on effectiveness can be studied such as how many terms/features should be kept in vector space model. In addition, the new proposed approach for search results clustering is based on knowledge repositories or web taxonomies such as ODP, WordNET, and WDP to collect the ambiguous queries and their meanings/senses. This adds overhead to the process even though such models are often built offline. The new proposed approach also needs to fetch search results for all formed clear queries after collecting meanings/senses but this often happens offline as well. Additionally, clustering search results based on predefined senses from a knowledge repository makes the disambiguation of the search results coupled with those senses only, meaning that there might be search results of the ambiguous query that do not belong to any of the senses returned from the knowledge repository.

8.2 Outlook

Future work that can be developed include:

1. Studying other important factors like how many words to keep in vectorization process in search results clustering that could significantly improve quality of clustering.

2. Working on the idea of increasing the weight of neighbor terms that are positionally close to query terms and investigating whether this improves the clustering performance significantly.

3. Working on the idea of selecting the initial centroids of K-means algorithm (i.e., initial seeds) by having the top search result of each clear query, and then doing the traditional clustering. This might speed up the convergence of the K-means algorithm.

4. When combining results from both engines, it is worth to try to normalize term frequency by the length of snippet and see whether the results improve.

5. Implementing a scalable architecture of the proposed approach of search results clustering that can scale to thousands of ambiguous queries without affecting the efficiency.

6. Extending the Spread framework to support different kinds of popular clustering algorithms that can be used when benchmarking.

7. Working on the idea to build ensemble clusterer that consists of clusterer for title, clusterer for snippet, and clusterer for inner page, then the final decision will be based on weighted function that takes results for all those clusterers as input parameters.

8. Working on the idea of using *word2vec* which is a two-layer neural network that processes text instead and figuring out how this can be leveraged in search results clustering.

Bibliography

[1] W Bruce Croft, Donald Metzler, and Trevor Strohmann. *Search engines: Information Retrieval in Practice*. Pearson Education, 2015.

[2] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2006.

[3] Daniele Vannella, Tiziano Flati, and Roberto Navigli. Wosit: A word sense induction toolkit for search result clustering and diversification. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 67–72, 2014.

[4] Ruihua Song, Zhenxiao Luo, Ji-Rong Wen, Yong Yu, and Hsiao-Wuen Hon. Identifying ambiguous queries in web search. In *Proceedings of the 16th international conference on World Wide Web*, pages 1169–1170. ACM, 2007.

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

[6] Weimao Ke, Cassidy R Sugimoto, and Javed Mostafa. Dynamicity vs. effectiveness: studying online clustering for scatter/gather. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM, 2009.

[7] Chi Lang Ngo and Hung Son Nguyen. A method of web search result clustering based on rough sets. In *Proceedings of The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 673–679. IEEE, 2005.

[8] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54. ACM, 1998.

[9] Oren Zamir and Oren Etzioni. Grouper: a dynamic clustering interface to web search results. *Computer Networks*, 31(11):1361–1374, 1999.

[10] Stanisław Osiński, Jerzy Stefanowski, and Dawid Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Proceedings of Intelligent information processing and web mining*, pages 359–368. Springer, 2004.

[11] Paul N Bennett and Nam Nguyen. Refined experts: improving classification in large taxonomies. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18. ACM, 2009.

[12] Hao Ma, Michael R Lyu, and Irwin King. Diversifying query suggestion results. In *AAAI*, volume 10, 2010.

[13] Antonio Di Marco and Roberto Navigli. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754, 2013.

[14] Jiyang Chen, Osmar R Zaïane, and Randy Goebel. An unsupervised approach to cluster web search results based on word sense communities. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 725–729. IEEE Computer Society, 2008.

[15] Zhi Huang, Zhendong Niu, Donglei Liu, Wenjuan Niu, and Wei Wang. A novel method for clustering web search results with wikipedia disambiguation pages. In *Proceedings of International Conference on Database Systems for Advanced Applications*, pages 3–16. Springer, 2015.

[16] Xuanhui Wang, Deepayan Chakrabarti, and Kunal Punera. Mining broad latent query aspects from search sessions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 867–876. ACM, 2009.

[17] Xiaobing Xue and Xiaoxin Yin. Topic modeling for named entity queries. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2009–2012. ACM, 2011.

[18] Ashwin Swaminathan, Cherian V Mathew, and Darko Kirovski. Essential pages. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 173–182. IEEE Computer Society, 2009.

[19] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

[20] Xuanhui Wang and ChengXiang Zhai. Learn from web search logs to organize search results. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 87–94. ACM, 2007.

[21] Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. A survey of web clustering engines. *ACM Computing Surveys (CSUR)*, 41(3):17, 2009.

[22] Yoelle S Maarek, Ronald Fagin, Israel Z Ben-Shaul, and Dan Pelleg. Ephemeral document clustering for web applications. In *Proceedings of IBM RESEARCH REPORT RJ 10186*. Citeseer, 2000.

[23] Issam Sahmoudi and Abdelmonaime Lachkar. Clustering web search results for effective arabic language browsing. *arXiv preprint arXiv:1305.2755*, 2013.

[24] Issam Sahmoudi and Abdelmonaime Lachkar. Interactive system based on web search results clustering for arabic query reformulation. In *Proceedings of 2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, pages 300–305. IEEE, 2014.

[25] Hanane Froud, Abdelmonaime Lachkar, and Said Alaoui Ouatik. Arabic text summarization based on latent semantic analysis to enhance arabic documents clustering. *arXiv preprint arXiv:1302.1612*, 2013.

[26] Diab Abuaiadah. Using bisect k-means clustering technique in the analysis of arabic documents. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 15(3):17, 2016.

[27] Hanan M Alghamdi and Ali Selamat. Arabic web page clustering: A review. *Journal of King Saud University-Computer and Information Sciences*, 2017.

[28] Gerard Salton. Automatic information organization and retrieval. 1968.

[29] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.

[30] Wikipedia cluster analysis page. `https://en.wikipedia.org/wiki/Cluster_analysis`. Accessed: 2016-09-30.

[31] Weka web site. `http://www.cs.waikato.ac.nz/ml/weka/`.

[32] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *Science and Information Conference (SAI), 2014*, pages 372–378. IEEE, 2014.

[33] Charu C Aggarwal and ChengXiang Zhai. A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer, 2012.

[34] Nikolay Chumerin and Marc M Van Hulle. Comparison of two feature extraction methods based on maximization of mutual information. In *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, pages 343–348. IEEE, 2006.

[35] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[36] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

[37] Scikitlearn feature extraction page. `http://scikit-learn.org/stable/modules/feature_extraction.html`. Accessed: 2018-05-06.

[38] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[39] Xiaodan Zhang, Xiaohua Hu, and Xiaohua Zhou. A comparative evaluation of different link types on enhancing document clustering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 555–562. ACM, 2008.

[40] Steve Branson and Ari Greenberg. Clustering web search results using suffix tree methods. *Stanford University, Final Project Report or cite to http://www.stanford.edu/class/archive/cs/cs276a/cs276a*, 1:032, 2002.

[41] Daniel Crabtree, Xiaoying Gao, and Peter Andreae. Improving web clustering by cluster selection. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 172–178. IEEE Computer Society, 2005.

[42] Andrea Bernardini, Claudio Carpineto, and Massimiliano D'Amico. Full-subtopic retrieval with keyphrase-based search results clustering. In *Proceedings of Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 206–213. IET, 2009.

[43] Claudio Carpineto, Giovanni Romano, and Fondazione Ugo Bordoni. Exploiting the potential of concept lattices for information retrieval with credo. *J. UCS*, 10(8):985–1013, 2004.

[44] David Cheng, Ravi Kannan, Santosh Vempala, and Grant Wang. A divide-and-merge methodology for clustering. *ACM Transactions on Database Systems (TODS)*, 31(4):1499–1525, 2006.

[45] Ying Liu, Wenyuan Li, Yongjing Lin, and Liping Jing. Spectral geometry for simultaneously clustering and ranking query search results. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 539–546. ACM, 2008.

[46] Fatih Gelgi, Hasan Davulcu, and Srinivas Vadrevu. Term ranking for clustering web search results. In *Proceedings of WebDB*, 2007.

[47] Emilio Di Giacomo, Walter Didimo, Luca Grilli, and Giuseppe Liotta. Graph visualization techniques for web clustering engines. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):294–304, 2007.

[48] J Mai. Classification of the web: challenges and inquiries. *Knowledge organization*, 31(2):92, 2004.

[49] Gui-Rong Xue, Dikan Xing, Qiang Yang, and Yong Yu. Deep classification in large-scale text hierarchies. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 619–626. ACM, 2008.

[50] Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma. Support vector machines classification with a very large-scale taxonomy. *ACM SIGKDD Explorations Newsletter*, 7(1):36–43, 2005.

[51] Peter Bruza, Robert McArthur, and Simon Dennis. Interactive internet search: keyword, directory and query reformulation mechanisms compared. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 280–287. ACM, 2000.

[52] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

[53] Harr Chen and David R Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 429–436. ACM, 2006.

[54] Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. Improving web search results using affinity graph. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 504–511. ACM, 2005.

[55] Celina Santamaría, Julio Gonzalo, and Javier Artiles. Wikipedia as sense inventory to improve diversity in web search results. In *Proceedings of the 48th annual meeting of the association for computational Linguistics*, pages 1357–1366. Association for Computational Linguistics, 2010.

[56] Praveen Chandar and Ben Carterette. Diversification of search results using webgraphs. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 869–870. ACM, 2010.

[57] Hinrich Schutze and Jan O Pedersen. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*. Citeseer, 1995.

[58] Goldee Udani, Shachi Dave, Anthony Davis, and Tim Sibley. Noun sense induction using web search results. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 657–658. ACM, 2005.

[59] Cam-Tu Nguyen, Xuan-Hieu Phan, Susumu Horiguchi, Thu-Trang Nguyen, and Quang-Thuy Ha. Web search clustering and labeling with hidden topics. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(3):12, 2009.

[60] Jean Véronis. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252, 2004.

[61] Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *Icml*, volume 97, pages 412–420, 1997.

[62] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[63] W John Wilbur and Karl Sirotkin. The automatic identification of stop words. *Journal of information science*, 18(1):45–55, 1992.

[64] Yiming Yang. Noise reduction in a statistical approach to text categorization. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263. ACM, 1995.

[65] Hend Al-Khalifa and Areej Al-Wabil. The arabic language and the semantic web: Challenges and opportunities. In *The 1st int. symposium on computer and Arabic language*. Citeseer, 2007.

[66] Majdi Beseiso, Abdul Rahim Ahmad, and Roslan Ismail. An arabic language framework for semantic web. In *Semantic Technology and Information Retrieval (STAIR), 2011 International Conference on*, pages 7–11. IEEE, 2011.

[67] Andrey Kutuzov. Semantic clustering of russian web search results: possibilities and problems. In *Proceedings of Russian Summer School in Information Retrieval*, pages 320–331. Springer, 2014.

[68] Jiangning Wu and Zhijiang Wang. Search results clustering in chinese context based on a new suffix tree. In *Proceedings of Computer and Information Technology Workshops, 2008. IEEE 8th International Conference*, pages 110–115. IEEE, 2008.

[69] Burak Dural and Banu Diri. Search result clustering studies in turkish. In *Proceedings of Signal Processing and Communications Applications Conference (SIU), 2013 21st*, pages 1–4. IEEE, 2013.

[70] Issam Sahmoudi and Abdelmonaime Lachkar. Formal concept analysis for arabic web search results clustering. *Journal of King Saud University-Computer and Information Sciences*, 2016.

[71] Lei Ma, Tengyu Fu, Thomas Blaschke, Manchun Li, Dirk Tiede, Zhenjin Zhou, Xiaoxue Ma, and Deliang Chen. Evaluation of feature selection methods for object-based land cover mapping of unmanned aerial vehicle imagery using random forest and support vector machine classifiers. *ISPRS International Journal of Geo-Information*, 6(2):51, 2017.

[72] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217. ACM, 2004.

[73] O Ghanem and Mohammed Alhanjouri. Evaluating the effect of preprocessing in arabic documents clustering. Master's thesis, Computer Engineering Dept., Islamic University of Gaza, Palestine, 2014.

[74] Mahmud S Alkoffash. Comparing between arabic text clustering using k-means and k-mediods. *International Journal of Computer Applications (0975–8887) Volume*, 2012.

[75] Salma A Mahmood and Firas H Neama. Arabic text clustering using different similarity measures. *Indian Journal of Applied Research*, 6(11), 2017.

[76] Osama A Ghanem and Wesam M Ashour. Stemming effectiveness in clustering of arabic documents. *International Journal of Computer Applications*, 49(5), 2012.

[77] Omaia M Al-Omari. Evaluating the effect of stemming in clustering of arabic documents. *Academic Research International*, 1(1):284, 2011.

[78] Claudio Carpineto, Stefano Mizzaro, Giovanni Romano, and Matteo Snidero. Mobile information retrieval with search results clustering: Prototypes and evaluations. *JASIST*, 60(5):877–895, 2009.

[79] Roberto Navigli and Giuseppe Crisafulli. Inducing word senses to improve web search result clustering. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 116–126. Association for Computational Linguistics, 2010.

[80] Google custom search engine service. `https://cse.google.com/`. Accessed: 2016-09-30.

[81] Bing search api. `https://datamarket.azure.com/dataset/bing/search`. Accessed: 2016-09-30.

[82] Wikipedia blind relevance feedback page. `https://en.wikipedia.org/wiki/Relevance_feedback#Blind_feedback`. Accessed: 2018-05-05.

[83] Fabrizio Caruso, Giovanni Giuffrida, Diego Reforgiato, Giuseppe Tribulato, and Calogero Zarba. Ambiguity-aware document similarity.

[84] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[85] Hasna Chouikhi, Malika Charrad, and Nadia Ghazzali. A comparison study of clustering validity indices. In *Computer & Information Technology (GSCIT), 2015 Global Summit on*, pages 1–4. IEEE, 2015.

[86] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. *Foundations of statistical natural language processing*, volume 999, chapter 8, pages 300–301. MIT Press, 1999.

[87] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Appendix A

Data Schema

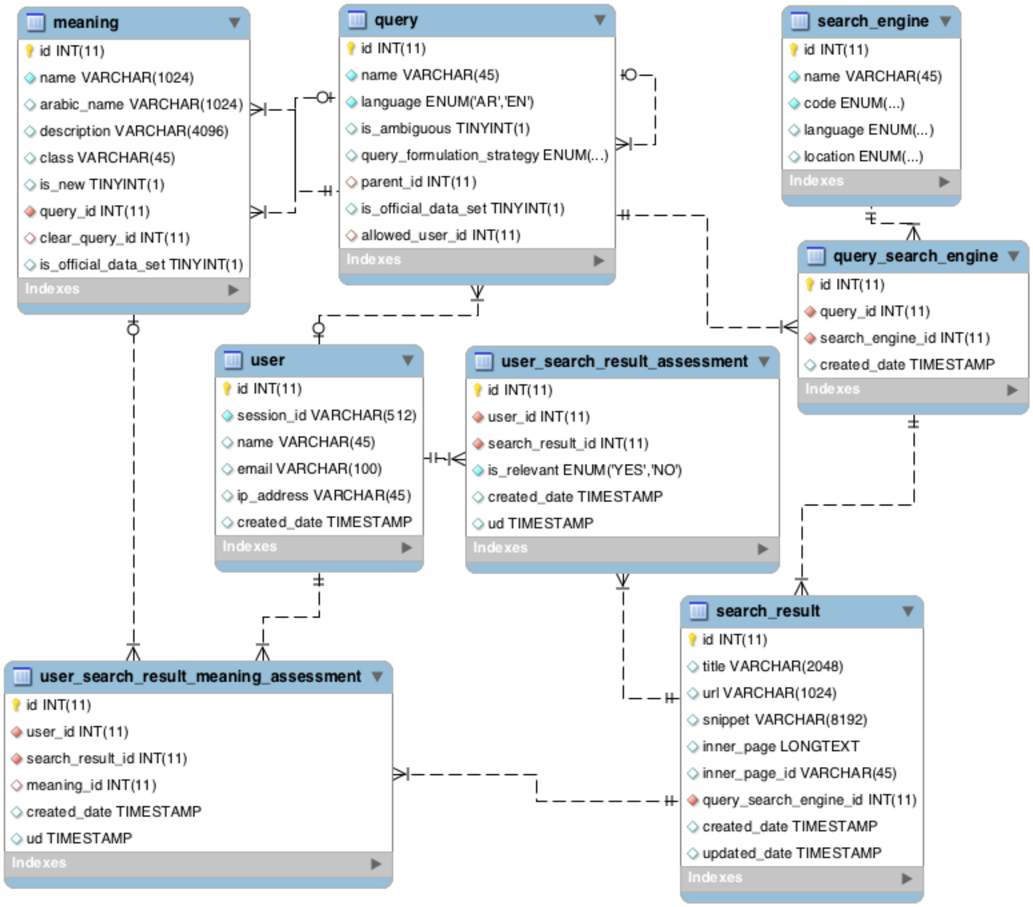


Figure A.1: A UML schema diagram for spread database.

Appendix B

Statistics about Words and Terms in Search Results

For the results of all official ambiguous and their clear queries. The following stats were measured:

Google

Table B.1: Statistics about words in the search results of Google.

Source	**Number of words detected in all search results**	**Number of results having words after preprocessing**	**Average number of words per search result**
title	77020	9300 out of 9300	8.3
snippet	261652	9295 out of 9300	28.2
title w/ snippet	338672	9300 out of 9300	36.4
inner page	11040849	8730 out of 9300	1264.7

Table B.2: Statistics about terms in the search results of Google.

Source	**Number of terms detected in all search results**	**Number of results having terms after preprocessing**	**Average number of terms per search result**
title	54076	9120 out of 9300	5.9
snippet	176775	9208 out of 9300	19.2
title w/ snippet	230851	9249 out of 9300	24.9
inner page	7231240	8474 out of 9300	853.3

BING

Table B.3: Statistics about words in the search results of Bing.

Source	**Number of words detected in all search results**	**Number of results having words after preprocessing**	**Average number of words per search result**
title	69480	9199 out of 9199	7.5
snippet	149547	9199 out of 9199	16.2
title w/ snippet	219027	9199 out of 9199	23.8
inner page	18629960	8482 out of 9199	2196.4

Table B.4: Statistics about terms in the search results of Bing.

