On Achieving High Quality User Reviews Retrieval in the Context of Conversational Faceted Search

Konstantinos Sgontzos

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Associate Prof. *Yiannis Tzitzikas*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

University of Crete
Computer Science Department

On Achieving High Quality User Reviews Retrieval in the Context of Conversational Faceted Search

Thesis submitted by
Student K. Sgontzos
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Student K. Sgontzos

Committee approvals: _____
Yiannis Tzitzikas
Associate Professor, Thesis Supervisor

_____
Dimitris Plexousakis
Professor, Committee Member

_____
Giorgos Flouris
Principal Researcher, Committee Member

Departmental approval: _____
Antonios Argyros
Professor, Director of Graduate Studies

Heraklion, February 2019

On Achieving High Quality User Reviews Retrieval in the Context of Conversational Faceted Search

Abstract

Question Answering Systems can be considered as an extension of Search Engines in the sense that they directly answer questions posed by users in Natural Language instead of simply returning relevant sources based on keyword search. Recently, such systems gained more attention and have great potential in various domains including e-commerce.

The general context of our work is to support Faceted Search (FS) in the form of spoken dialogue. FS is an exploratory search paradigm that is commonly applied over multidimensional or graph data. However, since structured data might not suffice for answering a user's query, unstructured information like user comments or reviews can be a valuable source of information. In this work we focus on how to leverage the textual information (user reviews and comments) for answering the user questions. This information is generally restricted in the length of a sentence or a paragraph and is known as Answer Passage (AP). Due to the speech interaction with the user, high quality user review retrieval is necessary.

To study this problem, we created an evaluation collection and we adjusted to our problem an existing TREC collection. We then proposed three basic approaches and variations of them. These approaches are based on handmade (Wordnet) and statistical (Word2vec) dictionaries. The comparative experimental results show that: (a) The combination of handmade and statistical dictionaries succeed better results in terms of precision in relevant AP (review) retrieval, (b) The expansion of the question with other relevant terms combined with methods that devalue question words which describe the general domain context, improve the precision, mainly in restricted domains such as hotel booking. In comparison to existing related work, our proposed approaches achieve higher precision.

Υψηλής Ποιότητας Ανάκτηση Σχολίων Χρηστών στα Πλαίσια της Διαλογικής Εξερευνητικής Αναζήτησης

Περίληψη

Τα Συστήματα Απάντησης Ερωτήσεων (Question Answering Systems) μπορούν να θεωρηθούν ως επέκταση των Μηχανών Αναζήτησης, υπό την έννοια ότι προσπαθούν να απαντήσουν τις ερωτήσεις των χρηστών που τίθενται σε φυσική γλώσσα, αντί απλά να επιστρέφουν πόρους συναφείς με τις λέξεις της ερώτησης. Η χρήση τέτοιων συστημάτων αυξάνεται ραγδαία τα τελευταία χρόνια και έχουν πολλά πεδία εφαρμογής (e-commerce, και αλλού).

Το γενικό πλαίσιο αυτής της εργασίας είναι η στήριξη της διαδικασίας Εξερευνητικής Αναζήτησης στη μορφή διαλόγου σε φυσική προφορική γλώσσα. Συγκεκριμένα επικεντρωνόμαστε στο πλαίσιο της Πολυεδρικής Αναζήτησης (Faceted Search), η οποία είναι μία μορφή εξερευνητικής αναζήτησης που συχνά εφαρμόζεται σε πολυδιάστατα δεδομένα ή δεδομένα γραφικής δομής. Ωστόσο τυχαίνει μια ερώτηση του χρήστη να μην μπορεί να απαντηθεί από τα δεδομένα, και σε αυτήν την περίπτωση τυχόν αδόμητη πληροφορία σε μορφή κειμένου, όπως σχόλια και κριτικές από χρήστες, αποτελούν μια πολύτιμη πηγή πληροφοριών. Στην εργασία αυτή, επικεντρωνόμαστε στο πως θα χρησιμοποιήσουμε αυτά τα έγγραφα (κριτικές και σχόλια χρηστών) για την απάντηση των ερωτήσεων του χρήστη. Τα έγγραφα αυτά περιορίζονται συνήθως στο μέγεθος μιας πρότασης ή μιας παραγράφου και είναι ευρέως γνωστά ως Παράγραφοι Απαντήσεων (Answer Passages). Λόγω της αλληλεπίδρασης με το χρήστη με προφορικό λόγο, απαιτείται υψηλής ποιότητας ανάκτηση σχολίων.

Στην εργασία αυτή προτείνουμε διάφορες μεθόδους για τον εντοπισμό των πιο συναφών σχολίων χρηστών. Για τη μελέτη του προβλήματος φτιάξαμε μια συλλογή αξιολόγησης, και προσαρμόσαμε μια υπάρχουσα του TREC στις ανάγκες του προβλήματός μας . Εν συνεχεία προτείναμε τρία βασικά μοντέλα, και παραλλαγές αυτών. Το μοντέλα βασίζονται σε ευρέως διαδεδομένα χειρόγραφα (Wordnet) και στατιστικά (Word2vec) λεξικά. Τα συγκριτικά πειραματικά αποτελέσματα έδειξαν ότι: (α) ο συνδυασμός χρήσης χειρόγραφων και στατιστικών λεξικών επιτυγχάνει μεγαλύτερα ποσοστά ακρίβειας στο πρόβλημα της εύρεσης συναφούς παραγράφου, (β) ο εμπλουτισμός της ερώτησης του χρήστη με περαιτέρω σχετικούς όρους, σε συνδυασμό με ανακατανομή των βαρών των όρων της, βασισμένη σε λέξεις που περιγράφουν το γενικό πλαίσιο του συνόλου των πληροφοριών, αυξάνουν την απόδοση του συστήματος, κυρίως όταν έχουμε ένα συγκεκριμένο πεδίο, όπως αυτό της κράτησης ξενοδοχείου. Σε σχέση με τις αντίστοιχες εργασίες, οι μέθοδοι που προτείνουμε επιτυγχάνουν υψηλότερο βαθμό ακρίβειας.

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον supervisor Γιάννη Τζίτζικα για την υποστήριξη του στην έρευνα και το έργο αυτού του thesis . Δεν θα μπορούσα να παραλείψω τις ευχαριστίες μου στους υπόλοιπους συναδέλφους του εργαστηρίου, Λευτέρη Δημητράκη, Μιχάλη Μουνταντονάκη, Παναγιώτη Παπαδάκο, Γιάννη Μαρκετάκη και Κώστα Μανιουδάκη. Χωρίς την συνεισφορά τους το έργο δεν θα είχε επιτευχθεί. Επίσης θα ήθελα να ευχαριστήσω την υπόλοιπη επιτροπή, που αποτελείται από τους Δημήτρη Πλεξουσάκη και Γιώργο Φλουρή, για τον χρόνο που αφιέρωσαν και τις πολύτιμες συμβουλές τους. Ακόμη, θα ήθελα να ευχαριστήσω το ΙΤΕ-ΙΠ που στήριξε αυτην την προσπάθεια με υποτροφία. Τέλος, περισσότερο από όλους θα ήθελα να ευχαριστήσω τους γονείς μου που με έφεραν στον κόσμο, με δίδαξαν την σημασία της γνώσης και της μόρφωσης και συνεχίζουν να με στηρίζουν και να με καθοδηγούν ακόμα με κάθε δυνατό τρόπο.

στους γονείς μου

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Nowadays, the interaction by voice has become very popular, as evidenced by the adoption of systems like Microsoft Cortana, Amazon Echo & Alexa, Ok Google and Apple Siri. Generally, Conversational Systems (CS) is the way that humans communicate with intelligent agents like Search Engines and Questions Answering Systems. They are very convenient, since they allow users without any background knowledge in the field of Artificial Intelligence (AI) to be able to interact with intelligent devices in their daily life. As an example consider smart homes, auto driving cars and personal assistants. In this work, we focus on the last case, and more specifically in Information Retrieval (IR) for CS.