Source	**Number of terms detected in all search results**	**Number of results having terms after preprocessing**	**Average number of terms per search result**
title	48353	9021 out of 9199	5.4
snippet	101594	9117 out of 9199	11.1
title w/ snippet	149947	9177 out of 9199	16.3
inner page	12762607	8386 out of 9199	1521.9

In this thesis, the intersections between the search results of Google and the search results of Bing and their Jaccard metric were calculated, you can browse them here for ambiguous queries[1] and clear queries[2].

[1] `https://github.com/haytham-salhi/Spread/blob/master/stats.csv`

[2] `https://github.com/haytham-salhi/Spread/blob/master/stats-clear-queries.csv`

Appendix C

List of Queries

Table C.1: List of all ambiguous queries and their clear queries.

Query	Number of meanings	Clear queries
امازون	2	نهر أمازون, شركة أمازون
عدنان إبراهيم	2	المخرج عدنان إبراهيم, المفكر عدنان إبراهيم
عمان	2	سلطنة عمان, مدينة عمان
جرار	2	جرار زراعي, بسام جرار
العشاء	2	صلاة العشاء, وجبة العشاء
نهاوند	2	مقام موسيقي نهاوند, مدينة فارسية نهاوند
القرش	2	عملة القرش, سمكة القرش
اسيا	2	قارة اسيا, اسية زوجة فرعون
شعر	2	شعر الجسم, شعر أدب
عرفات	2	جبل عرفات, ياسر عرفات
كاميرون	2	جمهورية كاميرون, ديفيد كاميرون
المالكي	3	المذهب المالكي, نوري المالكي, فايز المالكي
البقرة	2	حيوان البقرة, سورة البقرة
صخر	3	شركة صخر, صخر بن عمرو أخو الخنساء, صخر حجري
الاقتران	2	الاقتران الزواج, الاقتران في الرياضيات
العزيز	2	العزيز اسماء الله الحسنى, بوتيفار عزيز مصر
الاهرام	2	جريدة الاهرام المصرية, اهرام الجيزة
الجلمة	2	جلمة حماة, جلمة جنين
الملك عبد الله	2	عبد الله بن عبد العزيز آل سعود ملك السعودية, عبد الله الثاني بن الحسين ملك الأردن الحالي
الجامعة العربية	2	جامعة الدول العربية, الجامعة العربية المفتوحة
السكاكيني	2	جامع السكاكيني حلب, خليل السكاكيني
بورصة	2	مدينة بورصة التركية, بورصة سوق الاوراق المالية
العين	2	عين الحسد, عضو العين
الظاهرية	2	مذهب الظاهرية, ظاهرية الخليل
طرابلس	2	طرابلس مدينة لبنان, طرابلس عاصمة ليبيا
الازهر	2	جامعة الازهر مصر, جامعة الازهر غزة
العربية	2	اللغة العربية, قناة العربية
الاسد	3	بشار الاسد, حيوان الاسد, برج الاسد
العذراء	2	مريم العذراء, برج العذراء
القدرة	2	القدرة الخليلية, القدرة فيزياء

C.1 Examples of How the Clear Query is Formulated

Table C.2: Examples of how the clear query is formulated.

Ambiguous Query	Meaning	Formulated Query
المالكي		المالكي
	المذهب	المذهب المالكي
	فايز المالكي	فايز المالكي
عمان		عمان
	سلطنة	سلطنة عمان
	مدينة	مدينة عمان
الإمارات		الإمارات
	دولة	دولة الإمارات
	طيران	طيران الإمارات
أمازون		أمازون
	نهر	نهر أمازون
	شركة	شركة أمازون
البقرة		البقرة
	حيوان	حيوان البقرة
	سورة	سورة البقرة
عدنان إبراهيم		عدنان إبراهيم
	المخرج	المخرج عدنان إبراهيم
	المفكر	المفكر عدنان إبراهيم
اسيا		اسيا
	قارة	قارة اسيا
	اسية زوجة فرعون	اسية زوجة فرعون
	اسيا داغر	اسيا داغر
	مسلسل	مسلسل اسيا
عرفات		عرفات
	جبل	جبل عرفات
	ياسر عرفات	ياسر عرفات
العشاء		العشاء
	صلاة	صلاة العشاء
	وجبة	وجبة العشاء
القدرة		القدرة
	أكلة	أكلة القدرة
	فيزياء	فيزياء القدرة
القرش		القرش
	عملة	
	سمكة	سمكة القرش

C.2 Statistics about Human Judgments

Table C.3: Statistics about human judgments of Google mixing-based benchmark.

Ambiguous query	Meaning	Number of search items assessed	Judge 1	Judge 2	Number of agreements	Number of agreed sense-relevant items
Amazon	River	100	Haytham	Yaser	87	85
	Company	100	Haytham	Yaser	97	87
Adnan Ibrahim	Director	100	Haytham	Yaser	92	30
	Thinker	100	Haytham	Yaser	100	100
Amman	Sultanate	100	Haytham	Yaser	96	96
	City	100	Haytham	Yaser	90	83
Jarrar	Jarrar zeraee	100	Haytham	Yaser	99	99
	Bassam jarrar	100	Haytham	Yaser	99	99
Alishaa'	Prayer	100	Haytham	Yaser	99	99
	Meal	100	Haytham	Yaser	98	97
Nahawnd	Music	100	Haytham	Yaser	86	82
	City	100	Haytham	Yaser	86	30
Qersh	Currency	100	Haytham	Sireen	90	84
	Fish	100	Haytham	Sireen	93	93
Asia	Continent	100	Haytham	Sireen	87	84
	Girl	100	Haytham	Sireen	97	95
Shir (Shi3r)	Hair	100	Haytham	Motasem	98	98
	Art	100	Haytham	Motasem	100	100
Arafat	Mountain	100	Haytham	Motasem	100	100
	Yaser arafat	100	Haytham	Motasem	98	77
Cameron	Country	100	Haytham	Omar	81	65
	David	100	Haytham	Omar	100	99
Maliki	Doctrine	100	Haytham	Emad	92	92
	Noore	100	Haytham	Emad	94	92
	Fayez	100	Haytham	-	99	99
Baqara	Animal	100	Haytham	Emad	74	57
	Soura	100	Haytham	Emad	98	98
Sakhr	Company	100	Haytham	Anas	88	38
	Bn amro	100	Haytham	Anas	92	51
	Rock	100	Haytham	-	81	81
Iqteran	Marriage	100	Haytham	Anas	98	75
	Function	100	Haytham	Anas	100	100
Aziz	Aziz	100	Haytham	Anas	90	89
	Botefar	100	Haytham	Anas	98	96
Ahram	Newspaper	100	Haytham	Yasmeen	85	41
	Ahram	100	Haytham	Yasmeen	87	84
Jalamah	Homah	100	Haytham	Hamzah	90	48
	Jeneen	100	Haytham	Hamzah	91	78
Malek Abdullah	Saudi	100	Haytham	Aziza	98	72
	Jordanain	100	Haytham	Aziza	95	90
Jamaa Arabiya	League	100	Haytham	Aziza	88	88
	University	100	Haytham	Aziza	96	96
Sakakini	Mosque	100	Haytham	Mohannad	99	78
	Khalil	100	Haytham	Mohannad	95	52
Bursa	City	100	Haytham	Ahmad	98	97
	Market	100	Haytham	Ahmad	97	97
Ain	Envy	100	Haytham	Ahmad	94	94
	Eye	100	Haytham	Ahmad	93	33
Thaheryah	Doctrine	100	Haytham	Yazan	95	95
	Hebron	100	Haytham	Yazan	96	45
Tarablus	Lebanon city	100	Haytham	Mohannad	85	83
	Libyan City	100	Haytham	Mohannad	91	87
Azhar	Egypt	100	Haytham	Yazan	95	95
	Gazza	100	Haytham	Yazan	88	88
Arabiyah	Arabic	100	Haytham	Omar	97	93
	Channel	100	Haytham	Omar	84	68
Asad	Bashar	100	Haytham	Yaser	98	98
	Lion	100	Haytham	Yaser	100	100
	Leo	100	Haytham	-	100	100
Athraa'	Maryam	100	Haytham	Yaser	89	85
	Virgo	100	Haytham	Yaser	97	97
Qedra	Meal	100	Haytham	Emad	95	94
	Power	100	Haytham	Emad	91	71

Table C.4: Statistics about human judgments of Google plain benchmark.

Ambiguous query	**Sense**	**Number of search items assessed**	**Judge**	**Number of sense-relevant items**
Amman		100	Haytham, Yaser	
	Neither			70
	Sultanate			19
	Capital			11
Jarrar		100	Haytham, Yaser	
	Neither			30
	Tractor			2
	Bassam Jarrar			68
Alishaa’		100	Haytham, Yaser	
	Neither			23
	Prayer			63
	Meal			14
Arafat		100	Haytham, Yaser	
	Neither			62
	Arafat Mountain			8
	Yaser Arafat			30
Maliki		100	Haytham, Yaser	
	Neither			75
	Doctrine			5
	Nouri Almalki			20
Sakhr		100	Haytham, Yaser	
	Neither			90
	Company			3
	Sakhr Bn Amro			7
Malek Abdullah		100	Haytham, Yaser	
	Neither			72
	King of Saudia			17
	King of Jordan			11
Jamaa Arabiya		100	Haytham, Yaser	
	Neither			47
	Arab League			26
	Arab Open University			27
Bursa		100	Haytham, Yaser	
	Neither			43
	City			18
	Stock Exchange			39
Thaheryah		100	Haytham, Yaser	
	Neither			69
	Doctrine			8
	City			23
Tarablus		100	Haytham, Anas	
	Neither			67
	Lebanese City			11
	Capital of Libya			22
Arabiyah		100	Haytham, Yaser	
	Neither			79
	Arabic Language			9
	Alarabiya TV Channel			12
Athraa’		100	Haytham, Anas	
	Neither			38
	Maryam			28
	Virgo			34
Qedra		100	Haytham, Yaser	
	Neither			79
	Aklet Alqedra			7
	Physical Quantity			14

Table C.5: Statistics about human judgments of Bing plain benchmark.

Ambiguous query	**Sense**	**Number of search items assessed**	**Judge**	**Number of sense-relevant items**
Amman		200	Haytham, Anas	
	Neither			136
	Sultanate			43
	Capital			21
Jarrar		200	Haytham, Anas	
	Neither			104
	Tractor			75
	Bassam Jarrar			21
Alishaa’		200	Haytham, Anas	
	Neither			62
	Prayer			82
	Meal			56
Arafat		200	Haytham, Anas	
	Neither			96
	Arafat Mountain			36
	Yaser Arafat			68
Maliki		200	Haytham, Anas	
	Neither			140
	Doctrine			36
	Nouri Almalki			24
Sakhr		200	Haytham, Anas	
	Neither			153
	Company			25
	Sakhr Bn Amro			22
Malek Abdullah		200	Haytham, Anas	
	Neither			111
	King of Saudia			68
	King of Jordan			21
Jamaa Arabiya		200	Haytham, Anas	
	Neither			75
	Arab League			67
	Arab Open University			58
Bursa		200	Haytham, Anas	
	Neither			84
	City			61
	Stock Exchange			55
Thaheryah		200	Haytham, Anas	
	Neither			153
	Doctrine			19
	City			28
Tarablus		200	Haytham, Anas	
	Neither			123
	Lebanese City			26
	Capital of Libya			51
Arabiyah		200	Haytham, Anas	
	Neither			158
	Arabic Language			33
	Alarabiya TV Channel			9
Athraa’		200	Haytham, Anas	
	Neither			51
	Maryam			72
	Virgo			77
Qedra		200	Haytham, Anas	
	Neither			154
	Aklet Alqedra			17
	Physical Quantity			29

C.3 Detailed Evaluation Results

In all tables of this appendix, shortcuts are used as follows: title as **t**, snippet as **s**, inner page as **ip**, single words as **sw**, 2-grams as **2-g**, 3-grams as **3-g**.

C.3.1 Google/Clear Queries

Table C.6: Per level and query macro F-measure when using MBHA benchmarks for Google.

	t	**t**	**t**	**t**	**s**	**s**	**s**	**s**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	50%	40%	88%	88%	57%	44%	58%	58%
Adnan Ibrahim	42%	40%	70%	70%	79%	69%	79%	97%
Amman	47%	40%	40%	68%	43%	36%	43%	43%
Jarrar	40%	37%	40%	40%	52%	67%	67%	67%
Alishaa’	39%	47%	39%	55%	95%	43%	43%	43%
Nahawnd	58%	40%	58%	58%	83%	58%	79%	79%
Qersh	69%	40%	50%	59%	63%	37%	47%	47%
Asia	95%	37%	63%	63%	90%	55%	77%	77%
Shir (Shi3r)	44%	37%	86%	86%	81%	50%	83%	83%
Arafat	50%	37%	55%	38%	58%	63%	52%	81%
Cameron	54%	54%	50%	50%	36%	50%	81%	79%
Maliki	91%	19%	29%	29%	51%	29%	60%	55%
Baqara	43%	43%	43%	43%	46%	39%	61%	61%
Sakhr	36%	37%	38%	52%	52%	31%	44%	46%
Iqteran	44%	47%	58%	55%	61%	44%	58%	63%
Aziz	83%	44%	75%	82%	98%	52%	98%	98%
Ahram	65%	44%	49%	49%	67%	47%	50%	50%
Jalamah	86%	47%	55%	67%	59%	51%	51%	51%
Malek Abdullah	78%	47%	95%	90%	97%	98%	95%	95%
Jamaa Arabiya	42%	42%	42%	42%	65%	44%	95%	88%
Sakakini	65%	40%	69%	77%	93%	60%	93%	86%
Bursa	80%	37%	83%	58%	93%	88%	95%	93%
Ain	44%	37%	55%	58%	92%	40%	90%	90%
Thaheryah	71%	37%	44%	72%	46%	52%	65%	58%
Tarablus	48%	52%	48%	48%	95%	65%	93%	54%
Azhar	47%	37%	47%	56%	50%	39%	39%	39%
Arabiyah	97%	37%	47%	47%	50%	50%	50%	50%
Asad	46%	25%	30%	40%	32%	30%	34%	64%
Athraa’	63%	40%	40%	40%	86%	37%	90%	86%
Qedra	39%	44%	39%	39%	88%	50%	88%	88%

	t w/ s	**t w/ s**	**t w/ s**	**t w/ s**	**ip**	**ip**	**ip**	**ip**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	95%	47%	90%	90%	40%	42%	40%	40%
Adnan Ibrahim	79%	71%	98%	97%	64%	60%	64%	64%
Amman	81%	47%	69%	48%	37%	60%	37%	37%
Jarrar	97%	40%	65%	65%	40%	40%	40%	44%
Alishaa’	45%	47%	43%	43%	40%	40%	56%	73%
Nahawnd	88%	66%	63%	61%	75%	48%	71%	73%
Qersh	92%	44%	97%	88%	93%	37%	68%	68%
Asia	100%	60%	100%	100%	60%	61%	60%	60%
Shir (Shi3r)	86%	75%	86%	83%	40%	58%	93%	93%
Arafat	98%	65%	95%	95%	88%	79%	92%	92%
Cameron	54%	46%	54%	78%	63%	58%	65%	65%
Maliki	87%	36%	52%	61%	58%	36%	78%	78%
Baqara	97%	43%	83%	83%	50%	47%	50%	50%
Sakhr	49%	38%	68%	42%	38%	53%	57%	56%
Iqteran	65%	44%	55%	55%	44%	37%	93%	58%
Aziz	100%	56%	100%	100%	55%	45%	65%	37%
Ahram	60%	47%	60%	90%	81%	40%	81%	77%
Jalamah	92%	51%	83%	83%	85%	40%	43%	65%
Malek Abdullah	98%	98%	98%	98%	68%	67%	78%	95%
Jamaa Arabiya	92%	85%	98%	97%	71%	63%	73%	75%
Sakakini	92%	93%	93%	93%	92%	77%	92%	92%
Bursa	83%	58%	85%	85%	56%	67%	56%	59%
Ain	60%	50%	61%	61%	46%	58%	53%	55%
Thaheryah	100%	39%	100%	100%	37%	47%	63%	63%
Tarablus	65%	58%	82%	83%	40%	45%	40%	40%
Azhar	46%	61%	48%	48%	54%	36%	40%	40%
Arabiyah	67%	50%	69%	67%	47%	47%	47%	47%
Asad	65%	48%	62%	96%	74%	56%	63%	54%
Athraa’	63%	61%	86%	85%	95%	68%	75%	60%
Qedra	100%	42%	98%	98%	88%	63%	65%	65%