Faceted Search (FS) is a widely used exploratory search paradigm. It is used whenever the user wants to find the desired item from a list of items (either products, hotels, restaurants, publications, etc). Typically, FS offers exploratory search over multidimensional or graph data. However sometimes the structured data are insufficient for answering a user's query. User reviews (or comments) is a valuable source of information that could be exploited in such cases for aiding the user to explore the information space and to decide what options suits him/her better. Indeed, user reviews/comments are available in various applications of faceted search, e.g. for hotel booking and in product catalogs.

Enabling the interaction of FS through spoken dialogue, is appropriate for situations where the user cannot (or is not convenient to) use his hands or eyes. In such cases, the user interacts using his voice and provides commands or poses questions. If a question cannot be translated to a query over the structured resources of the dataset, then the system cannot deliver any answer. In such cases it is reasonable to resort to the available unstructured data, i.e. to users comments and reviews. Figure 1.1 illustrates the context. The objective is not to provide the user with a direct answer, but first to identify the relevant reviews based on the user's question. To grasp the idea an indicative dialogue is shown in Figure 1.2 (note that Minami is a region in Kyoto). Note that so far we support simple utterances. A user question either goes to the structured dataset, or to the reviews. Our system will fail to answer a mixed query like "Three star hotels in Kyoto with

no complaints about noise", which can be partially answered by the structured dataset and partially by user reviews. Except if all this information happens to be in a single passage which is quite difficult. To this end, we focus on the task required for answering the last question in this dialogue, i.e. the question "About these, has anyone reported a problem about noise?".

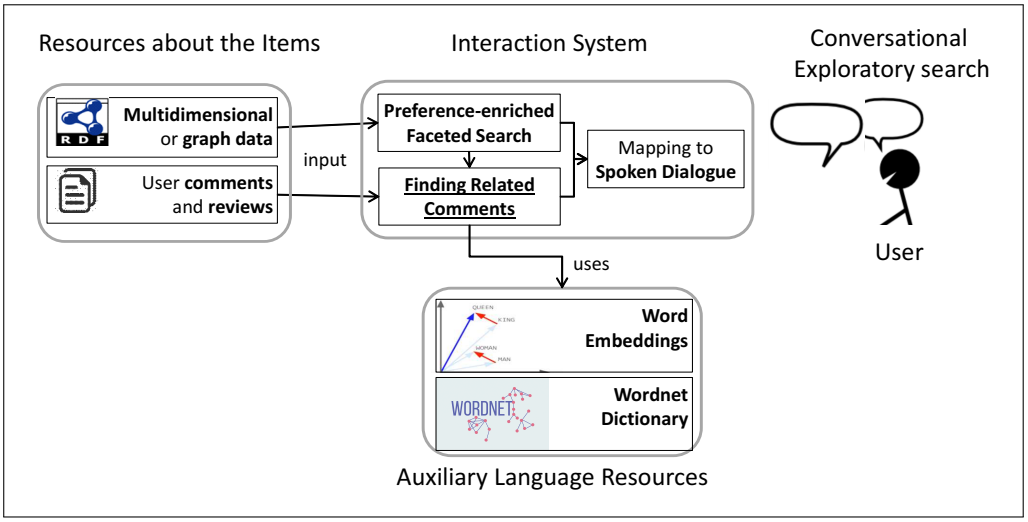

Figure 1.1: Finding Related Comments and Conversational Faceted Search

System	Hello, welcome to the Japanese Hotels spoken dialogue system. How may I help you?
User	I"m looking for a hotel in Kyoto but not in Minami where they offer free Wi-Fi and have non smoking rooms.
System	3 hotels match your preferences
User	About these, has anyone reported a problem about noise?
System	I have found some user mentions about this topic. For hotel H1 a user mentioned at 12/05/2018 the following: "There is too much noise because of the airport." For hotel H2 another user mentioned at 14/06/2018 the following: "The music of the bar is too loud."

Figure 1.2: Indicative dialogue

Note that direct query answering is reasonable only in cases where, there is a single and credible source of unstructured data (e.g. wikipedia). This is not the case with user comments since they can be numerous, and their content can be conflicting. If we manage to find the relevant comments, then the system could either read these comments to the user, or attempt to apply question answering if the user requests so, or any other kind of analysis, e.g. sentiment analysis. In any case spoken dialogue interaction, poses increased requirements on quality, since the system should not "read" irrelevant comments as reading costs user time.

Note that instead of analyzing the user comments for estimating whether a hotel is good or bad as a whole, the interaction that we propose enables the user

to get information about the particular aspects or topics that are important for him, e.g. about noise, cleanliness, the quality of the wifi, parking, etc. The set of such topics is practically endless and we cannot make the assumption that structured data will exist for all such topics. Therefore, it is beneficial to have systems that are able to exploit associated unstructured data, e.g. user comments and reviews. The problem is challenging since user comments are usually short, meaning that it is hard to achieve an acceptable level of recall. In this paper we focus on this problem, and we introduce methods relying on hand crafted and statistical dictionaries for identifying the relevant comments.

Subsequently, we report comparative evaluation results over various collections, specifically over FRUCE-v1[1] and FRUCE-v2[2], two different versions of a small evaluation collection that we have created for the problem at hand, as well as over WebAP[3], a TREC collection comprising 80 queries over about 8 thousands manually annotated passages. Since there is no collection for relevant User Review Retrieval, we have created the former one to get insights and spot issues about the problem at hand. While we have used the later one to evaluate our proposed methods in a larger collection that is publicly available and appropriate for Answer Passage Retrieval. WebAP is also suitable to us since we approach the User Review Retrieval as a sentence ranking problem, as we describe in §4.2. FRUCE-v1 evaluation collection introduced in [8] where we used it to comparatively evaluate the introduced methods.

In a nutshell, the key contributions of this work are: (a) we show how the FS interaction can be extended for exploiting unstructured data in the form of user comments and reviews, and (b) we introduce and comparatively evaluate four methods for identifying the more relevant user comments in datasets related to the task of hotel booking as well as a dataset with web passages based on TREC GOV2 queries.

This is an extended version of the paper presented in [8]. In comparison to that paper, in this work, we present a more complete analysis of the problem with emphasis on the evaluation to get new insights about the effectiveness of the introduced methods.

Note that the scope of this work is restricted in the User Review Retrieval task as an extension of Conversational Faceted Search (CFS) or even FS. The exploitation of structured data, i.e. the CFS interaction framework is not implemented here. Instead we use the research prototype LD-SDS [30] system as base, in order to understand the integration challenges, as well as to identify the cases where our system is useful and complementary to the CFS component.

The rest of the thesis is organized as follows: Chapter 2 presents the required background and related work. Chapter 3 discusses the challenges of the problem of review retrieval. Chapter 4 describes the proposed methods. Chapter 5 reports

[1] `http://islcatalog.ics.forth.gr/dataset/fruce-v1`
[2] `http://islcatalog.ics.forth.gr/dataset/fruce-v2`
[3] `https://ciir.cs.umass.edu/downloads/WebAP/`

experimental results. Chapter 6 explains the applicability of the system. Chapter 7 discusses possible extensions and alternative approaches. Finally, Chapter 8 concludes the thesis and discusses directions for future research and work.

Chapter 2

Context and Related Work

This chapter describes the context, the required background and related work with respect to our problem, i.e. Conversational Faceted Search (CFS). More specifically, §2.1.1 introduces FS, §2.1.2 introduces work that extends FS through voice interaction, §2.2.1 introduces word embeddings and presents various applications of them, §2.2.2 briefly describes the structure of Wordnet as well as the relations that we exploit, §2.2.3 describes an extension of Faceted Search with preferences, §2.3.1 exhibits work that exploits user reviews for products in various ways and §2.3.2 presents related work in the area of Answer Passage Retrieval (APR), i.e. retrieving small text spans to answer user's questions.

2.1 Context

2.1.1 Faceted Search

Faceted search is the de-facto standard in e-commerce and tourism services. It is an interaction framework based on a multi-dimensional classification of data objects, allowing users to browse and explore the information space in a guided, yet unconstrained way through a simple visual interface [37]. Features of this framework include: (a) display of current results in multiple categorization schemes (called facets, or dimensions, or just attributes), (b) display of facets and values leading to non-empty results only, (c) display of the count information for each value (i.e. the number of results the user will get by selecting that value), and (d) ability to refine the focus gradually, i.e. it is a session-based interaction paradigm in contrast to the stateless query-and-response dialogue of most search systems. Faceted search is currently the de facto standard in e-commerce (e.g. eBay, booking.com), and its popularity and adoption is increasing. It has been proposed and applied for web searching, (e.g. [27]), for semantically enriching web search results, (e.g. [9]), for patent-search, (e.g. [10]), as well as for exploring RDF and Linked Data (e.g. see [11, 38], as well as [43] for a recent survey). or even fuzzy RDF datasets (e.g [21]).

2.1.2 Conversational Faceted Search

Only a few works exist that involve speech interfaces on top of the faceted search paradigm. For example, the work described in [6], exploits a speech user interface over facets that index audio metadata associated with audio content. This system is used for the Spoken Web, an alternative to WWW based on audio content, and the associated Medieval Spoken Web Search Task [23], that involves searching language independent audio content in audio content using an audio content query. A faceted browser over datasets available in the Linked Open Data (LOD) cloud is described in [18], where SPARQL queries that are submitted to LOD end-points are generated through NLP over voice commands issued by the user. An other approach is presented in [40], however the authors are restricted in methods of result presentation through a voice channel. Another recent related work is [39] that proposes a ML framework to build a personalized conversational recommendation agent that optimizes a per session-based utility function, it presupposes having user's past rating and past queries. That work assumes only a structured dataset, textual user comments and reviews are not considered. A more recent work, highly relevant to ours, is FASELOD [19]. It is an FS engine that operates over various LOD datasets by selection of values over facets through voice commands, which are then translated constraints on the dataset. Enrichment of the results with related information from other datasets through a Silk-based mechanism is also supported. In contrast to this, in our work we also support preference priorities over the dataset facets that correspond to soft constraints. Neither that work considers textual user reviews in the spoken interaction.

To the best of our knowledge though, the only work that combines spoken dialogue systems with faceted search is the one presented in [30], where the described LD-SDS system is limited to spoken dialogues over structured datasets (expressed in RDF). In this work we extend conversational faceted search for exploiting also available unstructured data (e.g. user reviews). Note that, tackling the same problem using only a single large-scale source of unstructured data, e.g. Wikipedia (as described in [1]), is much easier since in that case we do not have the source selection problem (selection of user reviews in our case), and the source contains many and long texts, therefore it is not difficult to achieve a good recall level.

2.2 Background

2.2.1 Word Embeddings

Word Embeddings are widely used in information retrieval in order to bridge the lexical gap between the user questions and the underlying textual sources. *Word Embeddings* were initially proposed in [25]. It is a neural network based architecture that is unsupervised trained in large corpus of text and learns to predict either a missing word for a given context (CBOW), or to predict the context words

given a single word (skip-gram), both in a fixed window size. The learning process produces the so called *Word Embeddings*, that are vector representations of words, which their distances in the vector space reveal their semantic association. Thereafter, other works were build upon that, such as other models for creating *Word Embeddings* such as Glove [33], ELMO [34] and BERT [5] (newest), text retrieval approaches [14, 46] and query expansion techniques [7, 36, 15]. In this work we focus in approaches in the area of text retrieval at sentence level.

2.2.2 Wordnet

WordNet [26] is publically available[1] lexical database for English language. It consists of *166,000 (f,s) pairs*, where f is a word form and s the set of words that have the same sense. It also defines relations between words and senses like *Synonymy*, *Antonymy* (opposing-name), *Hyponymy* (sub-name), *Hypernymy* (super-name), *Meronymy* (part-name), *Holonymy* (whole-name), *Troponymy* (manner-name) and *Entailment*. There is also an extension of Wordnet, called Universal Wordnet [3] which extends it to over 200 languages. Wordnet's usage on Question Answering and IR can be for extending the input query with further relevant terms or for improving the lexical disambiguation task. In this work we exploit only the basic relations of Wordnet, i.e. *Synonymy*, *Antonymy* and *Hypernymy*.

2.2.3 Preference Enriched Faceted Search

The enrichment of faceted search with *preferences*, hereafter *Preference-enriched Faceted Search*, for short PFS, was proposed in [44, 28]. PFS offers actions that allow the user to order facets, values, and objects using *best, worst, prefer to* actions (i.e. relative preferences), *around to* actions (over a specific value), or actions that order them lexicographically, or based on their values or count values. Furthermore, the user is able to *compose* object related preference actions, using *Priority, Pareto, Pareto Optimal* (i.e. skyline) and other. The distinctive features of PFS is that it allows expressing preferences over attributes, whose values can be hierarchically organized (and/or multi-valued), it supports preference inheritance, and it offers scope-based rules for resolving automatically the conflicts that may arise. As a result the user is able to restrict his current focus by using the faceted interaction scheme (hard restrictions) that lead to non-empty results, and rank the objects of his focus according to the expressed preferences. Recently, PFS has been used in various domains, e.g. for offering a flexible process for the identification of fish species [42], as a Voting Advice Application [41], as well as, for data that contain also geographical information [17].

[1] `http://wordnet.princeton.edu/`

2.3 Related Work

2.3.1 Exploiting User Reviews

As regards the idea of leveraging user reviews about products, for further supporting the user to choose the one that suits him/her better, there are a few works that exploit them for various reasons.

For example in [32] they leverage user reviews to improve the accuracy of Mobile App Retrieval. To do so, they propose *AppLDA*, which is an LDA-based topic modeling approach for retrieving the most relevant app for a user query, based on the apps description and user reviews. Additionally to the proposed model, they created an evaluation collection for the task of app retrieval, with data retrieved from Google Play app store. In the aforementioned collection they reported that *AppLDA* performs better, in terms of nDCG, compared to Google Play's search results, as well as other state-of-the-art document retrieval models such us BM25, BM25F etc. In contrast with this approach, in our work we focus on retrieving unknown relevant reviews rather than leveraging them for a final object (e.g. hotel) ranking. That means that our approach is suitable for improving exploratory search capabilities, by further enriching the information space with relevant textual information rather than automatically affect the ranking of objects, since the later one is happening by the *CFS* component through user speech actions. Furthermore, since they exploit all user reviews and descriptions for each app for generating the topics, it is expected to have more textual evidence for each object. On the other hand, since our task is to rank each review as separate, the available textual evidence is quite restricted. To this end, we also provide query expansion and weightening techniques (see Section 4.3), which, as we experimentally show in sections 5.3.0.1 and 5.4.1, improve the retrieval performance in a wide variety of evaluation metrics.

An other work that could be considered similar to ours is [22], where the authors addressing complex and subjective product-related queries with customer reviews. To succeed that, they propose a model that jointly learns question answering and a query-review relevance function, given large volumes of questions-answer pairs about products and a collection of reviews for the related products. They learn which reviews are relevant, using a Mixtures of Experts techniques, where they use each review as an expert an let it vote for the correct answer. The reviews that vote correct are considered the ones that are relevant. However, we cannot use such a collection for our purpose, since our method contains no predictor for the final answer, and there is not a concrete collection of whether a query-review pair is relevant or not, and with what level of relevance.