Table C.7: Per level and query weighted recall when using MBHA benchmarks for Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	58%	53%	88%	88%	58%	55%	62%	62%
Adnan Ibrahim	50%	53%	72%	72%	80%	72%	80%	97%
Amman	57%	53%	53%	70%	52%	50%	52%	52%
Jarrar	53%	52%	53%	53%	55%	70%	70%	70%
Alishaa'	52%	57%	52%	62%	95%	52%	52%	52%
Nahawnd	60%	53%	60%	60%	83%	63%	80%	80%
Qersh	72%	53%	58%	63%	67%	52%	57%	57%
Asia	95%	52%	67%	67%	90%	62%	78%	78%
Shir (Shi3r)	55%	52%	87%	87%	82%	58%	83%	83%
Arafat	58%	52%	62%	50%	63%	67%	60%	82%
Cameron	60%	60%	58%	58%	50%	58%	82%	80%
Maliki	91%	35%	40%	40%	53%	38%	62%	59%
Baqara	53%	53%	53%	53%	55%	52%	65%	65%
Sakhr	47%	43%	42%	51%	62%	40%	55%	58%
Iqteran	55%	57%	63%	62%	65%	53%	62%	67%
Aziz	83%	55%	75%	82%	98%	60%	98%	98%
Ahram	68%	55%	53%	53%	70%	57%	58%	58%
Jalamah	87%	57%	62%	70%	63%	58%	58%	58%
Malek Abdullah	78%	57%	95%	90%	97%	98%	95%	95%
Jamaa Arabiya	50%	50%	50%	50%	68%	55%	95%	88%
Sakakini	68%	53%	72%	78%	93%	65%	93%	87%
Bursa	80%	52%	83%	62%	93%	88%	95%	93%
Ain	55%	52%	62%	63%	92%	53%	90%	90%
Thaheryah	73%	52%	55%	73%	55%	60%	68%	63%
Tarablus	52%	60%	52%	52%	95%	68%	93%	60%
Azhar	57%	52%	57%	62%	52%	52%	52%	52%
Arabiyah	97%	52%	57%	57%	58%	58%	58%	58%
Asad	56%	38%	40%	47%	40%	38%	41%	64%
Athraa'	67%	53%	53%	53%	87%	52%	90%	87%
Qedra	52%	55%	52%	52%	88%	58%	88%	88%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	95%	57%	90%	90%	53%	52%	53%	53%
Adnan Ibrahim	80%	73%	98%	97%	67%	63%	67%	67%
Amman	82%	57%	72%	53%	52%	65%	52%	52%
Jarrar	97%	53%	65%	65%	53%	53%	53%	55%
Alishaa'	52%	57%	52%	52%	53%	53%	60%	73%
Nahawnd	88%	68%	65%	63%	77%	53%	73%	75%
Qersh	92%	55%	97%	88%	93%	52%	70%	70%
Asia	100%	65%	100%	100%	65%	65%	65%	65%
Shir (Shi3r)	87%	77%	87%	83%	53%	63%	93%	93%
Arafat	98%	68%	95%	95%	88%	80%	92%	92%
Cameron	60%	55%	60%	78%	67%	63%	68%	68%
Maliki	88%	47%	56%	66%	60%	44%	79%	79%
Baqara	97%	53%	83%	83%	58%	57%	58%	58%
Sakhr	61%	49%	67%	50%	44%	52%	62%	61%
Iqteran	68%	53%	62%	62%	55%	52%	93%	63%
Aziz	100%	62%	100%	100%	62%	52%	68%	52%
Ahram	65%	57%	65%	90%	82%	50%	82%	78%
Jalamah	92%	58%	83%	83%	85%	53%	52%	68%
Malek Abdullah	98%	98%	98%	98%	68%	67%	78%	95%
Jamaa Arabiya	92%	85%	98%	97%	73%	67%	75%	77%
Sakakini	92%	93%	93%	93%	92%	78%	92%	92%
Bursa	83%	63%	85%	85%	62%	68%	62%	63%
Ain	63%	58%	65%	65%	55%	63%	58%	60%
Thaheryah	100%	52%	100%	100%	52%	55%	67%	67%
Tarablus	65%	63%	82%	83%	53%	52%	53%	53%
Azhar	55%	63%	57%	57%	60%	50%	53%	53%
Arabiyah	70%	58%	72%	70%	57%	57%	57%	57%
Asad	68%	52%	67%	96%	77%	58%	66%	57%
Athraa'	67%	65%	87%	85%	95%	68%	77%	60%
Qedra	100%	52%	98%	98%	88%	67%	68%	68%

Table C.8: Per level and query weighted precision when using MBHA benchmarks for Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	77%	76%	89%	89%	59%	76%	67%	67%
Adnan Ibrahim	50%	76%	79%	79%	86%	82%	86%	97%
Amman	77%	76%	76%	78%	54%	50%	54%	54%
Jarrar	76%	75%	76%	76%	57%	81%	81%	81%
Alishaa'	59%	77%	59%	78%	95%	54%	54%	54%
Nahawnd	62%	76%	62%	62%	88%	79%	86%	86%
Qersh	82%	76%	77%	74%	80%	75%	77%	77%
Asia	95%	75%	80%	76%	92%	78%	85%	85%
Shir (Shi3r)	76%	75%	89%	89%	87%	77%	88%	88%
Arafat	77%	75%	78%	50%	79%	80%	78%	87%
Cameron	72%	72%	77%	77%	50%	77%	85%	86%
Maliki	92%	44%	72%	72%	53%	38%	69%	61%
Baqara	63%	63%	63%	63%	66%	59%	75%	75%
Sakhr	31%	53%	62%	64%	48%	64%	41%	43%
Iqteran	76%	77%	79%	78%	75%	59%	67%	80%
Aziz	85%	76%	75%	83%	98%	78%	98%	98%
Ahram	81%	76%	55%	55%	81%	77%	77%	77%
Jalamah	89%	77%	78%	81%	74%	70%	70%	70%
Malek Abdullah	81%	77%	95%	92%	97%	98%	95%	95%
Jamaa Arabiya	50%	50%	50%	50%	81%	76%	95%	91%
Sakakini	81%	76%	82%	85%	94%	79%	94%	89%
Bursa	81%	75%	84%	67%	94%	91%	95%	94%
Ain	76%	75%	78%	79%	93%	76%	90%	91%
Thaheryah	83%	75%	76%	80%	66%	78%	81%	79%
Tarablus	52%	78%	52%	52%	95%	81%	93%	72%
Azhar	77%	75%	77%	73%	52%	59%	59%	59%
Arabiyah	97%	75%	77%	77%	77%	77%	77%	77%
Asad	42%	78%	79%	76%	49%	55%	56%	83%
Athraa'	76%	76%	76%	76%	89%	75%	92%	89%
Qedra	59%	76%	59%	59%	91%	77%	91%	91%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	95%	77%	90%	90%	76%	55%	76%	76%
Adnan Ibrahim	84%	83%	98%	97%	73%	71%	73%	73%
Amman	87%	77%	82%	56%	75%	79%	75%	75%
Jarrar	97%	76%	66%	66%	76%	76%	76%	76%
Alishaa'	53%	77%	54%	54%	76%	76%	66%	76%
Nahawnd	91%	74%	70%	67%	84%	56%	83%	83%
Qersh	93%	76%	97%	91%	94%	75%	78%	78%
Asia	100%	79%	100%	100%	79%	75%	79%	79%
Shir (Shi3r)	89%	84%	89%	88%	76%	79%	94%	94%
Arafat	98%	81%	95%	95%	91%	86%	93%	93%
Cameron	72%	66%	72%	83%	80%	79%	81%	81%
Maliki	91%	36%	52%	66%	82%	79%	87%	86%
Baqara	97%	63%	88%	88%	77%	77%	77%	77%
Sakhr	43%	34%	70%	42%	57%	63%	76%	76%
Iqteran	81%	59%	78%	78%	76%	75%	94%	79%
Aziz	100%	73%	100%	100%	78%	53%	81%	75%
Ahram	79%	77%	79%	91%	87%	50%	87%	85%
Jalamah	92%	70%	88%	88%	88%	76%	54%	81%
Malek Abdullah	98%	98%	98%	98%	69%	67%	80%	95%
Jamaa Arabiya	93%	88%	98%	97%	83%	80%	83%	84%
Sakakini	93%	94%	94%	94%	93%	85%	93%	93%
Bursa	88%	79%	86%	86%	73%	73%	73%	74%
Ain	71%	77%	75%	75%	66%	79%	66%	68%
Thaheryah	100%	59%	100%	100%	75%	62%	80%	80%
Tarablus	65%	79%	83%	84%	76%	53%	76%	76%
Azhar	66%	67%	69%	69%	72%	50%	76%	76%
Arabiyah	81%	77%	82%	81%	77%	77%	77%	77%
Asad	70%	74%	68%	96%	86%	81%	80%	73%
Athraa'	76%	75%	89%	88%	95%	68%	84%	60%
Qedra	100%	55%	98%	98%	91%	80%	81%	81%

C.3.2 Bing/Clear Queries

Table C.9: Per level and query macro F-measure when using MBHA benchmarks for Bing.

	t	**t**	**t**	**t**	**s**	**s**	**s**	**s**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	47%	37%	47%	50%	79%	58%	55%	55%
Adnan Ibrahim	88%	69%	88%	65%	98%	52%	52%	52%
Amman	85%	40%	85%	85%	48%	44%	93%	93%
Jarrar	71%	47%	47%	47%	79%	40%	47%	47%
Alishaa'	79%	38%	83%	83%	58%	40%	50%	50%
Nahawnd	67%	44%	47%	47%	95%	44%	95%	95%
Qersh	88%	39%	52%	52%	95%	47%	86%	86%
Asia	86%	44%	67%	63%	95%	47%	95%	93%
Shir (Shi3r)	47%	47%	47%	47%	50%	40%	50%	50%
Arafat	88%	52%	92%	87%	68%	40%	76%	76%
Cameron	73%	44%	66%	66%	55%	44%	55%	55%
Maliki	49%	25%	53%	53%	70%	33%	61%	61%
Baqara	59%	46%	51%	51%	83%	44%	50%	50%
Sakhr	48%	49%	47%	47%	93%	76%	93%	47%
Iqteran	38%	38%	38%	38%	47%	44%	55%	98%
Aziz	77%	73%	83%	83%	69%	83%	83%	83%
Ahram	74%	36%	75%	75%	63%	40%	95%	95%
Jalamah	75%	52%	65%	65%	88%	52%	85%	85%
Malek Abdullah	92%	55%	85%	85%	100%	100%	100%	100%
Jamaa Arabiya	59%	92%	95%	98%	92%	92%	50%	95%
Sakakini	47%	44%	47%	47%	98%	50%	98%	98%
Bursa	77%	81%	79%	77%	100%	93%	98%	98%
Ain	51%	40%	47%	47%	78%	44%	81%	81%
Thaheryah	71%	40%	71%	71%	90%	60%	60%	60%
Tarablus	79%	40%	79%	69%	85%	50%	88%	88%
Azhar	75%	50%	75%	75%	86%	52%	75%	75%
Arabiyah	71%	44%	71%	47%	65%	44%	58%	58%
Asad	33%	33%	66%	66%	49%	40%	56%	89%
Athraa'	98%	40%	95%	95%	85%	44%	93%	93%
Qedra	97%	40%	93%	93%	56%	40%	58%	58%

	t w/ s	**t w/ s**	**t w/ s**	**t w/ s**	**ip**	**ip**	**ip**	**ip**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	55%	58%	55%	55%	98%	74%	97%	97%
Adnan Ibrahim	65%	86%	55%	100%	90%	60%	50%	50%
Amman	100%	40%	98%	98%	47%	37%	49%	49%
Jarrar	47%	44%	47%	47%	43%	44%	56%	50%
Alishaa'	46%	38%	95%	95%	50%	46%	65%	81%
Nahawnd	98%	44%	98%	98%	55%	44%	55%	55%
Qersh	100%	40%	80%	80%	65%	40%	73%	73%
Asia	100%	100%	100%	100%	83%	79%	85%	88%
Shir (Shi3r)	47%	47%	47%	47%	50%	44%	54%	54%
Arafat	100%	44%	100%	100%	92%	88%	92%	92%
Cameron	98%	37%	83%	83%	85%	55%	75%	69%
Maliki	65%	29%	84%	84%	41%	38%	41%	41%
Baqara	81%	40%	64%	64%	75%	40%	63%	83%
Sakhr	94%	76%	93%	93%	45%	33%	40%	40%
Iqteran	98%	38%	63%	60%	85%	37%	88%	93%
Aziz	90%	75%	90%	86%	63%	58%	79%	79%
Ahram	100%	44%	73%	73%	60%	47%	60%	60%
Jalamah	97%	52%	100%	100%	37%	47%	93%	47%
Malek Abdullah	100%	100%	100%	100%	47%	67%	81%	83%
Jamaa Arabiya	100%	100%	100%	100%	63%	58%	60%	60%
Sakakini	100%	47%	92%	88%	55%	44%	55%	55%
Bursa	97%	90%	97%	97%	83%	44%	75%	83%
Ain	88%	37%	85%	85%	65%	58%	77%	77%
Thaheryah	93%	60%	60%	60%	79%	37%	79%	81%
Tarablus	100%	58%	100%	100%	51%	38%	50%	50%
Azhar	88%	71%	77%	73%	59%	37%	69%	71%
Arabiyah	73%	40%	71%	71%	37%	44%	55%	58%
Asad	98%	33%	62%	62%	46%	39%	84%	40%
Athraa'	100%	52%	100%	100%	69%	63%	95%	95%
Qedra	79%	58%	71%	71%	37%	46%	37%	37%

Table C.10: Per level and query weighted recall when using MBHA benchmarks for Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw_2-g**	**sw_2_3-g**	**sw**	**2-g**	**sw_2-g**	**sw_2_3-g**
Amazon	57%	52%	57%	58%	80%	63%	62%	62%
Adnan Ibrahim	88%	72%	88%	68%	98%	60%	60%	60%
Amman	85%	53%	85%	85%	57%	55%	93%	93%
Jarrar	73%	57%	57%	57%	80%	53%	57%	57%
Alishaa'	80%	50%	83%	83%	63%	53%	57%	57%
Nahawnd	70%	55%	57%	57%	95%	55%	95%	95%
Qersh	88%	52%	60%	60%	95%	57%	87%	87%
Asia	87%	55%	70%	67%	95%	57%	95%	93%
Shir (Shi3r)	55%	57%	57%	57%	58%	53%	58%	58%
Arafat	88%	60%	92%	87%	70%	53%	77%	77%
Cameron	75%	55%	68%	68%	62%	55%	62%	62%
Maliki	54%	37%	60%	60%	70%	42%	62%	62%
Baqara	63%	55%	58%	58%	83%	55%	58%	58%
Sakhr	52%	53%	51%	51%	93%	76%	93%	60%
Iqteran	50%	50%	50%	50%	55%	55%	62%	98%
Aziz	78%	75%	83%	83%	72%	83%	83%	83%
Ahram	75%	50%	77%	77%	67%	53%	95%	95%
Jalamah	77%	60%	68%	68%	88%	60%	85%	85%
Malek Abdullah	92%	62%	85%	85%	100%	100%	100%	100%
Jamaa Arabiya	60%	92%	95%	98%	92%	92%	58%	95%
Sakakini	55%	55%	55%	55%	98%	58%	98%	98%
Bursa	78%	82%	80%	78%	100%	93%	98%	98%
Ain	58%	53%	57%	57%	78%	55%	82%	82%
Thaheryah	73%	53%	73%	73%	90%	65%	65%	65%
Tarablus	80%	50%	80%	72%	85%	58%	88%	88%
Azhar	77%	58%	77%	77%	87%	60%	77%	77%
Arabiyah	73%	55%	73%	57%	68%	55%	63%	63%
Asad	42%	41%	69%	69%	60%	47%	63%	89%
Athraa'	98%	53%	95%	95%	85%	55%	93%	93%
Qedra	97%	53%	93%	93%	60%	53%	63%	63%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw_2-g**	**sw_2_3-g**	**sw**	**2-g**	**sw_2-g**	**sw_2_3-g**
Amazon	62%	63%	62%	62%	98%	75%	97%	97%
Adnan Ibrahim	68%	87%	62%	100%	90%	65%	58%	58%
Amman	100%	53%	98%	98%	52%	52%	53%	53%
Jarrar	57%	55%	57%	57%	53%	55%	62%	58%
Alishaa'	55%	50%	95%	95%	58%	55%	68%	82%
Nahawnd	98%	55%	98%	98%	62%	55%	62%	62%
Qersh	100%	53%	80%	80%	68%	53%	75%	75%
Asia	100%	100%	100%	100%	83%	80%	85%	88%
Shir (Shi3r)	55%	57%	55%	55%	58%	55%	60%	60%
Arafat	100%	55%	100%	100%	92%	88%	92%	92%
Cameron	98%	52%	83%	83%	85%	62%	77%	72%
Maliki	68%	39%	84%	84%	49%	46%	49%	49%
Baqara	82%	53%	67%	67%	77%	53%	67%	83%
Sakhr	94%	76%	93%	93%	54%	44%	51%	51%
Iqteran	98%	50%	67%	65%	85%	52%	88%	93%
Aziz	90%	77%	90%	87%	67%	63%	80%	80%
Ahram	100%	55%	75%	75%	65%	57%	65%	65%
Jalamah	97%	60%	100%	100%	52%	57%	93%	57%
Malek Abdullah	100%	100%	100%	100%	57%	70%	82%	83%
Jamaa Arabiya	100%	100%	100%	100%	67%	63%	65%	65%
Sakakini	100%	57%	92%	88%	62%	55%	62%	62%
Bursa	97%	90%	97%	97%	83%	55%	77%	83%
Ain	88%	52%	85%	85%	68%	63%	78%	78%
Thaheryah	93%	65%	65%	65%	80%	52%	80%	82%
Tarablus	100%	63%	100%	100%	53%	50%	52%	52%
Azhar	88%	73%	78%	75%	63%	52%	72%	73%
Arabiyah	75%	53%	73%	73%	52%	55%	62%	63%
Asad	98%	41%	63%	63%	55%	47%	84%	47%
Athraa'	100%	60%	100%	100%	72%	67%	95%	95%
Qedra	80%	63%	73%	73%	52%	53%	52%	52%

Table C.11: Per level and query weighted precision when using MBHA benchmarks for Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	77%	75%	77%	77%	86%	79%	78%	78%
Adnan Ibrahim	91%	82%	91%	81%	98%	78%	78%	78%
Amman	88%	76%	88%	88%	69%	76%	94%	94%
Jarrar	83%	77%	77%	77%	86%	76%	77%	77%
Alishaa’	84%	50%	83%	83%	79%	76%	64%	64%
Nahawnd	81%	76%	77%	77%	95%	76%	95%	95%
Qersh	91%	59%	78%	78%	95%	77%	89%	89%
Asia	89%	76%	81%	80%	95%	77%	95%	94%
Shir (Shi3r)	62%	77%	77%	77%	77%	76%	77%	77%
Arafat	89%	78%	92%	87%	78%	76%	82%	82%
Cameron	83%	76%	77%	77%	78%	76%	78%	78%
Maliki	64%	62%	57%	57%	82%	79%	82%	82%
Baqara	74%	66%	70%	70%	88%	76%	77%	77%
Sakhr	59%	81%	59%	59%	94%	86%	94%	41%
Iqteran	50%	50%	50%	50%	62%	76%	78%	98%
Aziz	85%	83%	88%	88%	82%	88%	88%	88%
Ahram	79%	50%	84%	84%	80%	76%	95%	95%
Jalamah	84%	78%	81%	81%	89%	78%	88%	88%
Malek Abdullah	92%	78%	88%	88%	100%	100%	100%	100%
Jamaa Arabiya	61%	93%	95%	98%	92%	93%	77%	95%
Sakakini	62%	76%	62%	62%	98%	77%	98%	98%
Bursa	85%	87%	86%	85%	100%	94%	98%	98%
Ain	70%	76%	77%	77%	80%	76%	85%	85%
Thaheryah	83%	76%	83%	83%	92%	79%	79%	79%
Tarablus	86%	50%	86%	82%	88%	77%	91%	91%
Azhar	84%	77%	84%	84%	89%	78%	84%	84%
Arabiyah	83%	76%	83%	77%	81%	76%	79%	79%
Asad	54%	57%	84%	84%	45%	79%	71%	92%
Athraa’	98%	76%	95%	95%	88%	76%	94%	94%
Qedra	97%	76%	94%	94%	66%	76%	79%	79%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	78%	79%	78%	78%	98%	79%	97%	97%
Adnan Ibrahim	81%	89%	78%	100%	91%	79%	77%	77%
Amman	100%	76%	98%	98%	52%	75%	55%	55%
Jarrar	77%	76%	77%	77%	63%	76%	73%	77%
Alishaa’	66%	50%	95%	95%	77%	66%	81%	85%
Nahawnd	98%	76%	98%	98%	78%	76%	78%	78%
Qersh	100%	76%	82%	82%	81%	76%	83%	83%
Asia	100%	100%	100%	100%	88%	86%	88%	91%
Shir (Shi3r)	62%	77%	62%	62%	77%	76%	72%	72%
Arafat	100%	76%	100%	100%	93%	91%	93%	93%
Cameron	98%	75%	86%	86%	88%	78%	84%	82%
Maliki	84%	62%	89%	89%	80%	79%	80%	80%
Baqara	83%	76%	73%	73%	84%	76%	80%	88%
Sakhr	95%	86%	94%	94%	47%	44%	41%	41%
Iqteran	98%	50%	80%	79%	85%	75%	91%	94%
Aziz	92%	84%	92%	89%	80%	79%	86%	86%
Ahram	100%	76%	83%	83%	79%	77%	79%	79%
Jalamah	97%	78%	100%	100%	75%	77%	94%	77%
Malek Abdullah	100%	100%	100%	100%	77%	81%	87%	88%
Jamaa Arabiya	100%	100%	100%	100%	80%	79%	79%	79%
Sakakini	100%	77%	93%	91%	78%	76%	78%	78%
Bursa	97%	92%	97%	97%	88%	76%	84%	88%
Ain	91%	75%	88%	88%	81%	79%	85%	85%
Thaheryah	94%	79%	79%	79%	86%	75%	86%	87%
Tarablus	100%	79%	100%	100%	54%	50%	52%	52%
Azhar	91%	83%	85%	83%	74%	75%	82%	83%
Arabiyah	83%	76%	83%	83%	75%	76%	78%	79%
Asad	98%	54%	83%	83%	47%	79%	89%	51%
Athraa’	100%	78%	100%	100%	82%	80%	95%	95%
Qedra	86%	79%	83%	83%	75%	57%	75%	75%