In [31] they retrieve relevant opinion sentences for products that lack of user reviews, based on similar products that have plenty of them. Specifically, they aim to find relevant sentences from user reviews of old or popular products (which they are expected to have many reviews), given as query new or unpopular products, for which reviews do not exist. They use probabilistic methods for the retrieval

problem and they show that they perform better than review-based modified versions of standard summarazation methods such as MEAD [35]. To evaluate the proposed methods they build a collection for this task. To do so, they used a dataset of products with known reviews, but they pretended that they did not know them. Thereafter, they compared the retrieved review sentences with the real ones. In contrast to this work, we aim at using a natural language questions as a query rather than a product, where the query consists of feature-value pairs. That means that the query itself contains noise as we discuss in Chapter 3 in the CW challenge, thus we also propose a method for reducing the noise from the query questions, as we described in Section 4.3.

Finally, a rather relevant work but in a different context, is [29] that focuses on measuring the relatedness of two forum posts. In this work the authors show that treating posts as composite objects, consisting of different parts or segments, where each part serves a different intention (e.g. problem description, background, desire, etc.), can lead to more effective retrieval. The relatedness of two posts is then based on comparing segments that serve the same goal, instead of comparing the whole posts. The authors present several experiments regarding the right segmentation criteria, the effectiveness of the segmentation algorithms and the formation of intention clusters. Their approach increased the mean precision by more than 10%.

2.3.2 Answer Passage Retrieval

Answer Passage Retrieval (APR) from the area of Question Answering (QA) is similar to the reviews retrieval task. One work in this area [1] performs natural language understanding over wikipedia to identify the answer boundaries from a given text passage (i.e. a text-unit like the size of a paragraph) and an input question, while [47], identifies for a given input question the right sentence from a list of candidate sentences. A recent work [2] trains a long short term memory (LSTM) network in order to extract relevant passages from noisy collections of sentences for given a query. Our approach resembles the process followed in APR, in the sense that we try to find the most relevant sentence within comment reviews in comparison to the portions of the document that are relevant to a particular query. Under this light, we argue that our approach could be used as well (at least for factoid questions). However, these works usually rely on a single and credible source (e.g. wikipedia in the case of [1]), while our work is applied over a collection of short (and in some cases contradicting) sentences. Although APR systems have the ability to answer more complex queries, since they combine multiple aspects in various text passages, they mostly focus on single turn QA, whereas in our work the retrieval process is nested within the dialogue process and triggered only when needed.

Chapter 3

Challenges and Requirements

3.1 Main Challenges

Here we discuss the five main challenges that we have identified for the task at hand and how to tackle them.

R1: *The short length of the reviews* prevents them from providing enough information about their topic. Therefore it is difficult to achieve a good recall for various queries. As an example consider the following review:
Negative part: "The WiFi was terrible, I was on the 8th floor and could barley get any signal. There should be more WiFi spots considering the size of the hotel."
Positive part: "I liked how close it was to the station, within walking distance but still quiet. The room was comfy and spacious and was very clean."
In this example the number of distinct words (if we filter out any stop-word), is eighteen. Moreover, note that hose eighteen words, cover the topics of *WiFi-quality*, *distance-from-station*, *room-comfort*, *room-cleanliness*. Thus for an input query about *room-cleanliness*, only the word "clean", in the last sentence of the positive part provide relevant information (challenge R2 provides more information about this challenge). To tackle this challenge, query expansion techniques are certainly required, and this is why we propose and comparatively evaluate query expansion techniques (later in §4.3).

R2: Reviews are posted to indicate the pros and cons of a resource (in our case study, a hotel). To do so, users write about various topics in a single paragraph and the *reviews lack a coherent context*. For example, the following review mentions two different topics despite its small length:
Negative part: "-"
Positive part: "The room was clean. The breakfast was acceptable."
The same stands for the example in the previous challenge (R1). To overcome this problem, the scoring of the reviews should be based on the most relevant sentence (and this is what we shall see in §4.3).

R3: Most of the reviews either *describe problems or favorable impressions for the resources of the domain*, even if they do not explicitly contain the relevant terms (e.g. "problem", "issues", etc). This means that in the question of the form "Has anyone reported a problem about noise?" we should not treat the words "problem" and "noise" equally, since the former describes the implicit context and the latter the focus of the question. Consequently, and especially for closed-domain datasets, it is beneficial to construct a set of words, denoted as $W_{context}$, describing the implicit context. For instance, for the domain of hotels such a set could include words like $\{comment, report, issue, problem, complaint, hotel\}$. It is important to note that context terms should not be completely removed from computations (like stop-words), since a user question could be general e.g. "Is there any problem with this hotel?". This is why in §4.3 we focus on this problem and propose a method that deals with the Context Words (CW) problem. Note that the objective is to keep the CW list small and general, since a larger and more complicated list would be time laborious and almost impossible to be manually made. To this end, a CW list with 5-10 words can be easily and quickly build (for the collections used in the experimental evaluation, described in Chapter 5, it took 1-2 minutes to define such a list for the small domain specific collection FRUCE-v2, and 3-6 minutes for the large government domain collection WebAP).

R4: In the context of CFS, *the user can be provided with a limited amount of relevant reviews by the speech interface*, since it is time consuming to read aloud the comments. For this reason, excellent quality is required for the top hits. This is why we focus on how to achieve absolute precision on the top-2 results. It is also quite beneficial, to reply by reading to the user only the highest scored sentence, to avoid reading unnecessary information. Thus in a case such as the example of R1, we would read to the user only the sentence "The room was comfy and spacious and was very clean.".

R5: Review retrieval should be fast enough to be done at real-time without the need to index the reviews in advance. This is succeeded from the *CFS* component. By restricting the objects in focus, the number of reviews to be scored, is restricted analogously.

Chapter 4

The Proposed Approach

In §4.1 we describe an extension of the interaction of PFS for exploiting also associated unstructured data, in §4.2 we focus on the problem of finding the relevant reviews and in §4.3 we explain in detail the scoring functions used for the retrieval task.

4.1 The Interaction

The user interacts with the system in spoken language. These actions are then translated to PFS (Preference-enriched Faceted Search)[1] actions and queries for the user comments/reviews. PFS actions allow the user to express hard and soft constraints. The first restrict the objects in the focus (objects of interest), while the latter express preferences and rank the objects in the focus (specifically they define a ranked list of buckets, each bucket containing incomparable, with respect to the preferences, objects). The process is illustrated in Figure 4.1.

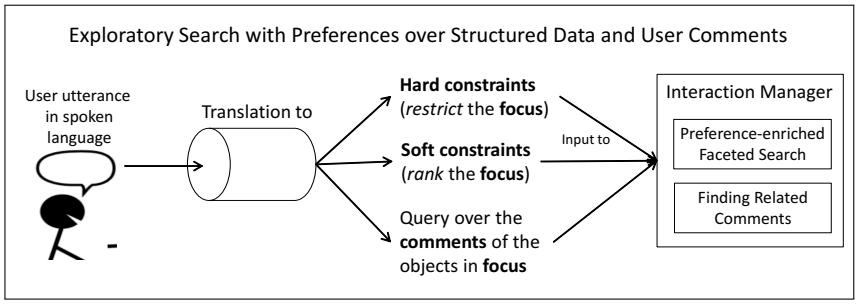

Figure 4.1: The basics of the interaction process

The user comments/reviews enrich the process as follows: If the cardinality of the top bucket, or the number of objects in focus, is below a configurable threshold (e.g. 10), and the user question cannot be answered by the structured data, then user reviews are exploited. Specifically we shall use the term *ofocus* to

[1]PFS was described in brief in §2.1.1.

refer to the restricted set of objects (those after applying all filters), and *pfocus* to refer to the first bucket of the focus, that contains the most preferred objects (obviously $pfocus \subseteq ofocus$). If the cardinality of either of the above sets is below a configurable threshold θ (say 10), then in case the user's questions cannot be answered by the structured dataset, the system resorts to the user reviews for this.

Note that if at some point in the interaction, the user's focus is big (i.e. if $\min(|ofocus|, |pfocus|) > \theta$) and the user asks a question that cannot be answered by the structured dataset, then the system suggests the user to *"please first refine the focus or your preferences"* in the sense that it is not useful to ask questions of the form "quiet hotel in Rome", or "hotels with fast wifi in London". In other words, we could say that the system enters this mode in the so-called "End Game" phase of faceted search [37].

This way of treating user comments/reviews has several benefits:
(a) *Applicability*: Since there is no need to index comments a priori, this model can be applied over RSS feeds and blog comment hosting services (e.g. Disqus).
(b) *Efficiency:* Since the analysis will be applied only for the comments of the hotels in the user focus, it is feasible to make this analysis at real time.
(c) *Less Noise, Better Quality:* As in (b) the quality of the retrieved comments is expected to be higher to that of the entire set of reviews of all domain objects.

Note that currently we focus on simple utterances, in the sense that a user question is analyzed either based on the structured dataset, or the available reviews. Our research prototype does not currently support mixed queries like "Three star hotels in Kyoto with no complaints about noise", which require to simultaneously exploit the available structured and unstructured information for answering the question. That scenario presupposes that all reviews are available and have been indexed. The investigation of which is the best way to treat such questions, in such a setting, goes beyond the scope of this work.

4.2 General Scoring Approach

Given the process described before, the general process for identifying the more relevant reviews of the objects in the focus, comprises the following steps:

- For each review r we split its text into individual sentences and we get a set of sentences r_k, where $1 \leq k \leq \gamma$ and γ is the number of sentences in each review. The score of a review is the highest score that one of its sentences achieved.
- Apply tokenization, removal of stop-words and punctuations, as well as lemmatization (using Stanford CoreNLP [20]) both to the input question q and each sentence r_k of r, getting in this way the sets of words W_q and W_{r_k} respectively.
- Construct the method-related representation of q and r_k (detailed below in §4.3).

- Score and rank reviews based on the defined relevancy formula (detailed below in §4.3).

We shall use a scoring function for estimating the relevance between an input question q and each user review r_k of the objects in focus. Below we introduce various scoring methods, based on the hand made WordNet and the statistical Word2vec dictionaries. Although there are various embedding models as described in section §2.2.1 we decided to use a widely used model, i.e. the skip-gram (Word2vec) model. More precisely we use the *dl4j*[2] implementation. *WordNet* [26] is a lexical database for the English language comprising 166,000 (f, s) pairs, where f is a word-form and s is the set of words that have the same sense. It also includes relations between words and senses (like Synonymy, Antonymy, Hypernymy etc.). *Word2vec* [25], as explained in §2.2.1, is a neural network based method for transforming individual words into vectors of low dimensionality (it is low in comparison to $|words|$-dimensionality, e.g. 300), so that their distances reveal their semantic association. The motivation for the selection of the above methods is their ability to capture semantically relevant reviews beyond the trivial task of exact string matching, and their rich and domain-agnostic vocabulary.

4.3 Method-related Representation and Relevancy Formulas

Below we detail the method-related representation and the relevancy formula for each method. Note that the semantic similarity is computed differently between statistical and hand made dictionaries. Specifically, regarding the hand made dictionary, after expanding the query using Wordnet synonyms, antonyms and hypernyms, we compute the Jaccard Similarity between the sets W_q and each W_{r_k}.

For the statistical approach, we use the Word Movers Distance (WMD) [14] to calculate the similarity between a given q and each r_k. Computing WMD involves the pair-wise calculation of the "word travel cost" between the terms in the bags of words W_q and W_{r_k}. The objective is to minimize the cost function $c(i, j) = \|x_i - x_j\|_2$ defined as their Euclidean distance in the embedding space. The distance is computed by solving the following optimization problem.

$$\min_{T \geq 0} \sum_{i,j=1}^{n} T_{ij}\ c(i, j) \text{ subject to:}$$

$$\sum_{j=1}^{n} T_{ij} = f(i, W_q) \quad \text{for each} \quad i \in \{1, ..., n\}, \tag{1.1}$$

[2]https://deeplearning4j.org/docs/latest/deeplearning4j-nlp-word2vec

$$\sum_{i=1}^{n} T_{ij} = f(j, W_{r_k}) \quad \text{for each} \quad j \in \{1, ..., n\}$$

where T is a sparse flow matrix in which $T_{ij} \geq 0$ denotes how much the word i in W_q displaces in the embedding space towards word j in W_{r_k}. Specifically $f(i, W_q)$ is the normalized frequency of word i in the bag of words W_q computed as:

$$f(i, W_q) = \frac{o(i, W_q)}{\sum_{j=1}^{n} o(j, W_q)} \tag{2.1}$$

where $o(i, W_q)$ is the number of occurrences of word i in W_q and n the vocabulary size.

Baseline (BSL)**:** We consider as baseline the Jaccard Similarity between the sets W_q and W_{r_k}, i.e. we compute the score as:

$$S(q, r) = \max_{\forall r_k \in r} JaccardSim(W_q, W_{r_k}) \tag{3}$$

WordNet Query Expansion (WQE)**:** Here we construct the WordNet-based representation of q and each r_k by taking the union of the sets of WordNet extracted synonyms, antonyms and hypernyms of each word in W_q and W_{r_k} denoted by $wordNet(q)$ and $wordNet(r_k)$ respectively. Then we compute the score by computing the Jaccard Similarity:

$$S(q, r) = \max_{\forall r_k \in r} WNS(q, r_k) \tag{4}$$

where $WNS(q, r_k) = JaccardSim(wordNet(q), wordNet(r_k))$.

Word2vec ($W2V$)**:** This method exploits the Word2vec word embeddings available in the GoogleNews 300-dimensional pre-trained model[3]. Specifically, we get the Word2vec vector representations of all words in W_q and W_{r_k}, denoted by $word2vec(q)$ and $word2vec(r_k)$ respectively. Then we apply WMD between the embedded words of the two bags. The score is defined as:

$$S(q, r) = \max_{\forall r_k \in r} WMS(q, r_k) \tag{5}$$

where $WMS(q, r_k) = 1 - WMD_n(word2vec(q), word2vec(r_k))$ and WMD_n is the normalized distance by dividing with the max WMD over all reviews.

WordNet and Word2vec (CMB)**:** Here we combine the two previous methods through a weighted sum, as follows:

[3] `https://code.google.com/archive/p/word2vec/`

$$S(q,r) = w_{wn} * \max_{\forall r_k \in r} WNS(q, r_k) + w_{w2v} * \max_{\forall r_k \in r} WMS(q, r_k) \tag{6}$$

where $w_{wn}, w_{w2v} \in [0,1]$ and $w_{wn} + w_{w2v} = 1$.

Context-Words (CW: $W2V_{cw}$ and CMB_{cw}): Here we focus on the context-word problem mentioned in chapter 3. For example in the question *q3:"Has anyone reported a problem about cleanliness?"* every approach of ours that is based on Word2vec, returns in the top-2 results the review *"One person in our party had mobility issues and to get to the elevator you need to climb some stairs."*. This happens due to the synonymy of word *"problem"* in the question and the word *"issue"* in the review. However, both of the mentioned words, fall in the category of CW. This example makes clear the rationale for including the words *"problem"* and *"issue"* in the CW list. Let $W_{context}$ be the set of words that we have defined as CW and W_{stop} the set with the defined stop words. To eliminate the CW problem, we expand the query with duplicates of all query words that are neither stop words nor CW. Thereafter, in our example of *q3*, which will produce a $W_q = \{anyone, report, problem, cleanliness\}$, will become $W_q = \{report, problem, cleanliness, cleanliness\}$[4], increasing the importance of word *"cleanliness"* in the question. Hereafter, we denote the set of such words as $W_{informative}$. More formally, $W_{informative} = W_q \setminus (W_{stop} \cup W_{context})$. By increasing their frequency c_i in the query context, we increase the flow of each word $i \in W_{informative}$ when calculating the WMD.