C.3.3 Google and Bing/Clear Queries

Table C.12: Per level and query macro F-measure when using MBHA benchmarks for both.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	39%	37%	67%	46%	75%	56%	64%	44%
Adnan Ibrahim	72%	39%	56%	69%	81%	63%	78%	71%
Amman	56%	56%	56%	56%	86%	39%	67%	67%
Jarrar	40%	37%	61%	61%	44%	42%	66%	66%
Alishaa'	39%	39%	39%	39%	63%	39%	63%	63%
Nahawnd	49%	48%	49%	49%	85%	46%	82%	82%
Qersh	83%	40%	51%	51%	42%	46%	41%	41%
Asia	87%	56%	85%	85%	92%	66%	68%	65%
Shir (Shi3r)	74%	67%	69%	69%	64%	42%	45%	63%
Arafat	44%	37%	51%	51%	45%	55%	52%	52%
Cameron	66%	37%	40%	40%	72%	54%	71%	69%
Maliki	25%	26%	58%	58%	35%	27%	33%	33%
Baqara	40%	40%	40%	40%	58%	44%	75%	75%
Sakhr	46%	30%	44%	38%	76%	33%	71%	71%
Iqteran	69%	56%	58%	58%	68%	47%	84%	48%
Aziz	50%	50%	50%	50%	56%	69%	54%	88%
Ahram	49%	43%	43%	43%	97%	52%	96%	96%
Jalamah	60%	41%	58%	82%	59%	43%	77%	77%
Malek Abdullah	69%	59%	59%	87%	98%	99%	99%	97%
Jamaa Arabiya	43%	43%	43%	43%	85%	89%	88%	86%
Sakakini	48%	48%	48%	48%	97%	65%	97%	65%
Bursa	57%	57%	57%	57%	95%	51%	93%	92%
Ain	47%	50%	35%	47%	54%	39%	58%	58%
Thaheryah	43%	39%	54%	54%	58%	45%	46%	59%
Tarablus	55%	60%	50%	50%	86%	38%	77%	50%
Azhar	51%	39%	51%	51%	47%	46%	46%	46%
Arabiyah	55%	37%	54%	54%	69%	51%	66%	66%
Asad	39%	28%	54%	65%	50%	27%	54%	67%
Athraa'	47%	39%	47%	47%	97%	48%	96%	96%
Qedra	52%	40%	54%	54%	93%	37%	93%	93%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	99%	37%	51%	51%	37%	37%	37%	37%
Adnan Ibrahim	95%	68%	57%	57%	68%	52%	43%	74%
Amman	53%	58%	57%	54%	60%	51%	57%	57%
Jarrar	44%	42%	44%	44%	42%	42%	44%	42%
Alishaa'	78%	39%	51%	51%	79%	37%	84%	82%
Nahawnd	49%	56%	49%	49%	70%	65%	71%	71%
Qersh	52%	47%	51%	51%	37%	58%	37%	37%
Asia	100%	69%	100%	100%	76%	69%	74%	73%
Shir (Shi3r)	65%	58%	63%	63%	73%	69%	40%	40%
Arafat	82%	50%	73%	52%	74%	39%	74%	72%
Cameron	74%	64%	72%	69%	68%	50%	67%	67%
Maliki	97%	28%	34%	34%	42%	44%	44%	44%
Baqara	72%	46%	76%	76%	41%	39%	42%	42%
Sakhr	44%	42%	44%	44%	29%	30%	29%	29%
Iqteran	54%	48%	68%	48%	71%	51%	65%	66%
Aziz	50%	52%	94%	94%	67%	43%	37%	37%
Ahram	56%	51%	56%	86%	61%	42%	60%	59%
Jalamah	99%	42%	99%	99%	38%	54%	57%	54%
Malek Abdullah	100%	97%	99%	99%	40%	51%	46%	52%
Jamaa Arabiya	43%	43%	43%	43%	37%	54%	92%	93%
Sakakini	97%	71%	71%	70%	37%	63%	47%	35%
Bursa	92%	56%	57%	57%	52%	53%	55%	93%
Ain	46%	52%	52%	50%	45%	45%	48%	48%
Thaheryah	100%	45%	80%	80%	80%	45%	78%	80%
Tarablus	81%	63%	45%	45%	49%	35%	49%	47%
Azhar	49%	63%	48%	48%	40%	39%	40%	40%
Arabiyah	58%	45%	56%	56%	50%	44%	56%	64%
Asad	99%	25%	57%	57%	31%	41%	54%	60%
Athraa'	55%	56%	98%	98%	52%	53%	84%	84%
Qedra	100%	40%	100%	100%	43%	71%	77%	75%

Table C.13: Per level and query weighted recall when using MBHA benchmarks for both.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	53%	52%	70%	55%	76%	63%	68%	53%
Adnan Ibrahim	74%	53%	62%	72%	82%	67%	79%	73%
Amman	61%	61%	61%	61%	87%	53%	70%	70%
Jarrar	53%	52%	66%	66%	53%	54%	69%	69%
Alishaa'	51%	51%	51%	51%	66%	51%	66%	66%
Nahawnd	53%	53%	53%	53%	85%	55%	83%	83%
Qersh	83%	53%	59%	59%	52%	56%	50%	50%
Asia	88%	63%	85%	85%	92%	69%	71%	68%
Shir (Shi3r)	76%	70%	72%	72%	68%	54%	56%	67%
Arafat	55%	52%	59%	59%	54%	62%	58%	58%
Cameron	69%	52%	53%	53%	73%	61%	73%	72%
Maliki	36%	37%	64%	64%	41%	37%	42%	42%
Baqara	52%	52%	52%	52%	63%	55%	76%	76%
Sakhr	49%	38%	47%	43%	78%	42%	73%	74%
Iqteran	72%	63%	63%	63%	71%	57%	84%	58%
Aziz	54%	54%	54%	54%	63%	72%	61%	88%
Ahram	58%	52%	52%	52%	97%	60%	96%	96%
Jalamah	65%	53%	63%	83%	63%	54%	78%	78%
Malek Abdullah	69%	59%	60%	88%	98%	99%	99%	98%
Jamaa Arabiya	52%	52%	52%	52%	85%	89%	88%	86%
Sakakini	58%	58%	58%	58%	98%	68%	98%	68%
Bursa	61%	61%	61%	61%	95%	59%	93%	93%
Ain	57%	58%	51%	57%	61%	51%	63%	63%
Thaheryah	53%	53%	61%	61%	63%	56%	55%	64%
Tarablus	58%	65%	58%	58%	86%	51%	78%	58%
Azhar	58%	53%	58%	58%	54%	56%	53%	53%
Arabiyah	62%	52%	61%	61%	72%	59%	68%	68%
Asad	47%	40%	57%	67%	62%	39%	62%	72%
Athraa'	57%	53%	57%	57%	97%	57%	96%	96%
Qedra	60%	53%	61%	61%	93%	50%	93%	93%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	99%	50%	59%	59%	52%	52%	52%	52%
Adnan Ibrahim	95%	71%	63%	63%	70%	58%	52%	76%
Amman	59%	63%	62%	60%	62%	59%	59%	59%
Jarrar	53%	54%	53%	53%	54%	54%	55%	54%
Alishaa'	79%	51%	59%	59%	79%	52%	84%	83%
Nahawnd	53%	60%	53%	53%	73%	68%	73%	73%
Qersh	60%	57%	59%	59%	52%	63%	52%	52%
Asia	100%	72%	100%	100%	78%	72%	76%	75%
Shir (Shi3r)	68%	63%	67%	67%	75%	72%	53%	53%
Arafat	83%	58%	74%	58%	76%	52%	76%	74%
Cameron	75%	68%	73%	72%	71%	54%	70%	70%
Maliki	97%	37%	42%	42%	53%	53%	51%	51%
Baqara	73%	53%	77%	77%	53%	53%	54%	54%
Sakhr	49%	47%	48%	48%	40%	39%	40%	40%
Iqteran	61%	58%	71%	58%	73%	59%	68%	69%
Aziz	54%	56%	94%	94%	67%	51%	52%	52%
Ahram	63%	59%	58%	86%	63%	52%	63%	63%
Jalamah	99%	51%	99%	99%	51%	61%	63%	59%
Malek Abdullah	100%	97%	99%	99%	53%	57%	56%	53%
Jamaa Arabiya	51%	52%	52%	52%	51%	61%	93%	93%
Sakakini	98%	73%	73%	73%	52%	67%	57%	51%
Bursa	93%	60%	61%	61%	59%	58%	60%	93%
Ain	51%	59%	58%	58%	53%	53%	57%	57%
Thaheryah	100%	56%	81%	81%	81%	56%	79%	81%
Tarablus	81%	67%	51%	51%	53%	51%	53%	52%
Azhar	57%	65%	56%	56%	53%	51%	53%	53%
Arabiyah	63%	56%	63%	63%	53%	55%	63%	68%
Asad	99%	36%	63%	63%	42%	47%	65%	68%
Athraa'	56%	62%	98%	98%	54%	56%	84%	84%
Qedra	100%	53%	100%	100%	54%	73%	78%	77%

Table C.14: Per level and query weighted precision when using MBHA benchmarks for both.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	76%	75%	81%	64%	80%	79%	80%	61%
Adnan Ibrahim	83%	76%	73%	82%	87%	80%	85%	83%
Amman	70%	70%	70%	70%	89%	76%	81%	81%
Jarrar	66%	75%	80%	80%	61%	76%	81%	81%
Alishaa'	54%	54%	54%	54%	74%	54%	74%	74%
Nahawnd	55%	54%	55%	55%	88%	66%	84%	84%
Qersh	86%	76%	78%	78%	55%	71%	50%	50%
Asia	90%	79%	88%	88%	93%	81%	82%	81%
Shir (Shi3r)	84%	81%	82%	82%	80%	76%	77%	80%
Arafat	76%	75%	78%	78%	63%	78%	68%	68%
Cameron	81%	75%	66%	66%	81%	78%	81%	82%
Maliki	53%	65%	70%	70%	45%	45%	44%	44%
Baqara	57%	57%	57%	57%	72%	76%	79%	79%
Sakhr	60%	50%	59%	58%	86%	79%	85%	85%
Iqteran	82%	79%	79%	79%	82%	77%	88%	77%
Aziz	56%	56%	56%	56%	79%	82%	78%	91%
Ahram	73%	54%	54%	54%	97%	78%	96%	96%
Jalamah	79%	68%	76%	87%	68%	69%	85%	85%
Malek Abdullah	70%	59%	61%	90%	98%	99%	99%	98%
Jamaa Arabiya	54%	54%	54%	54%	88%	91%	91%	89%
Sakakini	77%	77%	77%	77%	98%	81%	98%	81%
Bursa	67%	67%	67%	67%	95%	78%	94%	93%
Ain	77%	69%	75%	77%	78%	54%	79%	79%
Thaheryah	58%	76%	78%	78%	76%	77%	66%	79%
Tarablus	60%	79%	77%	77%	89%	55%	85%	77%
Azhar	67%	76%	67%	67%	60%	71%	55%	55%
Arabiyah	78%	75%	78%	78%	82%	78%	77%	77%
Asad	64%	44%	69%	75%	45%	36%	57%	85%
Athraa'	77%	76%	77%	77%	97%	72%	96%	96%
Qedra	78%	76%	75%	75%	94%	50%	94%	94%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	99%	50%	78%	78%	75%	75%	75%	75%
Adnan Ibrahim	95%	82%	76%	76%	78%	64%	54%	84%
Amman	69%	72%	71%	70%	64%	78%	62%	62%
Jarrar	61%	76%	61%	61%	76%	76%	76%	76%
Alishaa'	85%	54%	78%	78%	81%	75%	87%	86%
Nahawnd	55%	67%	55%	55%	82%	81%	83%	83%
Qersh	78%	77%	78%	78%	75%	79%	75%	75%
Asia	100%	82%	100%	100%	84%	82%	84%	83%
Shir (Shi3r)	81%	79%	80%	80%	83%	82%	76%	76%
Arafat	86%	69%	79%	68%	84%	59%	84%	83%
Cameron	82%	80%	81%	82%	82%	56%	81%	81%
Maliki	97%	44%	42%	42%	35%	44%	63%	62%
Baqara	78%	57%	81%	81%	68%	76%	76%	76%
Sakhr	62%	60%	61%	61%	41%	59%	41%	41%
Iqteran	75%	77%	82%	77%	83%	78%	81%	81%
Aziz	56%	59%	95%	95%	67%	52%	75%	75%
Ahram	79%	78%	60%	86%	69%	55%	67%	68%
Jalamah	99%	52%	99%	99%	55%	78%	76%	66%
Malek Abdullah	100%	97%	99%	99%	76%	63%	71%	53%
Jamaa Arabiya	52%	54%	54%	54%	59%	78%	93%	94%
Sakakini	98%	83%	83%	82%	75%	80%	77%	75%
Bursa	93%	67%	67%	67%	74%	66%	68%	94%
Ain	51%	71%	68%	69%	58%	58%	72%	72%
Thaheryah	100%	77%	86%	86%	86%	77%	85%	86%
Tarablus	81%	80%	51%	51%	55%	75%	55%	52%
Azhar	66%	68%	65%	65%	76%	54%	76%	76%
Arabiyah	79%	77%	79%	79%	55%	76%	79%	80%
Asad	99%	61%	60%	60%	62%	62%	49%	78%
Athraa'	56%	73%	98%	98%	55%	58%	88%	88%
Qedra	100%	76%	100%	100%	69%	83%	85%	84%

C.3.4 Google/BRF

Table C.15: Per level and query macro F-measure when using BRF benchmarks for Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	sw	2-g	sw__2-g	sw__2__3-g	sw	2-g	sw__2-g	sw__2__3-g
Amazon	41%	41%	41%	41%	75%	49%	85%	86%
Adnan Ibrahim	58%	48%	58%	58%	50%	49%	53%	43%
Amman	71%	47%	60%	71%	38%	38%	79%	79%
Jarrar	42%	38%	45%	45%	38%	51%	66%	66%
Alishaa'	47%	40%	47%	45%	66%	44%	93%	54%
Nahawnd	59%	50%	59%	59%	89%	48%	89%	89%
Qersh	38%	52%	39%	43%	50%	50%	50%	50%
Asia	92%	38%	92%	68%	94%	71%	92%	92%
Shir (Shi3r)	92%	42%	81%	81%	76%	47%	71%	79%
Arafat	42%	42%	42%	42%	59%	40%	67%	67%
Cameron	56%	56%	56%	56%	91%	39%	49%	52%
Maliki	44%	21%	56%	65%	30%	30%	42%	42%
Baqara	47%	67%	49%	49%	50%	45%	45%	45%
Sakhr	39%	33%	36%	39%	50%	30%	48%	47%
Iqteran	66%	41%	49%	49%	73%	49%	73%	77%
Aziz	61%	67%	57%	58%	72%	52%	72%	72%
Ahram	49%	38%	49%	49%	78%	40%	87%	87%
Jalamah	43%	38%	43%	43%	63%	56%	63%	63%
Malek Abdullah	83%	93%	93%	92%	97%	92%	93%	92%
Jamaa Arabiya	49%	56%	49%	49%	57%	91%	94%	91%
Sakakini	73%	38%	40%	51%	90%	60%	90%	80%
Bursa	72%	72%	72%	72%	96%	52%	95%	89%
Ain	45%	45%	45%	45%	45%	39%	72%	72%
Thaheryah	47%	54%	47%	47%	59%	40%	94%	94%
Tarablus	49%	49%	49%	49%	51%	52%	51%	51%
Azhar	50%	42%	43%	49%	63%	45%	47%	47%
Arabiyah	38%	38%	52%	51%	45%	45%	44%	45%
Asad	35%	31%	50%	50%	41%	25%	59%	42%
Athraa'	58%	58%	58%	59%	92%	45%	94%	94%
Qedra	57%	40%	89%	88%	44%	45%	44%	44%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	sw	2-g	sw__2-g	sw__2__3-g	sw	2-g	sw__2-g	sw__2__3-g
Amazon	96%	49%	96%	96%	38%	44%	38%	37%
Adnan Ibrahim	59%	58%	58%	58%	35%	52%	67%	69%
Amman	97%	67%	69%	69%	36%	58%	60%	55%
Jarrar	99%	44%	68%	64%	77%	71%	76%	79%
Alishaa'	99%	49%	99%	99%	37%	45%	37%	37%
Nahawnd	61%	45%	59%	59%	48%	43%	74%	76%
Qersh	53%	40%	94%	53%	86%	36%	80%	69%
Asia	100%	39%	99%	99%	45%	61%	97%	97%
Shir (Shi3r)	80%	71%	80%	82%	92%	54%	92%	92%
Arafat	97%	49%	76%	96%	38%	38%	39%	48%
Cameron	92%	57%	93%	56%	55%	55%	52%	53%
Maliki	58%	38%	58%	58%	53%	23%	52%	74%
Baqara	49%	49%	65%	49%	50%	77%	79%	79%
Sakhr	70%	39%	69%	49%	62%	46%	48%	51%
Iqteran	58%	49%	78%	59%	76%	47%	78%	75%
Aziz	73%	58%	91%	59%	68%	52%	65%	65%
Ahram	64%	44%	49%	49%	63%	36%	63%	63%
Jalamah	52%	72%	63%	63%	43%	70%	43%	43%
Malek Abdullah	99%	96%	100%	97%	72%	48%	75%	76%
Jamaa Arabiya	55%	97%	49%	49%	60%	54%	72%	70%
Sakakini	90%	60%	90%	85%	91%	81%	88%	91%
Bursa	99%	94%	71%	86%	79%	64%	67%	66%
Ain	92%	45%	56%	56%	71%	59%	71%	72%
Thaheryah	100%	57%	100%	100%	56%	57%	81%	60%
Tarablus	55%	52%	52%	49%	53%	43%	49%	42%
Azhar	42%	43%	86%	86%	36%	62%	50%	50%
Arabiyah	56%	47%	52%	52%	57%	47%	59%	59%
Asad	37%	38%	38%	95%	33%	41%	23%	23%
Athraa'	56%	56%	58%	58%	52%	65%	68%	65%
Qedra	44%	44%	99%	99%	62%	40%	63%	64%