Particularly, the addition of duplicate words increases $f(i, W_q)$ into $f(i, W_q)^+$ if the word $i \in W_{informative}$. Formally, we compute the increased frequency as:

$$f(i, W_q)^+ = \frac{o(i, W_q) + h}{\sum_{j=1}^{n} o(j, W_q)} \tag{2.2}$$

where h is a hyperparameter denoting frequency increment of the word i in document W_q due to $i \in W_{informative}$. In this way, we change the constraints of the WMD optimization problem and redefine it as follows:

$$\min_{T \geq 0} \sum_{i,j=1}^{n} T_{ij}\ c(i,j) \text{ subject to:}$$

$$\sum_{j=1}^{n} T_{ij} = \begin{cases} f(i, W_q)^+, & \text{if } i \in W_{informative} \\ f(i, W_q), & \text{otherwise .} \end{cases} \quad \text{for each} \quad i \in \{1, ..., n\}, \tag{1.2}$$

$$\sum_{i=1}^{n} T_{ij} = f(j, W_{r_k}) \quad \text{for each} \quad j \in \{1, ..., n\}$$

[4]Note that report is also in the CW list as described in §3.1.

Thus, all words $i \in W_{informative}$ gain more importance when calculating semantic similarity between the query q and a given review sentence r_k.

As it will be shown in the evaluation §5.3, at least in a domain specific dataset like FRUCE-v2, increasing the flow of all words $i \in W_{informative}$ in WMD improves the precision of the model. Note that this approach is only applicable in $W2V$ and CMB models, since it affects WMD but not Jaccard Similarity. We denote the new models as $W2V_{cw}$ and CMB_{cw} respectively and from now on we may refer to this technique as CW-aware WMD.

Statistical Query Expansion (SQE: $W2V_{sqe}$, and CMB_{sqe}): The query expansion approach described earlier in the WordNet WQE model, has no impact on the statistical similarity computation of model $W2V$. An alternative approach is to apply query expansion by exploiting Word2vec. To investigate this case below we describe a Word2vec-based expansion method. Figure 4.2 illustrates a running example of this method. In this example we can see how for the query "Is this hotel quiet", we compute the expanded bag of words $es^{\approx} = \{hotel, quiet, quiet, calm, tranquil, quieter, hushed, quiet\}$ using a greedy algorithm. For reasons of readability, Figure 4.2 shows only some of the expansion terms.

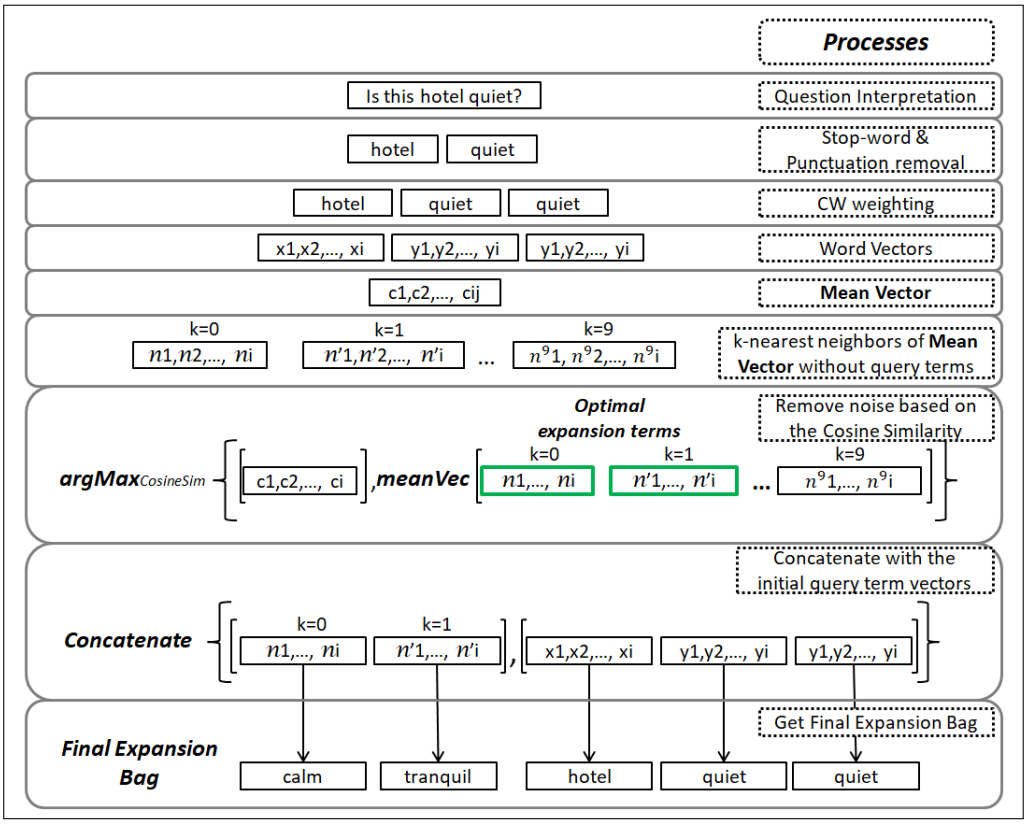

Figure 4.2: Statistical Query Expansion running example.

In more details, given a query q we compute the mean vector $\vec{v_q}$ of all query terms vector representations. By using $\vec{v_q}$ we retrieve terms that are close to the

whole context of the question and thus reducing possible ambiguities of single words. We denote as *cets* the ordered set of candidate expansion terms of the k-nearest neighbors of $\vec{v_q}$ based on the cosine similarity, minus all terms that are also in q. To further reduce noisy context from *cets*, we start a greedy iterative process where 'non-relevant' expansion terms in *cets* are removed to create the approximation $es^{\approx}$ to the optimal expansion set es^*. For example the initial expansion set consists of the words {*calm, tranquil, quieter, placid, serene, hushed, neighbor_Abdul_Shash,quiet*}, while the final expansion set $es^{\approx}$ consists of the subset {*calm, tranquil, quieter, hushed, quiet*}. The impact of our noise removal approach, managed to filter out terms such as *placid* and *serene* which have a different sense than that of the word *quite* in our context, as well as Named Entities like *neighbor_Abdul_Shash*, resulting to a more fine grained query expansion terms set.

Formally, let es_j be the j_{th} set of all possible $2^{|cets|}$ subsets of *cets*. Our objective is to find the optimal es_j, denoted as es^*, whose mean vector $\vec{es^*}$ maximizes the cosine similarity S with the query vector $\vec{v_q}$, i.e.:

$$\vec{es^*} = \arg\max_{\vec{es_j}} S(\vec{v_q}, \vec{es_j}), \ where \quad 1 \leq j \leq 2^{|cets|} \tag{7.1}$$

Note that each es_j can be of indefinite length between 1 and k, where k denotes the number of the k-nearest neighbors of $\vec{v_q}$ in the embedding space i.e. the candidate expansion terms. Since there is no possible way to calculate S for all possible subsets es_j efficiently, we follow a more efficient greedy approach. Since *cets* contains candidate expansion terms ordered by their cosine similarity with $\vec{v_q}$, it is reasonable to trust more terms that are ordered first. Let $cets_i$ and $\vec{cets_i}$ denote the i_{th} term of *cets* and its Word2vec vector respectively. Initially we compute the cosine similarity *sim* between $\vec{v_q}$ and the $\vec{cets_1}$ (the closest term to $\vec{v_q}$ based on the cosine similarity) and add the term $cets_1$ to the expansion set $es^{\approx}$. Then, we continue with the second term $cets_2$, compute the mean vector of the terms in $es^{\approx} \cup cets_2$ and then calculate the similarity sim' of this mean vector with the $\vec{v_q}$. If $sim' > sim$ then we add the term $cets_2$ to the expansion set $es^{\approx}$ and set $sim = sim'$, else we ignore this term. Then we continue with the next term in *cets* and so on. Formally, the described procedure approximates the optimal $\vec{es^*}$ with $\vec{es^{\approx}}$, the mean vector of $es^{\approx}$, that is computed using the following method:

$$\vec{es^*} \approx \vec{es^{\approx}} = \frac{\vec{cets_1} + \sum_{i=2}^{k}(\vec{cets_i} * h(\vec{cets_i}))}{H} \tag{7.2}$$

where k is the number of the nearest (in the embedding space) neighbours of $\vec{v_q}$, i.e. the candidate expansion terms[5], while $h(\vec{cets_i})$ is the following function:

[5]The value for k can be set based on latency constraints (more candidates require more time to be filtered).

$$h(\vec{cets_i}) = \begin{cases} 1, & \text{if } sim' > sim \\ 0, & \text{otherwise .} \end{cases} \tag{7.3}$$

Where sim is the cosine similarity between the query vector $\vec{v_q}$ and our expansion vector approximation $\vec{es^{\approx}}$ up until the $i_{th} - 1$ nearest neighbor and sim' the cosine similarity up until the i_{th} neighbor. We divide the sum of vectors with H, which stands for the number of times that $h(\vec{cets_i})$ got the value 1, for calculating the mean vector only of those terms that improved the cosine similarity with $\vec{v_q}$.

Finally, we concatenate q and $es^{\approx}$ to produce the expanded query bag of terms q_{sqe} to feed the $W2V$ or CMB models. We denote the new models as $W2V_{sqe}$ and CMB_{sqe} respectively. Note that when applying both CW-aware WMD and SQE, the first affects the second one, since CW directly affects $\vec{v_q}$, the mean vector representation of the query, leading to a different set of candidate expansion terms $cets$.

Chapter 5

Evaluation

In this Chapter, §5.0.1 provide the experimental settings details, §5.1 discusses the runtime performance of the proposed methods, while §5.2, §5.3 and §5.4 discuss the evaluation results over the FRUCE-v1, FRUCE-v2 and WebAP collections respectively.

5.0.1 Experimental Setting

As regards implementation we used the *Deeplearning4j*[1] library for Word2vec, and the *MIT Java Wordnet Interface (JWI)*[2] for Wordnet.

The timings that are reported below, were performed using a commodity laptop with a quad-core 2.60 GHz i7, with 6MB cache and 16 GB RAM.

We find the optimal weights for CMB and it's variations using grid search over FRUCE-v1($w_{wn} = 0.7$ and $w_{w2v} = 0.3$) and FRUCE-v2 ($w_{wn} = 0.6$ and $w_{w2v} = 0.4$) and we use the later ones for the evaluation over the WebAP collection. We use the AveP measure (see table 5.7 for definition) as indicator of whether the weights are optimal. We empirically set the hyperparameters h and k of the CW-aware WMD and SQE techniques to $h = 1$ and $k = 10$ respectively. Also note that the evaluation over FRUCE-v1 is presented for consistency reasons. However the collection is extremely small, thus it is suggested to pay more attention to the results presented in §5.3 and §5.4 than to the results presented in §5.2.

5.1 Run-time performance

Table 5.1 shows that the proposed approaches can be applied in real-time. For each method we measured and computed the average timings for scoring one review per query ($AvgRQ$), and for all reviews (7,475 distinct reviews) of 5 hotels in the user focus for all queries (line: $AvgThreshold$). As expected, all variations of CMB are

[1] `https://mvnrepository.com/artifact/org.deeplearning4j/deeplearning4j-core/0.9.1`

[2] `https://mvnrepository.com/artifact/edu.mit/jwi/2.2.3`

Table 5.1: Timings in *ms* over 7475 reviews scraped from booking.com. **AvgRQ** is the average time for scoring one review per query, and **AvgThreshold** is the average time for scoring all reviews of 5 hotels for a query

Method	WQE	$W2V$	$W2V_{cw}$	$W2V_{sqe}$	$W2V_{cw+sqe}$	CMB	CMB_{cw}	CMB_{sqe}	CMB_{cw+sqe}
AvgRQ	11	10	10	22	24	19	19	31	34
AvgThreshold	1099	1007	1032	2164	2423	1945	1923	3034	3393

more expensive. Further, SQE doubles the execution times due to the increased number of distinct terms in WMD calculations. On the other hand, CW-aware WMD does not change the execution time since the duplicate terms affect only the frequency of the terms.

5.2 Evaluation over the Collection FRUCE-v1

To understand the situations that may arise and to compare the methods presented, we constructed a small evaluation collection. It consists of 40 hand crafted user reviews/comments $c_1, ..., c_{40}$ about hotels, and 2 manually crafted queries, i.e. q_1, q_2, related to the topic of *noise*. The complete list of reviews is web accessible [3]. The queries are shown below:
q_1 = "Has anyone reported a problem about noise?",
q_2 = "Is this hotel quiet?".

For the needs of the evaluation, we manually judged each review, based on their relevancy. Specifically, each review c_i is labeled with 1 if it is relevant, and with 0 otherwise. The ratio of relevant/irrelevant (c_i, q_i) pairs is 1/3.

5.2.1 Quality.

The two queries that were used in the evaluation collection along with the 5 most relevant reviews for each one of them by applying the method IV are show in Table 5.5. We observe that noisy words (e.g. *hotel*) that carry no real information about the users question may harm the retrieval results, as we described in Chapter 3, in the Context Words challenge. Thus we proposed methods for tackling such issues by applying the CW-Aware-WMD presented in chapter 4.3. To further see about the performance of this method, see sections 5.3 and 5.4.

For evaluating the quality of the above methods, we measured the $mean\ R-Precision$ and $mean\ AveP$ over the two hand crafted queries q_1 and q_2 for various weights combinations of IV, choosing the model that achieved the highest $mean\ AveP$. Note that, for all the evaluation metrics, instead of using for each query a specific recall level, we chose the number of its relevant comments denoted as R, in order to capture how well each method performs over different relevant/irrelevant ratios.

[3] at `http://www.ics.forth.gr/isl/sar/resources/dataset/fruce`

R	ID	Review Text	Rel
q1: Has anyone reported a problem about noise?			
1	c_5	There is too much noise because of the airport.	1
2	c_2	The music of the bar is too loud.	1
3	c_7	The hotel is great. I love the pool, however it is a bit noisy.	1
4	c_{30}	Nothing but good things to say about this hotel - excellent in all possible ways.	0
5	c_6	Appliances are noisy (air-condition, refrigerator).	1
q2: Is this hotel quiet?			
1	c_8	The hotel is very calm, I enjoyed sleeping.	1
2	c_1	The road is very noisy.	1
3	c_9	Camp Nou is more calm. (irony)	1
4	c_6	Appliances are noisy (air-condition, refrigerator).	1
5	c_7	The hotel is great. I love the pool, however it is a bit noisy.	1

Table 5.2: Indicative Query-Oriented Review Retrieval over FRUCE-v1. Where Rel: query-review relevance, ID: review id and R: rank of review in system's response.

Method	Mean AveP	Mean R-Precision
I	0.05	0.05
II	0.398	0.449
III	0.366	0.4
IV	**0.569**	**0.649**

Table 5.3: Mean Average Precision of methods I-IV.

Method	Total time (ms)	Aver. time (ms)
I	141	3
II	797	19
III	**47**	**1**
IV	546	13

Table 5.4: Time for computing the score of 40 reviews for each method.