Table C.16: Per level and query weighted recall when using BRF benchmarks for Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	51%	51%	51%	51%	76%	58%	85%	86%
Adnan Ibrahim	63%	57%	63%	63%	55%	58%	57%	50%
Amman	73%	57%	65%	73%	50%	52%	79%	79%
Jarrar	54%	52%	56%	56%	50%	59%	69%	69%
Alishaa'	56%	53%	57%	56%	69%	55%	93%	61%
Nahawnd	61%	55%	61%	61%	89%	57%	89%	89%
Qersh	50%	59%	52%	54%	57%	57%	57%	57%
Asia	92%	52%	92%	70%	94%	73%	92%	92%
Shir (Shi3r)	92%	54%	82%	82%	77%	57%	73%	80%
Arafat	54%	54%	54%	54%	64%	53%	70%	70%
Cameron	61%	61%	61%	61%	91%	52%	56%	59%
Maliki	50%	36%	64%	69%	39%	38%	47%	47%
Baqara	57%	70%	58%	58%	55%	56%	54%	54%
Sakhr	43%	39%	41%	43%	57%	41%	60%	60%
Iqteran	69%	51%	58%	58%	75%	58%	75%	78%
Aziz	62%	67%	59%	60%	74%	60%	74%	74%
Ahram	56%	52%	56%	56%	79%	53%	87%	87%
Jalamah	51%	52%	51%	51%	67%	62%	67%	67%
Malek Abdullah	83%	93%	93%	92%	97%	92%	93%	92%
Jamaa Arabiya	56%	62%	56%	56%	62%	91%	94%	91%
Sakakini	75%	52%	51%	59%	90%	65%	90%	81%
Bursa	73%	73%	73%	73%	96%	59%	95%	89%
Ain	56%	56%	56%	56%	54%	52%	74%	74%
Thaheryah	55%	61%	55%	55%	61%	53%	94%	94%
Tarablus	51%	58%	58%	58%	58%	59%	58%	58%
Azhar	58%	54%	52%	58%	65%	56%	54%	54%
Arabiyah	52%	52%	58%	59%	55%	55%	54%	55%
Asad	43%	42%	53%	53%	51%	36%	63%	47%
Athraa'	63%	63%	63%	64%	92%	56%	94%	94%
Qedra	63%	53%	89%	88%	53%	56%	53%	53%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	96%	56%	96%	96%	52%	55%	52%	51%
Adnan Ibrahim	64%	63%	63%	63%	50%	59%	67%	70%
Amman	97%	69%	71%	71%	51%	63%	62%	59%
Jarrar	99%	54%	69%	66%	78%	73%	77%	80%
Alishaa'	99%	58%	99%	99%	51%	56%	51%	51%
Nahawnd	64%	56%	61%	61%	56%	50%	75%	77%
Qersh	59%	53%	94%	60%	86%	51%	80%	71%
Asia	100%	50%	99%	99%	56%	65%	97%	97%
Shir (Shi3r)	81%	73%	81%	83%	92%	61%	92%	92%
Arafat	97%	58%	77%	96%	52%	52%	51%	56%
Cameron	92%	62%	93%	61%	61%	60%	59%	60%
Maliki	62%	44%	63%	63%	55%	37%	55%	75%
Baqara	58%	58%	67%	58%	58%	78%	80%	80%
Sakhr	71%	48%	69%	62%	65%	47%	59%	61%
Iqteran	63%	58%	79%	64%	77%	52%	79%	75%
Aziz	75%	60%	91%	61%	69%	57%	66%	66%
Ahram	66%	55%	56%	56%	67%	51%	67%	67%
Jalamah	53%	74%	67%	67%	51%	72%	51%	51%
Malek Abdullah	99%	96%	100%	97%	72%	51%	75%	76%
Jamaa Arabiya	61%	97%	56%	56%	65%	61%	74%	72%
Sakakini	90%	65%	90%	85%	91%	82%	88%	91%
Bursa	99%	94%	72%	86%	80%	67%	70%	69%
Ain	92%	56%	62%	62%	73%	64%	73%	74%
Thaheryah	100%	63%	100%	100%	62%	63%	82%	65%
Tarablus	59%	59%	59%	51%	59%	52%	55%	53%
Azhar	51%	54%	86%	86%	51%	66%	58%	58%
Arabiyah	61%	52%	58%	58%	63%	56%	64%	64%
Asad	41%	43%	43%	95%	42%	48%	37%	37%
Athraa'	62%	62%	63%	63%	58%	66%	69%	66%
Qedra	53%	53%	99%	99%	66%	53%	67%	68%

Table C.17: Per level and query weighted precision when using BRF benchmarks for Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	53%	53%	53%	53%	82%	77%	87%	88%
Adnan Ibrahim	75%	71%	75%	75%	58%	77%	61%	50%
Amman	79%	77%	79%	79%	50%	76%	80%	80%
Jarrar	76%	76%	77%	77%	50%	77%	81%	81%
Alishaa'	70%	76%	77%	77%	81%	76%	93%	78%
Nahawnd	64%	59%	64%	64%	90%	71%	90%	90%
Qersh	50%	73%	63%	68%	65%	65%	65%	65%
Asia	93%	76%	93%	77%	95%	82%	93%	93%
Shir (Shi3r)	92%	76%	87%	87%	84%	77%	82%	86%
Arafat	76%	76%	76%	76%	79%	76%	81%	81%
Cameron	72%	72%	72%	72%	91%	63%	64%	70%
Maliki	71%	44%	74%	84%	48%	46%	59%	59%
Baqara	77%	81%	77%	77%	59%	77%	61%	61%
Sakhr	56%	59%	53%	53%	52%	52%	42%	42%
Iqteran	81%	53%	77%	77%	83%	77%	83%	85%
Aziz	64%	68%	61%	62%	83%	78%	83%	83%
Ahram	62%	76%	62%	62%	85%	76%	90%	90%
Jalamah	52%	76%	52%	52%	80%	78%	80%	80%
Malek Abdullah	83%	94%	94%	93%	97%	93%	94%	93%
Jamaa Arabiya	62%	78%	62%	62%	72%	92%	95%	92%
Sakakini	83%	76%	54%	77%	92%	79%	92%	86%
Bursa	77%	77%	77%	77%	96%	73%	95%	91%
Ain	77%	77%	77%	77%	61%	63%	83%	83%
Thaheryah	63%	78%	63%	63%	64%	76%	94%	94%
Tarablus	51%	77%	77%	77%	69%	70%	69%	69%
Azhar	72%	76%	56%	77%	70%	77%	58%	58%
Arabiyah	76%	76%	67%	77%	69%	69%	64%	69%
Asad	53%	37%	61%	61%	42%	55%	63%	51%
Athraa'	75%	75%	75%	79%	92%	77%	94%	94%
Qedra	79%	76%	89%	88%	59%	77%	59%	59%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	96%	62%	96%	96%	76%	76%	76%	59%
Adnan Ibrahim	76%	75%	75%	75%	50%	70%	68%	72%
Amman	97%	77%	78%	78%	75%	75%	66%	65%
Jarrar	99%	64%	73%	71%	85%	82%	84%	86%
Alishaa'	99%	77%	99%	99%	59%	77%	59%	59%
Nahawnd	69%	77%	64%	64%	67%	50%	82%	83%
Qersh	68%	76%	94%	74%	87%	75%	83%	80%
Asia	100%	50%	99%	99%	77%	77%	97%	97%
Shir (Shi3r)	86%	82%	86%	87%	93%	78%	93%	93%
Arafat	97%	77%	84%	96%	76%	76%	55%	67%
Cameron	92%	72%	93%	72%	74%	69%	73%	74%
Maliki	62%	69%	62%	62%	75%	78%	78%	83%
Baqara	77%	77%	73%	77%	72%	85%	84%	84%
Sakhr	73%	45%	72%	43%	67%	61%	43%	48%
Iqteran	75%	77%	85%	79%	81%	53%	85%	75%
Aziz	83%	62%	92%	63%	72%	62%	67%	67%
Ahram	72%	76%	62%	62%	80%	75%	80%	80%
Jalamah	53%	83%	80%	80%	52%	82%	52%	52%
Malek Abdullah	99%	96%	100%	97%	72%	51%	75%	76%
Jamaa Arabiya	74%	97%	62%	62%	79%	78%	83%	82%
Sakakini	92%	79%	92%	88%	92%	87%	90%	92%
Bursa	99%	94%	77%	87%	84%	74%	79%	79%
Ain	92%	77%	75%	75%	82%	79%	82%	83%
Thaheryah	100%	79%	100%	100%	78%	79%	87%	79%
Tarablus	64%	70%	70%	51%	68%	56%	60%	62%
Azhar	53%	68%	86%	86%	75%	77%	72%	72%
Arabiyah	72%	53%	67%	67%	79%	70%	79%	79%
Asad	50%	53%	53%	95%	51%	58%	35%	35%
Athraa'	75%	75%	75%	75%	67%	67%	71%	67%
Qedra	59%	59%	99%	99%	80%	76%	80%	80%

C.3.5 Bing/BRF

Table C.18: Per level and query macro F-measure when using BRF benchmarks for Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	37%	38%	42%	42%	51%	42%	51%	51%
Adnan Ibrahim	49%	59%	56%	77%	39%	42%	76%	63%
Amman	85%	37%	59%	59%	40%	42%	40%	40%
Jarrar	42%	40%	42%	42%	46%	36%	42%	42%
Alishaa'	60%	38%	66%	66%	37%	40%	59%	66%
Nahawnd	49%	45%	55%	40%	95%	45%	93%	93%
Qersh	37%	35%	37%	37%	95%	44%	88%	88%
Asia	38%	38%	66%	52%	96%	60%	93%	93%
Shir (Shi3r)	68%	40%	63%	63%	42%	40%	42%	42%
Arafat	45%	44%	44%	44%	56%	47%	49%	49%
Cameron	39%	35%	39%	39%	61%	54%	62%	57%
Maliki	56%	22%	28%	30%	43%	27%	43%	43%
Baqara	42%	47%	55%	55%	51%	39%	71%	71%
Sakhr	52%	33%	48%	48%	48%	48%	46%	46%
Iqteran	51%	38%	45%	45%	87%	42%	87%	86%
Aziz	87%	76%	89%	89%	92%	90%	91%	91%
Ahram	49%	38%	49%	49%	57%	49%	57%	57%
Jalamah	49%	40%	55%	55%	58%	42%	62%	62%
Malek Abdullah	90%	81%	61%	64%	98%	100%	98%	98%
Jamaa Arabiya	40%	38%	94%	94%	87%	80%	92%	89%
Sakakini	54%	44%	38%	38%	87%	49%	88%	88%
Bursa	82%	38%	76%	71%	97%	44%	100%	100%
Ain	40%	38%	69%	66%	53%	36%	40%	40%
Thaheryah	51%	40%	42%	42%	84%	40%	47%	52%
Tarablus	65%	38%	47%	44%	88%	45%	99%	98%
Azhar	49%	47%	48%	48%	48%	44%	48%	48%
Arabiyah	63%	38%	73%	73%	91%	40%	66%	66%
Asad	83%	26%	31%	30%	63%	41%	40%	60%
Athraa'	100%	38%	100%	100%	97%	40%	97%	97%
Qedra	99%	37%	86%	87%	78%	40%	63%	63%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	90%	42%	90%	90%	68%	39%	48%	48%
Adnan Ibrahim	49%	62%	64%	57%	39%	39%	49%	49%
Amman	92%	42%	92%	92%	59%	44%	59%	60%
Jarrar	42%	42%	42%	42%	54%	37%	54%	55%
Alishaa'	98%	45%	76%	76%	95%	59%	79%	78%
Nahawnd	96%	54%	97%	97%	36%	59%	36%	36%
Qersh	97%	39%	97%	97%	42%	36%	49%	51%
Asia	100%	58%	100%	77%	67%	73%	90%	89%
Shir (Shi3r)	96%	42%	75%	75%	70%	38%	68%	68%
Arafat	45%	44%	45%	45%	86%	52%	87%	87%
Cameron	62%	56%	63%	57%	64%	63%	37%	64%
Maliki	98%	25%	98%	98%	43%	33%	44%	55%
Baqara	56%	40%	54%	54%	56%	40%	51%	56%
Sakhr	49%	42%	46%	46%	71%	35%	73%	68%
Iqteran	90%	52%	88%	88%	78%	60%	51%	79%
Aziz	97%	92%	93%	93%	82%	57%	73%	74%
Ahram	99%	49%	99%	99%	58%	39%	62%	62%
Jalamah	67%	44%	63%	63%	42%	44%	40%	40%
Malek Abdullah	98%	100%	98%	98%	49%	39%	74%	61%
Jamaa Arabiya	95%	91%	97%	97%	76%	36%	74%	86%
Sakakini	44%	47%	70%	64%	53%	45%	57%	57%
Bursa	94%	87%	96%	95%	37%	36%	37%	38%
Ain	45%	40%	45%	49%	67%	51%	66%	68%
Thaheryah	94%	52%	48%	48%	81%	47%	38%	80%
Tarablus	99%	70%	99%	99%	36%	44%	36%	36%
Azhar	53%	37%	53%	53%	54%	45%	54%	47%
Arabiyah	92%	39%	66%	66%	58%	44%	63%	69%
Asad	42%	34%	76%	76%	63%	21%	81%	43%
Athraa'	100%	47%	100%	100%	81%	38%	85%	85%
Qedra	100%	45%	85%	85%	52%	36%	50%	47%

Table C.19: Per level and query weighted recall when using BRF benchmarks for Bing.

	t	**t**	**t**	**t**	**s**	**s**	**s**	**s**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	50%	52%	52%	52%	59%	54%	59%	59%
Adnan Ibrahim	58%	64%	62%	78%	52%	54%	77%	67%
Amman	85%	50%	64%	64%	52%	54%	50%	50%
Jarrar	53%	52%	53%	53%	55%	52%	53%	53%
Alishaa’	65%	52%	69%	69%	51%	53%	64%	69%
Nahawnd	54%	56%	55%	51%	95%	56%	93%	93%
Qersh	50%	50%	50%	50%	95%	55%	88%	88%
Asia	50%	50%	69%	60%	96%	65%	93%	93%
Shir (Shi3r)	71%	53%	67%	67%	54%	53%	54%	54%
Arafat	55%	55%	55%	55%	62%	57%	58%	58%
Cameron	52%	50%	52%	52%	64%	60%	65%	62%
Maliki	62%	35%	38%	40%	50%	38%	55%	55%
Baqara	54%	57%	60%	60%	51%	52%	73%	73%
Sakhr	57%	41%	53%	53%	60%	59%	59%	59%
Iqteran	59%	52%	56%	56%	87%	54%	87%	86%
Aziz	87%	77%	89%	89%	92%	90%	91%	91%
Ahram	58%	52%	58%	58%	63%	58%	63%	63%
Jalamah	56%	53%	60%	60%	62%	54%	66%	66%
Malek Abdullah	90%	82%	61%	65%	98%	100%	98%	98%
Jamaa Arabiya	53%	52%	94%	94%	87%	81%	92%	89%
Sakakini	61%	55%	50%	50%	87%	58%	88%	88%
Bursa	83%	52%	77%	73%	97%	55%	100%	100%
Ain	52%	52%	69%	69%	60%	51%	53%	53%
Thaheryah	59%	53%	54%	54%	84%	53%	56%	60%
Tarablus	65%	52%	57%	55%	88%	56%	99%	98%
Azhar	53%	56%	57%	57%	53%	55%	53%	53%
Arabiyah	67%	52%	74%	74%	91%	52%	69%	69%
Asad	83%	37%	40%	42%	64%	47%	50%	65%
Athraa’	100%	52%	100%	100%	97%	53%	97%	97%
Qedra	99%	51%	86%	87%	79%	53%	67%	67%

	t w/ s	**t w/ s**	**t w/ s**	**t w/ s**	**ip**	**ip**	**ip**	**ip**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	90%	54%	90%	90%	69%	51%	57%	57%
Adnan Ibrahim	58%	66%	68%	62%	50%	52%	53%	54%
Amman	92%	54%	92%	92%	62%	55%	62%	62%
Jarrar	53%	53%	53%	53%	58%	53%	58%	59%
Alishaa’	98%	55%	77%	77%	95%	64%	80%	79%
Nahawnd	96%	61%	97%	97%	51%	64%	51%	51%
Qersh	97%	52%	97%	97%	52%	51%	58%	59%
Asia	100%	63%	100%	78%	70%	75%	90%	89%
Shir (Shi3r)	96%	54%	76%	76%	72%	52%	71%	71%
Arafat	56%	55%	56%	56%	86%	60%	87%	87%
Cameron	65%	62%	66%	62%	68%	67%	50%	68%
Maliki	98%	38%	98%	98%	52%	43%	53%	58%
Baqara	58%	53%	60%	60%	58%	53%	54%	59%
Sakhr	61%	53%	59%	59%	71%	45%	73%	68%
Iqteran	90%	60%	88%	88%	79%	65%	59%	80%
Aziz	97%	92%	93%	93%	83%	63%	73%	74%
Ahram	99%	58%	99%	99%	62%	52%	65%	65%
Jalamah	70%	55%	67%	67%	54%	55%	53%	53%
Malek Abdullah	98%	100%	98%	98%	57%	51%	75%	62%
Jamaa Arabiya	95%	91%	97%	97%	77%	51%	75%	86%
Sakakini	55%	57%	72%	68%	56%	56%	59%	59%
Bursa	94%	87%	96%	95%	50%	51%	50%	52%
Ain	53%	53%	53%	58%	70%	59%	69%	71%
Thaheryah	94%	60%	57%	57%	81%	57%	52%	80%
Tarablus	99%	72%	99%	99%	51%	53%	51%	51%
Azhar	54%	51%	55%	55%	55%	54%	57%	52%
Arabiyah	92%	51%	69%	69%	63%	55%	67%	71%
Asad	48%	43%	79%	79%	64%	35%	82%	51%
Athraa’	100%	57%	100%	100%	82%	52%	85%	85%
Qedra	100%	56%	85%	85%	57%	51%	56%	54%

Table C.20: Per level and query weighted precision when using BRF benchmarks for Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	50%	76%	57%	57%	77%	76%	77%	77%
Adnan Ibrahim	77%	79%	78%	85%	63%	76%	82%	80%
Amman	88%	50%	76%	76%	59%	76%	50%	50%
Jarrar	77%	76%	77%	77%	77%	27%	77%	77%
Alishaa'	79%	76%	81%	81%	59%	76%	76%	81%
Nahawnd	56%	77%	55%	54%	95%	77%	94%	94%
Qersh	50%	50%	50%	50%	95%	76%	89%	89%
Asia	50%	50%	81%	78%	96%	79%	94%	94%
Shir (Shi3r)	82%	76%	80%	80%	76%	76%	76%	76%
Arafat	69%	76%	76%	76%	75%	77%	77%	77%
Cameron	63%	50%	63%	63%	69%	71%	73%	72%
Maliki	82%	53%	49%	53%	51%	56%	40%	40%
Baqara	76%	77%	69%	69%	51%	63%	82%	82%
Sakhr	63%	60%	62%	62%	42%	44%	41%	41%
Iqteran	77%	76%	77%	77%	90%	76%	90%	89%
Aziz	90%	84%	91%	91%	93%	92%	92%	92%
Ahram	77%	76%	77%	77%	79%	77%	79%	79%
Jalamah	64%	76%	69%	69%	69%	76%	80%	80%
Malek Abdullah	92%	87%	61%	68%	98%	100%	98%	98%
Jamaa Arabiya	76%	76%	95%	95%	90%	86%	93%	91%
Sakakini	78%	76%	50%	50%	90%	77%	90%	90%
Bursa	87%	76%	84%	82%	97%	76%	100%	100%
Ain	59%	76%	69%	81%	74%	75%	76%	76%
Thaheryah	77%	76%	76%	76%	87%	76%	70%	78%
Tarablus	65%	76%	77%	76%	90%	77%	99%	98%
Azhar	54%	70%	71%	71%	55%	76%	55%	55%
Arabiyah	80%	76%	79%	79%	91%	59%	81%	81%
Asad	86%	53%	52%	36%	83%	77%	44%	82%
Athraa'	100%	76%	100%	100%	97%	76%	97%	97%
Qedra	99%	59%	89%	90%	85%	76%	80%	80%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	91%	76%	91%	91%	72%	55%	71%	71%
Adnan Ibrahim	77%	80%	80%	72%	50%	63%	55%	56%
Amman	93%	76%	93%	93%	66%	76%	66%	66%
Jarrar	77%	77%	77%	77%	67%	28%	67%	69%
Alishaa'	98%	69%	83%	83%	95%	79%	83%	83%
Nahawnd	96%	78%	97%	97%	75%	76%	75%	75%
Qersh	97%	63%	97%	97%	57%	75%	77%	77%
Asia	100%	73%	100%	85%	81%	83%	92%	91%
Shir (Shi3r)	96%	76%	84%	84%	82%	76%	82%	82%
Arafat	77%	76%	77%	77%	88%	78%	88%	88%
Cameron	73%	75%	75%	72%	80%	80%	50%	80%
Maliki	98%	44%	98%	98%	46%	44%	47%	69%
Baqara	60%	76%	71%	71%	60%	76%	55%	62%
Sakhr	43%	41%	41%	41%	78%	46%	79%	74%
Iqteran	92%	78%	90%	90%	85%	79%	77%	86%
Aziz	97%	93%	94%	94%	87%	79%	73%	74%
Ahram	99%	77%	99%	99%	69%	63%	71%	71%
Jalamah	81%	76%	80%	80%	76%	76%	76%	76%
Malek Abdullah	98%	100%	98%	98%	68%	55%	79%	63%
Jamaa Arabiya	95%	92%	97%	97%	84%	75%	78%	89%
Sakakini	76%	77%	82%	80%	58%	77%	61%	61%
Bursa	95%	90%	96%	95%	50%	75%	50%	76%
Ain	58%	76%	58%	77%	81%	77%	81%	82%
Thaheryah	95%	78%	71%	71%	82%	77%	76%	82%
Tarablus	99%	82%	99%	99%	75%	59%	75%	75%
Azhar	54%	59%	56%	56%	56%	61%	59%	53%
Arabiyah	93%	55%	81%	81%	75%	76%	80%	80%
Asad	46%	58%	83%	83%	78%	61%	88%	75%
Athraa'	100%	77%	100%	100%	87%	76%	88%	88%
Qedra	100%	77%	88%	88%	61%	75%	61%	58%

C.3.6 Google and Bing/BRF

Table C.21: Per level and query macro F-measure when using BRF benchmarks for both.

	t	t	t	t	s	s	s	s	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	62%	40%	65%	65%	92%	39%	92%	92%	73%	38%	73%	73%	66%	43%	68%	68%
Adnan Ibrahim	75%	48%	79%	68%	79%	41%	50%	48%	84%	46%	66%	56%	56%	55%	52%	60%
Amman	56%	56%	56%	56%	68%	38%	57%	45%	90%	52%	56%	56%	52%	47%	52%	52%
Jarrar	61%	59%	61%	61%	41%	64%	41%	39%	45%	62%	45%	45%	66%	64%	90%	90%
Alishaa’	68%	38%	42%	42%	52%	42%	52%	52%	60%	42%	55%	55%	37%	53%	37%	37%
Nahawnd	50%	36%	41%	39%	95%	40%	48%	48%	90%	37%	86%	50%	68%	34%	62%	63%
Qersh	86%	38%	86%	80%	90%	44%	91%	91%	98%	45%	96%	96%	34%	41%	52%	52%
Asia	54%	54%	54%	54%	94%	44%	92%	93%	100%	54%	55%	54%	77%	54%	94%	94%
Shir (Shi3r)	95%	64%	93%	64%	58%	55%	60%	59%	98%	55%	99%	64%	90%	70%	57%	57%
Arafat	43%	43%	43%	43%	91%	47%	52%	91%	99%	47%	99%	99%	72%	45%	82%	82%
Cameron	48%	41%	44%	48%	71%	62%	68%	66%	71%	67%	54%	66%	89%	61%	90%	90%
Maliki	45%	21%	54%	56%	92%	34%	79%	79%	99%	25%	98%	98%	71%	50%	67%	76%
Baqara	85%	57%	84%	84%	60%	46%	42%	42%	95%	52%	94%	94%	50%	63%	64%	63%
Sakhr	30%	32%	50%	50%	64%	41%	48%	48%	67%	58%	64%	64%	53%	40%	57%	56%
Iqteran	46%	43%	56%	72%	46%	45%	88%	88%	66%	59%	64%	64%	80%	77%	86%	84%
Aziz	49%	49%	89%	88%	56%	46%	62%	93%	100%	55%	99%	99%	79%	63%	97%	98%
Ahram	43%	43%	47%	47%	92%	51%	86%	86%	99%	48%	99%	99%	66%	48%	67%	67%
Jalamah	75%	43%	67%	66%	54%	42%	85%	85%	93%	58%	71%	71%	79%	38%	80%	80%
Malek Abdullah	86%	59%	69%	59%	98%	98%	97%	95%	99%	98%	99%	98%	45%	43%	92%	95%
Jamaa Arabiya	86%	52%	90%	90%	86%	85%	91%	90%	97%	97%	96%	98%	73%	84%	88%	88%
Sakakini	58%	39%	74%	58%	68%	50%	61%	61%	68%	50%	64%	64%	57%	69%	57%	57%
Bursa	71%	57%	57%	57%	96%	81%	94%	95%	97%	84%	93%	95%	93%	53%	94%	93%
Ain	49%	49%	43%	43%	77%	41%	86%	86%	47%	48%	89%	86%	72%	51%	70%	70%
Thaheryah	53%	38%	47%	47%	60%	50%	51%	51%	98%	46%	53%	53%	82%	48%	81%	81%
Tarablus	80%	43%	45%	45%	91%	41%	93%	93%	99%	44%	45%	45%	41%	41%	47%	48%
Azhar	39%	38%	39%	44%	75%	40%	70%	70%	72%	40%	72%	72%	57%	53%	58%	59%
Arabiyah	91%	54%	57%	54%	89%	43%	78%	78%	57%	54%	94%	86%	36%	41%	36%	36%
Asad	60%	21%	50%	57%	84%	30%	56%	58%	96%	32%	96%	96%	52%	39%	52%	52%
Athraa’	88%	56%	54%	94%	97%	39%	97%	97%	99%	59%	100%	100%	91%	54%	91%	91%
Qedra	94%	44%	40%	40%	97%	43%	96%	96%	99%	40%	99%	99%	77%	41%	79%	93%

Table C.22: Per level and query weighted recall when using BRF benchmarks for both.

	t	**t**	**t**	**t**	**s**	**s**	**s**	**s**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	67%	51%	69%	69%	92%	53%	92%	92%
Adnan Ibrahim	77%	58%	80%	71%	79%	52%	56%	54%
Amman	62%	62%	62%	62%	71%	52%	63%	56%
Jarrar	66%	63%	66%	66%	53%	68%	53%	51%
Alishaa'	71%	52%	54%	54%	60%	54%	60%	60%
Nahawnd	55%	51%	51%	52%	95%	53%	57%	57%
Qersh	87%	52%	87%	81%	91%	55%	92%	92%
Asia	60%	60%	60%	60%	95%	55%	93%	93%
Shir (Shi3r)	96%	68%	93%	68%	64%	62%	65%	64%
Arafat	55%	55%	55%	55%	92%	56%	60%	91%
Cameron	56%	54%	54%	56%	73%	66%	71%	69%
Maliki	55%	35%	65%	64%	92%	42%	80%	80%
Baqara	85%	63%	85%	85%	64%	56%	53%	53%
Sakhr	37%	38%	51%	51%	70%	47%	60%	60%
Iqteran	55%	55%	61%	74%	57%	56%	88%	88%
Aziz	55%	55%	90%	89%	62%	57%	67%	94%
Ahram	53%	55%	57%	57%	92%	59%	86%	86%
Jalamah	76%	55%	70%	70%	61%	54%	86%	86%
Malek Abdullah	86%	62%	69%	61%	98%	98%	98%	96%
Jamaa Arabiya	87%	60%	91%	90%	86%	86%	92%	91%
Sakakini	64%	51%	76%	64%	71%	59%	66%	66%
Bursa	72%	61%	61%	61%	97%	82%	95%	95%
Ain	57%	57%	55%	55%	79%	54%	86%	86%
Thaheryah	61%	52%	57%	57%	65%	59%	59%	59%
Tarablus	81%	55%	51%	51%	92%	54%	93%	93%
Azhar	51%	52%	51%	53%	76%	53%	72%	72%
Arabiyah	91%	61%	63%	61%	90%	55%	79%	79%
Asad	63%	36%	61%	61%	84%	39%	62%	63%
Athraa'	89%	62%	59%	94%	98%	52%	98%	98%
Qedra	95%	55%	52%	52%	97%	55%	97%	97%

	t w/ s	**t w/ s**	**t w/ s**	**t w/ s**	**ip**	**ip**	**ip**	**ip**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	75%	52%	75%	75%	70%	55%	71%	71%
Adnan Ibrahim	85%	54%	67%	58%	59%	59%	52%	61%
Amman	91%	59%	61%	61%	58%	57%	58%	58%
Jarrar	55%	65%	55%	55%	69%	68%	90%	90%
Alishaa'	65%	54%	62%	62%	52%	61%	52%	52%
Nahawnd	91%	51%	86%	55%	71%	51%	67%	67%
Qersh	98%	56%	97%	97%	51%	53%	59%	59%
Asia	100%	61%	61%	60%	79%	61%	95%	94%
Shir (Shi3r)	99%	62%	99%	68%	91%	72%	63%	63%
Arafat	99%	56%	99%	99%	74%	56%	83%	83%
Cameron	73%	70%	61%	69%	90%	66%	90%	90%
Maliki	99%	37%	98%	98%	71%	56%	66%	76%
Baqara	95%	59%	95%	95%	54%	67%	68%	67%
Sakhr	70%	60%	67%	67%	56%	44%	61%	60%
Iqteran	70%	64%	68%	68%	81%	78%	86%	85%
Aziz	100%	58%	99%	99%	80%	67%	98%	99%
Ahram	99%	58%	99%	99%	69%	58%	69%	69%
Jalamah	93%	64%	74%	74%	80%	52%	81%	81%
Malek Abdullah	99%	99%	99%	98%	56%	50%	93%	95%
Jamaa Arabiya	98%	98%	97%	98%	75%	85%	88%	88%
Sakakini	71%	59%	68%	68%	58%	72%	58%	58%
Bursa	97%	85%	94%	96%	93%	59%	94%	94%
Ain	56%	57%	90%	87%	74%	59%	73%	73%
Thaheryah	98%	57%	60%	60%	82%	57%	82%	82%
Tarablus	99%	55%	51%	51%	54%	51%	51%	52%
Azhar	74%	53%	73%	73%	62%	60%	60%	61%
Arabiyah	63%	61%	95%	86%	51%	54%	51%	51%
Asad	96%	43%	96%	96%	63%	48%	63%	63%
Athraa'	100%	64%	100%	100%	92%	58%	91%	91%
Qedra	99%	52%	100%	100%	79%	52%	80%	93%

Table C.23: Per level and query weighted precision when using BRF benchmarks for both.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	80%	54%	81%	81%	93%	76%	93%	93%
Adnan Ibrahim	83%	77%	81%	81%	81%	55%	60%	57%
Amman	72%	72%	72%	72%	81%	76%	79%	77%
Jarrar	80%	70%	80%	80%	55%	80%	55%	49%
Alishaa'	82%	76%	76%	76%	73%	76%	78%	78%
Nahawnd	59%	75%	53%	59%	95%	76%	71%	71%
Qersh	89%	76%	89%	85%	91%	72%	92%	92%
Asia	70%	70%	70%	70%	95%	76%	93%	94%
Shir (Shi3r)	96%	80%	94%	80%	79%	78%	79%	79%
Arafat	76%	76%	76%	76%	92%	66%	78%	91%
Cameron	65%	76%	65%	65%	77%	77%	79%	78%
Maliki	45%	31%	49%	66%	94%	68%	87%	87%
Baqara	88%	79%	87%	87%	74%	70%	59%	59%
Sakhr	54%	57%	62%	62%	82%	80%	43%	43%
Iqteran	64%	76%	72%	82%	77%	76%	89%	89%
Aziz	58%	58%	91%	91%	78%	77%	80%	94%
Ahram	60%	76%	77%	77%	92%	77%	87%	87%
Jalamah	79%	76%	81%	81%	72%	67%	89%	89%
Malek Abdullah	89%	66%	70%	63%	98%	98%	98%	96%
Jamaa Arabiya	89%	78%	92%	92%	89%	88%	93%	92%
Sakakini	79%	52%	84%	79%	82%	77%	80%	80%
Bursa	77%	69%	69%	69%	97%	86%	95%	95%
Ain	69%	69%	76%	76%	85%	76%	89%	89%
Thaheryah	78%	76%	77%	77%	75%	75%	75%	75%
Tarablus	85%	76%	51%	51%	93%	76%	93%	93%
Azhar	52%	61%	52%	57%	82%	65%	79%	79%
Arabiyah	92%	76%	79%	78%	90%	76%	85%	85%
Asad	66%	44%	46%	65%	89%	60%	61%	64%
Athraa'	90%	78%	64%	94%	98%	59%	98%	98%
Qedra	95%	76%	59%	59%	97%	76%	97%	97%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amazon	81%	61%	81%	81%	81%	76%	81%	81%
Adnan Ibrahim	87%	57%	68%	60%	60%	65%	52%	63%
Amman	91%	70%	72%	72%	66%	71%	66%	66%
Jarrar	65%	69%	65%	65%	80%	80%	92%	92%
Alishaa'	79%	76%	78%	78%	75%	78%	75%	75%
Nahawnd	91%	54%	87%	59%	81%	75%	80%	80%
Qersh	98%	76%	97%	97%	75%	62%	71%	71%
Asia	100%	72%	71%	70%	85%	78%	95%	95%
Shir (Shi3r)	99%	78%	99%	80%	92%	82%	79%	79%
Arafat	99%	66%	99%	99%	79%	77%	87%	87%
Cameron	79%	80%	78%	78%	90%	78%	90%	90%
Maliki	99%	60%	98%	98%	80%	81%	77%	85%
Baqara	95%	71%	95%	95%	55%	77%	78%	77%
Sakhr	73%	82%	78%	69%	66%	62%	68%	68%
Iqteran	81%	79%	80%	80%	86%	85%	89%	88%
Aziz	100%	60%	99%	99%	84%	80%	98%	99%
Ahram	99%	77%	99%	99%	74%	77%	73%	73%
Jalamah	94%	79%	83%	83%	86%	76%	86%	86%
Malek Abdullah	99%	99%	99%	98%	77%	50%	93%	95%
Jamaa Arabiya	98%	98%	97%	98%	83%	88%	90%	90%
Sakakini	82%	77%	80%	80%	58%	82%	58%	58%
Bursa	97%	86%	94%	96%	94%	69%	94%	94%
Ain	68%	69%	90%	88%	83%	77%	82%	82%
Thaheryah	98%	77%	76%	76%	84%	74%	84%	84%
Tarablus	99%	76%	51%	51%	76%	53%	51%	53%
Azhar	81%	65%	77%	77%	70%	74%	61%	62%
Arabiyah	77%	76%	95%	89%	75%	70%	75%	75%
Asad	96%	44%	96%	96%	47%	43%	47%	47%
Athraa'	100%	74%	100%	100%	93%	63%	92%	92%
Qedra	99%	59%	100%	100%	85%	55%	85%	93%

C.3.7 Google/Plain

Table C.24: Per level and query macro F-measure when using plain benchmarks for Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	55%	55%	52%	58%	61%	60%	55%	55%
Arafat	70%	70%	67%	50%	76%	52%	61%	61%
Maliki	70%	68%	70%	70%	77%	44%	48%	48%
Sakhr	71%	76%	67%	67%	70%	58%	58%	73%
Jamaa Arabiya	57%	54%	45%	52%	61%	76%	72%	76%
Thaheryah	67%	70%	67%	67%	54%	55%	64%	64%
Tarablus	57%	75%	57%	57%	44%	47%	69%	69%
Arabiyah	76%	68%	72%	68%	82%	42%	82%	82%
Asad	68%	71%	86%	83%	61%	69%	59%	57%
Athraa'	61%	61%	61%	61%	92%	49%	87%	58%
Qedra	67%	70%	72%	72%	90%	49%	72%	72%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	76%	49%	76%	76%	56%	49%	60%	49%
Arafat	70%	70%	70%	70%	59%	70%	70%	70%
Maliki	85%	43%	51%	65%	66%	66%	66%	66%
Sakhr	77%	58%	73%	73%	66%	71%	69%	58%
Jamaa Arabiya	63%	75%	77%	77%	34%	34%	34%	34%
Thaheryah	48%	71%	73%	73%	86%	61%	51%	51%
Tarablus	52%	47%	52%	52%	58%	50%	58%	58%
Arabiyah	76%	71%	76%	76%	56%	70%	56%	51%
Asad	71%	49%	76%	61%	72%	72%	72%	72%
Athraa'	95%	59%	61%	61%	48%	53%	48%	48%
Qedra	61%	56%	61%	67%	61%	61%	61%	61%

Table C.25: Per level and query weighted recall when using plain benchmarks for Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	55%	55%	47%	59%	47%	56%	57%	57%
Arafat	74%	74%	51%	44%	74%	50%	58%	58%
Maliki	75%	63%	75%	75%	79%	41%	47%	46%
Sakhr	71%	79%	68%	68%	76%	71%	71%	61%
Jamaa Arabiya	53%	46%	33%	38%	45%	62%	57%	62%
Thaheryah	66%	77%	66%	66%	52%	44%	61%	61%
Tarablus	50%	79%	50%	50%	51%	48%	68%	68%
Arabiyah	76%	61%	63%	57%	72%	57%	72%	72%
Asad	60%	64%	86%	82%	51%	60%	50%	48%
Athraa'	66%	66%	66%	66%	92%	37%	83%	40%
Qedra	67%	66%	71%	71%	82%	44%	63%	63%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	63%	63%	63%	63%	67%	63%	67%	63%
Arafat	74%	74%	74%	74%	54%	79%	79%	79%
Maliki	89%	36%	47%	56%	75%	75%	75%	75%
Sakhr	79%	71%	76%	76%	68%	74%	71%	71%
Jamaa Arabiya	64%	60%	62%	63%	51%	51%	51%	51%
Thaheryah	44%	74%	77%	77%	78%	70%	54%	54%
Tarablus	52%	31%	52%	52%	67%	52%	67%	67%
Arabiyah	76%	65%	76%	76%	62%	61%	62%	62%
Asad	65%	39%	72%	52%	66%	66%	66%	66%
Athraa'	92%	65%	66%	66%	58%	61%	58%	58%
Qedra	62%	46%	62%	67%	62%	62%	62%	62%

Table C.26: Per level and query weighted precision when using plain benchmarks for Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	79%	79%	82%	79%	87%	82%	72%	72%
Arafat	68%	68%	100%	58%	88%	78%	81%	81%
Maliki	65%	80%	65%	65%	80%	53%	74%	76%
Sakhr	79%	72%	71%	71%	74%	50%	50%	92%
Jamaa Arabiya	75%	81%	82%	90%	93%	100%	100%	100%
Thaheryah	73%	83%	73%	73%	85%	74%	70%	70%
Tarablus	68%	84%	68%	68%	39%	57%	85%	85%
Arabiyah	85%	85%	90%	89%	94%	33%	94%	94%
Asad	82%	82%	86%	86%	91%	92%	91%	91%
Athraa'	74%	74%	74%	74%	92%	81%	94%	100%
Qedra	83%	85%	85%	85%	100%	79%	88%	88%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	100%	40%	100%	100%	78%	40%	67%	40%
Arafat	68%	68%	68%	68%	82%	62%	62%	62%
Maliki	83%	54%	56%	80%	64%	64%	64%	64%
Sakhr	78%	50%	73%	71%	72%	71%	73%	50%
Jamaa Arabiya	81%	100%	100%	100%	26%	26%	26%	26%
Thaheryah	61%	75%	75%	75%	96%	54%	48%	48%
Tarablus	54%	100%	54%	54%	62%	49%	62%	62%
Arabiyah	85%	85%	85%	85%	63%	86%	63%	77%
Asad	78%	80%	80%	91%	79%	79%	79%	79%
Athraa'	100%	73%	74%	74%	65%	70%	65%	65%
Qedra	82%	85%	82%	83%	82%	82%	82%	82%

C.3.8 Bing/Plain

Table C.27: Per level and query macro F-measure when using plain benchmarks for Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	55%	54%	54%	54%	60%	62%	60%	60%
Jarrar	46%	56%	66%	66%	56%	70%	56%	56%
Alishaa'	55%	40%	56%	56%	52%	37%	58%	58%
Arafat	58%	59%	64%	64%	50%	47%	54%	54%
Maliki	72%	51%	72%	72%	61%	55%	61%	61%
Sakhr	71%	42%	69%	67%	63%	38%	31%	31%
Malek Abdullah	56%	54%	56%	53%	50%	51%	57%	56%
Jamaa Arabiya	61%	65%	67%	73%	62%	52%	73%	68%
Bursa	56%	43%	56%	56%	54%	44%	54%	54%
Thaheryah	50%	50%	53%	53%	66%	58%	66%	66%
Tarablus	44%	52%	42%	42%	44%	49%	44%	44%
Arabiyah	79%	84%	81%	81%	57%	62%	63%	59%
Asad	57%	41%	68%	68%	85%	38%	71%	42%
Athraa'	41%	54%	54%	54%	94%	51%	54%	54%
Qedra	71%	73%	60%	60%	65%	52%	65%	65%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	64%	60%	66%	66%	54%	54%	54%	54%
Jarrar	72%	62%	68%	68%	64%	62%	49%	48%
Alishaa'	58%	39%	57%	57%	70%	63%	68%	64%
Arafat	92%	45%	51%	51%	83%	59%	82%	83%
Maliki	84%	52%	84%	84%	62%	50%	63%	64%
Sakhr	73%	38%	85%	73%	44%	32%	40%	36%
Malek Abdullah	54%	55%	65%	62%	58%	55%	53%	59%
Jamaa Arabiya	62%	67%	73%	60%	83%	85%	90%	93%
Bursa	46%	40%	46%	46%	74%	42%	74%	75%
Thaheryah	53%	43%	53%	53%	91%	46%	89%	76%
Tarablus	58%	43%	58%	58%	50%	53%	50%	50%
Arabiyah	87%	67%	81%	81%	75%	55%	71%	71%
Asad	72%	42%	74%	74%	60%	42%	60%	60%
Athraa'	100%	47%	100%	100%	74%	52%	84%	84%
Qedra	73%	70%	67%	67%	82%	52%	69%	69%

Table C.28: Per level and query weighted recall when using plain benchmarks for Bing.

	t	**t**	**t**	**t**	**s**	**s**	**s**	**s**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	45%	61%	43%	43%	49%	64%	49%	49%
Jarrar	31%	43%	55%	55%	43%	61%	43%	43%
Alishaa'	45%	44%	41%	41%	38%	37%	46%	46%
Arafat	41%	59%	64%	64%	34%	42%	47%	47%
Maliki	65%	61%	65%	65%	51%	56%	51%	51%
Sakhr	70%	50%	71%	69%	61%	50%	43%	43%
Malek Abdullah	53%	38%	43%	41%	46%	34%	42%	41%
Jamaa Arabiya	48%	49%	51%	59%	45%	42%	59%	53%
Bursa	49%	52%	49%	49%	38%	43%	38%	38%
Thaheryah	40%	50%	60%	60%	58%	54%	58%	58%
Tarablus	32%	52%	34%	34%	32%	44%	32%	32%
Arabiyah	66%	81%	69%	69%	43%	67%	48%	45%
Asad	43%	45%	57%	57%	84%	35%	78%	51%
Athraa'	28%	60%	37%	37%	94%	52%	37%	37%
Qedra	56%	70%	48%	48%	48%	62%	48%	48%

	t w/ s	**t w/ s**	**t w/ s**	**t w/ s**	**ip**	**ip**	**ip**	**ip**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	47%	61%	49%	49%	56%	67%	53%	67%
Jarrar	65%	48%	58%	58%	61%	67%	48%	47%
Alishaa'	41%	40%	40%	40%	71%	65%	69%	66%
Arafat	90%	34%	35%	35%	83%	55%	82%	83%
Maliki	77%	52%	77%	77%	67%	54%	69%	70%
Sakhr	70%	50%	83%	67%	46%	41%	43%	39%
Malek Abdullah	51%	39%	51%	48%	57%	54%	52%	57%
Jamaa Arabiya	45%	51%	58%	43%	83%	86%	90%	93%
Bursa	32%	37%	32%	32%	76%	52%	76%	77%
Thaheryah	52%	50%	52%	52%	91%	57%	89%	64%
Tarablus	42%	38%	42%	42%	51%	66%	51%	51%
Arabiyah	80%	58%	72%	72%	79%	56%	76%	76%
Asad	58%	35%	82%	82%	66%	51%	66%	66%
Athraa'	100%	44%	100%	100%	74%	59%	85%	85%
Qedra	57%	73%	50%	50%	80%	57%	57%	57%

Table C.29: Per level and query weighted precision when using plain benchmarks for Bing.

	t	**t**	**t**	**t**	**s**	**s**	**s**	**s**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	85%	62%	85%	85%	87%	81%	87%	87%
Jarrar	100%	83%	100%	100%	100%	87%	100%	100%
Alishaa'	82%	76%	88%	88%	87%	76%	85%	85%
Arafat	100%	80%	82%	82%	100%	78%	85%	85%
Maliki	92%	77%	92%	92%	86%	79%	86%	86%
Sakhr	88%	59%	87%	86%	85%	50%	55%	55%
Malek Abdullah	60%	91%	87%	89%	63%	100%	89%	93%
Jamaa Arabiya	85%	98%	98%	99%	100%	78%	100%	100%
Bursa	85%	76%	85%	85%	100%	77%	100%	100%
Thaheryah	81%	78%	79%	79%	86%	80%	86%	86%
Tarablus	81%	79%	57%	57%	85%	79%	85%	85%
Arabiyah	100%	91%	100%	100%	91%	58%	93%	90%
Asad	92%	78%	91%	91%	92%	79%	68%	59%
Athraa'	84%	79%	100%	100%	95%	79%	100%	100%
Qedra	100%	84%	83%	83%	100%	77%	100%	100%

	t w/ s	**t w/ s**	**t w/ s**	**t w/ s**	**ip**	**ip**	**ip**	**ip**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	100%	80%	100%	100%	53%	45%	62%	45%
Jarrar	100%	86%	100%	100%	68%	58%	50%	50%
Alishaa'	100%	78%	100%	100%	82%	81%	81%	80%
Arafat	95%	79%	100%	100%	87%	83%	87%	87%
Maliki	98%	79%	98%	98%	60%	66%	62%	65%
Sakhr	90%	50%	96%	91%	61%	43%	47%	51%
Malek Abdullah	69%	97%	92%	89%	59%	56%	84%	85%
Jamaa Arabiya	100%	100%	100%	100%	84%	89%	92%	94%
Bursa	85%	77%	85%	85%	83%	76%	83%	84%
Thaheryah	80%	78%	80%	80%	93%	55%	90%	94%
Tarablus	100%	77%	100%	100%	50%	44%	50%	50%
Arabiyah	97%	86%	97%	97%	75%	54%	70%	70%
Asad	96%	81%	69%	69%	63%	60%	63%	63%
Athraa'	100%	78%	100%	100%	76%	78%	88%	88%
Qedra	100%	83%	100%	100%	88%	52%	89%	89%

C.3.9 Google/Supervised Clustering/Human-annotated

Table C.30: Per level and query macro F-measure when using MBHA supervised approach/Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	62%	49%	20%	54%	36%	49%	36%	36%
Arafat	70%	70%	18%	21%	72%	91%	68%	89%
Maliki	81%	6%	19%	19%	89%	71%	92%	73%
Sakhr	76%	17%	61%	76%	17%	1%	10%	13%
Jamaa Arabiya	53%	53%	53%	53%	59%	42%	90%	82%
Thaheryah	43%	11%	34%	78%	71%	70%	85%	81%
Tarablus	50%	17%	50%	50%	85%	23%	79%	50%
Arabiyah	91%	42%	42%	42%	35%	35%	35%	35%
Asad	95%	91%	91%	91%	1%	85%	87%	72%
Athraa'	61%	35%	35%	35%	84%	32%	89%	85%
Qedra	30%	17%	30%	30%	77%	53%	77%	77%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	77%	60%	86%	52%	20%	49%	20%	20%
Arafat	100%	91%	95%	95%	74%	52%	76%	76%
Maliki	97%	44%	63%	92%	61%	15%	79%	76%
Sakhr	19%	80%	74%	80%	15%	71%	22%	22%
Jamaa Arabiya	83%	77%	89%	90%	61%	64%	61%	64%
Thaheryah	100%	11%	88%	88%	63%	68%	90%	90%
Tarablus	64%	17%	61%	64%	60%	52%	60%	60%
Arabiyah	50%	35%	44%	37%	33%	35%	35%	33%
Asad	94%	75%	97%	95%	97%	91%	71%	91%
Athraa'	61%	61%	90%	88%	98%	77%	75%	69%
Qedra	81%	15%	49%	49%	81%	63%	76%	76%

Table C.31: Per level and query weighted recall when using MBHA supervised approach/Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	70%	63%	37%	63%	43%	63%	43%	43%
Arafat	79%	79%	26%	26%	76%	92%	76%	89%
Maliki	82%	19%	25%	25%	89%	75%	93%	68%
Sakhr	79%	26%	65%	76%	21%	9%	14%	17%
Jamaa Arabiya	58%	58%	58%	58%	64%	55%	91%	83%
Thaheryah	45%	26%	39%	81%	74%	77%	87%	84%
Tarablus	48%	33%	48%	48%	85%	36%	79%	61%
Arabiyah	90%	57%	57%	57%	48%	48%	48%	48%
Asad	96%	93%	93%	93%	7%	84%	91%	63%
Athraa'	66%	48%	48%	48%	84%	47%	89%	85%
Qedra	29%	33%	29%	29%	76%	67%	76%	76%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	77%	67%	87%	53%	37%	63%	37%	37%
Arafat	100%	92%	95%	95%	71%	50%	74%	74%
Maliki	96%	45%	57%	93%	54%	18%	82%	79%
Sakhr	25%	82%	74%	82%	26%	68%	24%	24%
Jamaa Arabiya	83%	77%	89%	91%	66%	68%	66%	68%
Thaheryah	100%	26%	87%	87%	74%	74%	90%	90%
Tarablus	64%	33%	61%	64%	70%	52%	70%	70%
Arabiyah	52%	48%	48%	43%	43%	48%	48%	43%
Asad	93%	63%	98%	93%	98%	93%	63%	89%
Athraa'	66%	66%	90%	89%	98%	77%	76%	69%
Qedra	81%	29%	52%	52%	81%	71%	76%	76%

Table C.32: Per level and query weighted precision when using MBHA supervised approach/Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	80%	40%	13%	59%	62%	40%	62%	62%
Arafat	62%	62%	84%	64%	70%	93%	62%	89%
Maliki	81%	3%	75%	75%	90%	74%	95%	83%
Sakhr	72%	15%	58%	78%	22%	1%	19%	19%
Jamaa Arabiya	65%	65%	65%	65%	79%	76%	92%	87%
Thaheryah	82%	7%	82%	80%	71%	83%	89%	87%
Tarablus	52%	11%	52%	52%	90%	78%	87%	43%
Arabiyah	92%	33%	33%	33%	76%	76%	76%	76%
Asad	95%	92%	92%	92%	0%	86%	83%	94%
Athraa'	74%	76%	76%	76%	88%	76%	91%	89%
Qedra	31%	11%	31%	31%	86%	44%	86%	86%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	82%	67%	89%	65%	13%	40%	13%	13%
Arafat	100%	93%	96%	96%	88%	85%	85%	85%
Maliki	97%	72%	77%	95%	91%	73%	86%	80%
Sakhr	19%	79%	80%	77%	11%	83%	92%	92%
Jamaa Arabiya	85%	79%	91%	92%	80%	80%	80%	80%
Thaheryah	100%	7%	91%	91%	55%	69%	91%	91%
Tarablus	73%	11%	76%	73%	79%	52%	79%	79%
Arabiyah	58%	76%	53%	46%	47%	76%	76%	47%
Asad	99%	98%	97%	99%	97%	92%	92%	92%
Athraa'	74%	74%	92%	91%	98%	79%	84%	73%
Qedra	88%	10%	80%	80%	81%	80%	76%	76%

C.3.10 Bing/Supervised Clustering/Human-annotated

Table C.33: Per level and query macro F-measure when using MBHA supervised approach/Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	72%	19%	77%	77%	31%	57%	86%	81%
Jarrar	85%	29%	29%	29%	90%	22%	37%	37%
Alishaa'	76%	44%	74%	74%	47%	25%	45%	45%
Arafat	83%	60%	85%	78%	73%	56%	79%	79%
Maliki	28%	21%	49%	49%	79%	44%	70%	70%
Sakhr	29%	34%	35%	35%	54%	27%	52%	34%
Malek Abdullah	59%	54%	40%	38%	96%	65%	69%	63%
Jamaa Arabiya	73%	49%	55%	51%	75%	66%	34%	96%
Bursa	38%	38%	38%	36%	72%	47%	60%	56%
Thaheryah	76%	23%	74%	74%	68%	57%	57%	57%
Tarablus	82%	19%	80%	71%	71%	20%	72%	73%
Arabiyah	86%	17%	86%	74%	81%	77%	81%	81%
Asad	35%	37%	71%	71%	41%	44%	40%	83%
Athraa'	97%	37%	95%	95%	93%	37%	95%	95%
Qedra	91%	54%	89%	91%	22%	49%	69%	69%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	98%	54%	94%	98%	57%	19%	56%	56%
Jarrar	73%	20%	73%	73%	23%	20%	46%	35%
Alishaa'	40%	44%	77%	77%	59%	29%	70%	72%
Arafat	95%	52%	95%	95%	84%	91%	85%	85%
Maliki	66%	8%	94%	94%	29%	38%	27%	27%
Sakhr	84%	27%	74%	74%	53%	41%	53%	53%
Malek Abdullah	72%	63%	71%	64%	29%	49%	64%	70%
Jamaa Arabiya	80%	67%	77%	77%	55%	41%	53%	54%
Bursa	74%	47%	59%	56%	78%	45%	65%	80%
Thaheryah	74%	57%	57%	57%	74%	44%	74%	82%
Tarablus	95%	20%	99%	99%	62%	25%	62%	61%
Arabiyah	87%	21%	84%	84%	8%	43%	49%	60%
Asad	97%	37%	65%	65%	38%	49%	91%	46%
Athraa'	100%	53%	100%	100%	80%	58%	93%	93%
Qedra	89%	72%	84%	84%	49%	56%	49%	49%

Table C.34: Per level and query weighted recall when using MBHA supervised approach/Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	72%	34%	77%	77%	41%	69%	86%	83%
Jarrar	88%	33%	33%	33%	91%	29%	39%	39%
Alishaa'	76%	59%	76%	76%	52%	39%	59%	59%
Arafat	83%	69%	85%	78%	77%	67%	81%	81%
Maliki	28%	36%	48%	48%	81%	57%	75%	75%
Sakhr	41%	44%	44%	44%	59%	43%	57%	46%
Malek Abdullah	57%	53%	42%	40%	96%	63%	66%	61%
Jamaa Arabiya	73%	57%	61%	55%	76%	69%	48%	96%
Bursa	53%	53%	53%	53%	74%	58%	66%	63%
Thaheryah	77%	40%	74%	74%	72%	66%	66%	66%
Tarablus	83%	34%	81%	70%	70%	35%	71%	73%
Arabiyah	88%	26%	88%	81%	83%	81%	83%	83%
Asad	47%	48%	72%	72%	54%	52%	51%	83%
Athraa'	97%	52%	95%	95%	93%	51%	95%	95%
Qedra	91%	65%	89%	91%	35%	63%	74%	74%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	98%	67%	94%	98%	63%	34%	61%	61%
Jarrar	80%	28%	80%	80%	30%	28%	46%	38%
Alishaa'	49%	59%	79%	79%	67%	42%	74%	72%
Arafat	95%	65%	95%	95%	84%	91%	85%	85%
Maliki	72%	13%	94%	94%	28%	33%	27%	27%
Sakhr	85%	43%	77%	77%	61%	51%	61%	61%
Malek Abdullah	70%	61%	69%	62%	34%	48%	62%	67%
Jamaa Arabiya	81%	70%	78%	78%	61%	52%	59%	60%
Bursa	76%	58%	65%	63%	79%	57%	69%	81%
Thaheryah	77%	66%	66%	66%	77%	60%	77%	83%
Tarablus	95%	35%	99%	99%	68%	38%	68%	66%
Arabiyah	88%	29%	86%	86%	21%	43%	48%	57%
Asad	97%	48%	66%	66%	51%	56%	91%	54%
Athraa'	100%	60%	100%	100%	81%	63%	93%	93%
Qedra	89%	76%	85%	85%	63%	59%	63%	63%

Table C.35: Per level and query weighted precision when using MBHA supervised approach/Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	sw	2-g	sw__2-g	sw__2__3-g	sw	2-g	sw__2-g	sw__2__3-g
Amman	85%	78%	86%	86%	79%	79%	90%	86%
Jarrar	89%	84%	84%	84%	92%	83%	84%	84%
Alishaa’	82%	35%	79%	79%	70%	36%	49%	49%
Arafat	87%	79%	88%	85%	83%	78%	84%	84%
Maliki	64%	40%	73%	73%	83%	66%	72%	72%
Sakhr	45%	55%	68%	68%	80%	53%	79%	53%
Malek Abdullah	85%	84%	83%	83%	96%	86%	86%	85%
Jamaa Arabiya	73%	78%	79%	61%	83%	81%	62%	96%
Bursa	75%	75%	75%	28%	83%	77%	79%	78%
Thaheryah	85%	16%	84%	84%	81%	78%	78%	78%
Tarablus	84%	44%	80%	82%	82%	78%	82%	83%
Arabiyah	90%	83%	90%	85%	82%	79%	82%	82%
Asad	54%	64%	83%	83%	36%	78%	76%	86%
Athraa’	97%	75%	96%	96%	94%	76%	96%	96%
Qedra	91%	78%	89%	92%	34%	40%	82%	82%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	sw	2-g	sw__2-g	sw__2__3-g	sw	2-g	sw__2-g	sw__2__3-g
Amman	99%	45%	94%	99%	56%	78%	55%	55%
Jarrar	84%	83%	84%	84%	83%	83%	84%	84%
Alishaa’	77%	35%	84%	84%	79%	56%	82%	81%
Arafat	95%	43%	95%	95%	89%	91%	89%	89%
Maliki	81%	64%	94%	94%	55%	91%	55%	55%
Sakhr	89%	53%	85%	85%	57%	55%	57%	57%
Malek Abdullah	87%	85%	87%	85%	76%	84%	85%	86%
Jamaa Arabiya	86%	82%	85%	85%	79%	76%	78%	79%
Bursa	83%	77%	79%	78%	84%	76%	80%	86%
Thaheryah	83%	78%	78%	78%	83%	35%	83%	87%
Tarablus	95%	78%	99%	99%	65%	78%	65%	62%
Arabiyah	88%	84%	85%	85%	5%	84%	85%	86%
Asad	97%	54%	74%	74%	33%	61%	92%	61%
Athraa’	100%	78%	100%	100%	85%	79%	94%	94%
Qedra	91%	83%	88%	88%	40%	55%	40%	40%

C.3.11 Google/Supervised Clustering/BRF

Table C.36: Per level and query macro F-measure when using BRF supervised approach/Google.

	t	t	t	t	s	s	s	s
Dataset (Query)	sw	2-g	sw__2-g	sw__2__3-g	sw	2-g	sw__2-g	sw__2__3-g
Amman	54%	62%	73%	54%	18%	49%	83%	83%
Arafat	75%	75%	75%	75%	84%	80%	91%	91%
Maliki	40%	60%	15%	9%	43%	35%	65%	65%
Sakhr	11%	61%	2%	61%	11%	70%	12%	12%
Jamaa Arabiya	53%	46%	53%	53%	58%	85%	90%	89%
Thaheryah	73%	63%	73%	73%	69%	17%	82%	82%
Tarablus	53%	53%	53%	53%	49%	50%	50%	50%
Arabiyah	42%	42%	60%	42%	42%	44%	44%	44%
Asad	13%	1%	89%	89%	89%	87%	87%	86%
Athraa’	61%	61%	61%	52%	90%	42%	94%	94%
Qedra	17%	17%	55%	57%	60%	53%	60%	60%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	sw	2-g	sw__2-g	sw__2__3-g	sw	2-g	sw__2-g	sw__2__3-g
Amman	86%	48%	54%	54%	20%	49%	48%	36%
Arafat	100%	70%	94%	95%	75%	75%	7%	7%
Maliki	75%	50%	75%	75%	64%	2%	58%	76%
Sakhr	75%	11%	78%	17%	74%	67%	2%	17%
Jamaa Arabiya	48%	90%	53%	53%	67%	46%	61%	64%
Thaheryah	97%	63%	97%	97%	63%	63%	97%	70%
Tarablus	54%	50%	50%	53%	59%	52%	48%	53%
Arabiyah	48%	56%	60%	60%	60%	35%	67%	67%
Asad	86%	86%	86%	96%	7%	91%	2%	2%
Athraa’	61%	59%	61%	61%	62%	69%	70%	69%
Qedra	60%	72%	86%	86%	84%	53%	84%	84%

Table C.37: Per level and query weighted recall when using BRF supervised approach/Google.

	t	t	t	t	s	s	s	s	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	63%	70%	77%	63%	33%	63%	83%	83%	87%	60%	63%	63%	37%	63%	47%	37%
Arafat	82%	82%	82%	82%	87%	84%	92%	92%	100%	79%	95%	95%	82%	82%	18%	18%
Maliki	39%	71%	18%	14%	43%	36%	61%	61%	71%	46%	71%	71%	57%	11%	50%	71%
Sakhr	15%	62%	6%	65%	15%	76%	17%	17%	71%	15%	76%	25%	68%	65%	10%	21%
Jamaa Arabiya	58%	57%	58%	58%	62%	85%	91%	89%	57%	91%	58%	58%	70%	57%	66%	68%
Thaheryah	77%	74%	77%	77%	71%	29%	81%	81%	97%	74%	97%	97%	74%	74%	97%	77%
Tarablus	52%	67%	67%	67%	58%	61%	61%	61%	61%	61%	61%	52%	64%	64%	52%	67%
Arabiyah	57%	57%	67%	57%	57%	52%	52%	52%	57%	62%	67%	67%	67%	48%	71%	71%
Asad	12%	7%	88%	88%	93%	91%	89%	88%	86%	86%	86%	95%	9%	91%	9%	9%
Athraa'	66%	66%	66%	61%	90%	56%	94%	94%	66%	65%	66%	66%	66%	69%	71%	69%
Qedra	33%	33%	57%	57%	67%	67%	67%	67%	67%	71%	86%	86%	86%	67%	86%	86%

Table C.38: Per level and query weighted precision when using BRF supervised approach/Google.

	t	t	t	t	s	s	s	s	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	59%	80%	83%	59%	13%	40%	84%	84%	87%	39%	59%	59%	13%	40%	50%	43%
Arafat	85%	85%	85%	85%	89%	87%	93%	93%	100%	62%	95%	95%	85%	85%	4%	4%
Maliki	76%	51%	73%	73%	83%	75%	80%	80%	83%	87%	83%	83%	91%	1%	91%	92%
Sakhr	21%	59%	2%	58%	21%	74%	15%	15%	82%	21%	83%	17%	85%	83%	1%	21%
Jamaa Arabiya	65%	77%	65%	65%	69%	88%	92%	91%	67%	92%	65%	65%	81%	77%	80%	80%
Thaheryah	75%	55%	75%	75%	67%	81%	85%	85%	97%	55%	97%	97%	55%	55%	97%	83%
Tarablus	60%	44%	44%	44%	42%	43%	43%	43%	52%	43%	43%	60%	59%	44%	46%	44%
Arabiyah	33%	33%	79%	33%	33%	77%	77%	77%	55%	63%	79%	79%	79%	76%	81%	81%
Asad	61%	1%	95%	95%	86%	83%	85%	84%	86%	86%	86%	99%	61%	94%	1%	1%
Athraa'	74%	74%	74%	77%	90%	76%	94%	94%	74%	73%	74%	74%	71%	73%	76%	73%
Qedra	11%	11%	81%	71%	62%	44%	62%	62%	62%	85%	90%	90%	88%	44%	88%	88%

C.3.12 Bing/Supervised Clustering/BRF

Table C.39: Per level and query macro F-measure when using BRF supervised approach/Bing.

	t	t	t	t	s	s	s	s	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	81%	57%	85%	85%	31%	31%	30%	30%	92%	57%	92%	92%	63%	36%	63%	62%
Jarrar	29%	22%	29%	29%	38%	68%	30%	30%	35%	32%	33%	33%	48%	69%	45%	45%
Alishaa'	55%	27%	60%	60%	29%	27%	42%	66%	94%	32%	75%	75%	91%	54%	73%	71%
Arafat	50%	52%	52%	52%	62%	56%	56%	56%	63%	52%	63%	63%	83%	58%	84%	84%
Maliki	69%	22%	42%	55%	26%	41%	70%	70%	99%	53%	99%	99%	66%	53%	66%	76%
Sakhr	61%	32%	60%	60%	50%	27%	32%	34%	63%	28%	31%	31%	65%	39%	65%	56%
Malek Abdullah	46%	31%	50%	47%	70%	65%	65%	66%	88%	66%	66%	66%	49%	16%	81%	60%
Jamaa Arabiya	36%	34%	53%	53%	95%	59%	68%	82%	79%	72%	74%	72%	80%	29%	75%	90%
Bursa	44%	31%	40%	38%	71%	36%	61%	61%	71%	48%	61%	56%	36%	32%	36%	31%
Thaheryah	68%	49%	61%	61%	71%	44%	65%	57%	89%	57%	65%	65%	87%	49%	49%	79%
Tarablus	51%	53%	22%	17%	79%	17%	81%	82%	100%	22%	99%	99%	53%	27%	53%	53%
Arabiyah	86%	8%	93%	93%	91%	25%	84%	84%	98%	74%	84%	84%	49%	8%	60%	60%
Asad	76%	29%	43%	31%	48%	46%	30%	78%	36%	44%	83%	83%	71%	29%	90%	56%
Athraa'	97%	35%	97%	97%	94%	33%	94%	94%	100%	45%	100%	100%	85%	38%	91%	91%
Qedra	91%	29%	89%	89%	78%	54%	72%	72%	100%	49%	81%	89%	66%	54%	64%	57%

Table C.40: Per level and query weighted recall when using BRF supervised approach/Bing.

	t	t	t	t	s	s	s	s
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	83%	67%	86%	86%	41%	41%	39%	39%
Jarrar	33%	29%	33%	33%	40%	77%	34%	34%
Alishaa'	59%	42%	62%	62%	39%	42%	49%	67%
Arafat	63%	65%	65%	65%	70%	67%	67%	67%
Maliki	72%	36%	54%	63%	28%	55%	74%	74%
Sakhr	66%	43%	64%	64%	58%	43%	46%	47%
Malek Abdullah	46%	36%	47%	45%	67%	63%	63%	64%
Jamaa Arabiya	50%	49%	59%	59%	95%	63%	70%	82%
Bursa	56%	47%	54%	53%	73%	53%	66%	66%
Thaheryah	72%	62%	68%	68%	74%	60%	70%	66%
Tarablus	51%	66%	36%	34%	81%	34%	81%	82%
Arabiyah	88%	21%	93%	93%	90%	31%	86%	86%
Asad	77%	44%	50%	45%	54%	53%	45%	80%
Athraa'	97%	52%	97%	97%	94%	49%	94%	94%
Qedra	91%	41%	89%	89%	80%	65%	76%	76%

	t w/ s	t w/ s	t w/ s	t w/ s	ip	ip	ip	ip
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	92%	69%	92%	92%	67%	44%	67%	66%
Jarrar	38%	35%	36%	36%	47%	78%	45%	45%
Alishaa'	93%	45%	75%	75%	91%	58%	73%	72%
Arafat	71%	65%	71%	71%	83%	68%	84%	84%
Maliki	99%	62%	99%	99%	72%	63%	72%	78%
Sakhr	70%	43%	45%	45%	66%	49%	66%	52%
Malek Abdullah	88%	64%	64%	64%	48%	27%	81%	57%
Jamaa Arabiya	79%	74%	75%	74%	81%	46%	77%	90%
Bursa	73%	59%	66%	63%	53%	48%	53%	47%
Thaheryah	89%	66%	70%	70%	87%	62%	62%	81%
Tarablus	100%	36%	99%	99%	66%	39%	66%	66%
Arabiyah	98%	81%	86%	86%	48%	21%	57%	57%
Asad	47%	52%	84%	84%	71%	44%	90%	61%
Athraa'	100%	55%	100%	100%	85%	53%	91%	91%
Qedra	100%	63%	83%	89%	67%	65%	67%	63%

Table C.41: Per level and query weighted precision when using BRF supervised approach/Bing.

	t	**t**	**t**	**t**	**s**	**s**	**s**	**s**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	86%	62%	88%	88%	79%	79%	67%	67%
Jarrar	84%	83%	84%	84%	84%	61%	84%	84%
Alishaa'	80%	76%	80%	80%	41%	76%	63%	82%
Arafat	42%	43%	43%	43%	80%	78%	78%	78%
Maliki	74%	67%	67%	68%	37%	66%	74%	74%
Sakhr	67%	61%	64%	64%	55%	53%	52%	53%
Malek Abdullah	84%	83%	69%	70%	86%	86%	86%	86%
Jamaa Arabiya	76%	76%	78%	78%	95%	79%	82%	85%
Bursa	76%	22%	76%	75%	82%	28%	79%	79%
Thaheryah	81%	77%	79%	79%	82%	35%	80%	78%
Tarablus	52%	44%	78%	11%	82%	11%	88%	88%
Arabiyah	90%	5%	93%	93%	93%	84%	85%	85%
Asad	84%	45%	60%	25%	76%	78%	29%	86%
Athraa'	97%	27%	97%	97%	95%	75%	95%	95%
Qedra	91%	77%	89%	89%	85%	78%	83%	83%

	t w/ s	**t w/ s**	**t w/ s**	**t w/ s**	**ip**	**ip**	**ip**	**ip**
Dataset (Query)	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**	**sw**	**2-g**	**sw__2-g**	**sw__2__3-g**
Amman	94%	79%	94%	94%	64%	79%	64%	62%
Jarrar	84%	84%	84%	84%	85%	61%	84%	84%
Alishaa'	94%	77%	84%	84%	92%	79%	83%	82%
Arafat	80%	43%	80%	80%	88%	79%	89%	89%
Maliki	99%	67%	99%	99%	71%	67%	71%	84%
Sakhr	60%	53%	52%	52%	73%	54%	73%	70%
Malek Abdullah	92%	86%	86%	86%	80%	82%	81%	80%
Jamaa Arabiya	86%	83%	84%	83%	86%	22%	81%	92%
Bursa	82%	77%	79%	78%	28%	75%	28%	22%
Thaheryah	92%	78%	80%	80%	89%	77%	77%	86%
Tarablus	100%	78%	99%	99%	44%	78%	44%	44%
Arabiyah	98%	85%	85%	85%	85%	5%	86%	86%
Asad	38%	66%	84%	84%	78%	45%	92%	75%
Athraa'	100%	77%	100%	100%	88%	75%	93%	93%
Qedra	100%	40%	86%	91%	66%	78%	66%	59%