Note that methods II and III correspond to the pairs $(w_{wN} = 1.0, w_{w2v} = 0.0)$ and $(w_{wN} = 0.0, w_{w2v} = 1.0)$ respectively. In our case the maximizing weights were found to be $w_{wN} = 0.7$ and $w_{w2v} = 0.3$ with $mean\ AveP = 0.569$ and $mean\ R - Precision = 0.649$. The corresponding scores for method II were $mean\ AveP = 0.398$ and $mean\ R - Precision = 0.449$, while method III achieved $mean\ AveP = 0.366$ and $mean\ R - Precision = 0.4$ (note that the precision of Word2vec-based methods in analogous challenges, as reported in [24], is around 55%, i.e. not much different from what he have observed in our setting). Finally, IV outperforms both II, III, while II slightly outpoints III. As expected, all of the above models outperformed our baseline ($mean\ AveP = 0.05$ and $mean\ R - Precision = 0.05$) as shown in Table 5.3. We have to stress though that the results of methods II and IV could be further improved by combining other thesaurus with WordNet or an updated version of WordNet, since WordNet currently fails to provide synonyms, hypernyms and antonyms about many words. Further, since we consider all possible senses of a word when applying the WordNet based approach, we are possibly introducing a lot of noise (wrong terms) in the $wordNet(q)$ and $wordNet(r_{ij})$ set, which could be avoided with proper sense identification methods.

In addition, we plot a 2D diagram for each of the three models II, III, IV (baseline excluded), where the *y-axis* represents the computed score for (r_i, q_i), and the *x-axis* indicates its true binary relevance. The plots are shown in Figure 5.1. As we can see, the points are not separable by a threshold in any of the figures (parallel line to *x-axis*). However, it is obvious that the IV approach clearly improves the separation, preserving higher scores to the true relevant reviews, like III, and lower scores to non relevant ones, like II.

5.3 Evaluation over the Collection FRUCE-v2

To better understand the factors that affect the effectiveness of the above methods, we created a new version of FRUCE-v1[4], namely FRUCE-v2[5], which now consists of 139 hand crafted user reviews/comments related to hotels. Moreover we created the following 6 manually crafted queries related to the topics of *noise*, *cleanliness*, and *politeness*:

[4] `http://www.ics.forth.gr/isl/sar/resources/dataset/fruce`
[5] `http://www.ics.forth.gr/isl/sar/resources/dataset/fruce-v2`

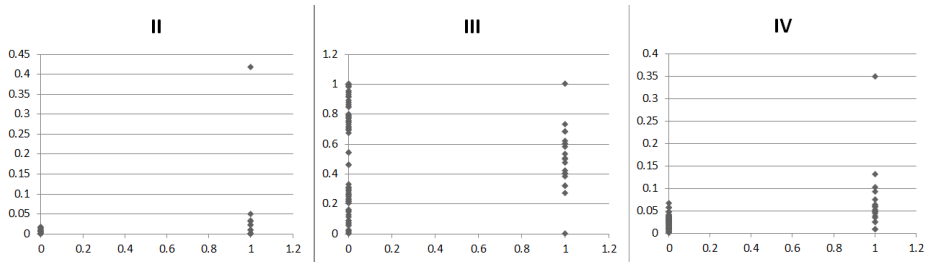

Figure 5.1: Distribution of query-review pairs as a function of their calculated (floating point) and true binary relevance score for methods II, III, IV.

q_1 = "Has anyone reported a problem about noise?",
q_2 = "Is this hotel quiet?".
q_3 = "Has anyone reported a problem about cleanliness?",
q_4 = "Has anyone complained about the bed linen?"
q_5 = "Is the personnel polite?",
q_6 = "Is the hotel staff helpful?"
For the needs of the evaluation, for each query we manually crafted 40 reviews, 10 relevant and 30 irrelevant. Each review c_i is labeled with 1 if it is relevant, otherwise with 0. The ratio of relevant/irrelevant (c_i, q_i) pairs is 1/3 for each query, in order to simulate that from the whole collection only a few comments are usually related to a users question. The collection comprises 240 pairs in total. The judging was not exchaustive, i.e. not all reviews have judgement for all queries. Table 5.5 presents some indicative (c_i, q_i) pairs as retrieved by the system, whereas table 5.6 present a general set of indicative (c_i, q_i) pairs for various queries q_i of the collection.

Table 5.5: Indicative FRUCE-v2 Queries and Reviews Retrieval (R: rank of review in system's response, ID: the review id, and Rel: query-review relevance)

R	ID	Review Text	Rel
q1: Has anyone reported a problem about noise?			
1	c_5	There is too much noise because of the airport.	1
2	c_2	The music of the bar is too loud.	1
q2: Is this hotel quiet?			
1	c_8	The hotel is very calm, I enjoyed sleeping.	1
2	c_1	The road is very noisy.	1

5.3.0.1 Quality.

Regarding the effectiveness of the introduced methods, we computed the following evaluation metrics: Precision@2, R-Precision, AveP, Recall@R, Bpref, nDCG and MRR. For reasons of self-containedness, Table 5.7 shows how these metrics are defined. Note that Recall is not mentioned since its value is the same with R-Precision that is reported.

Table 5.6: Indicative FRUCE-v2 Queries and Review Pairs. (*ID*: the review id, and *Rel*: query-review relevance)

ID	Review Text	Rel
q1: Has anyone reported a problem about noise?		
c_5	There is too much noise because of the airport.	1
c_2	The music of the bar is too loud.	1
c_3	The rooms are totally soundproof.	1
c_3	The hotel is great. I love the pool, however it is a bit noisy.	1
c_{12}	The hotel was very close to high quality restaurants.	0
c_{13}	The hotel's bar serve premium whiskey.	0
c_{14}	The building is very old.	0
q2: Is this hotel quiet?		
c_8	The hotel is very calm, I enjoyed sleeping.	1
c_9	Camp Nou is more calm.	1
c_{10}	I didn't sleep well because the hotel is very close to Camp Nou.	1
c_3	The rooms are totally soundproof.	1
c_{22}	Expensive breakfast.	0
c_{24}	Room size could be slightly bigger.	0
c_{25}	Location isn't the best for tourists but it is close to train stations to get around.	0
q3: Has anyone reported a problem about cleanliness?		
c_{41}	The furnitures were quite dusty.	1
c_{42}	The bed linen has some smudges on it.	1
c_{43}	The bed was very comfortable, however its linen was quite dirty.	1
c_{45}	The toilet was not descaled and had a terrible smell.	1
c_{54}	The hotel delivered beyond our expectations.	0
c_{55}	Very nice food and good location.	0
c_{56}	The chamber was a little bit to small but for this price can't expect more!	0
q4: Has anyone complained about the bed linen?		
c_{43}	The bed was very comfortable, however its linen was quite dirty.	1
c_{49}	The bed sheets were smelly, probably not washed well.	1
c_{81}	The pillowcase had a couple of holes from cigarette burners.	1
c_{83}	The linen of the bed was too itchy.	1
c_{64}	Huge toilet area.	0
c_{55}	Very nice food and good location.	0
c_{66}	Acceptable quality of breakfast.	0

Table 5.7: Evaluation metrics definitions.

Notation	**Definition**
Q	the set of queries of the evaluation collection
R	the set of relevant reviews for the query at hand $q \in Q$
RL	the set of retrieved reviews for the query at hand $q \in Q$ by the system
r	the rank of review c at hand $c \in RL$
$isrel(r)$	equals 1 if review at rank r is relevant, 0 otherwise
rel_i	the true graded relevance of review at rank i wrt the collection
REL	the *ordered list* of relevant reviews in the collection
$rank_i$	the rank of the 1st relevant review of the i_{th} query $q \in Q$
$\|Q\|$	the total number of queries
$count(r)$	the number of relevant reviews in top r ranks of RL
R-Precision	$\frac{\|R \cap RL\|}{\|RL\|}$ (same as Recall at position $\|R\|$)
AveP	$\frac{1}{\|R\|} \sum_r (isrel(r) \frac{count(r)}{r})$
Bpref	$\frac{1}{\|R\|} \sum_r isrel(r)(1 - \frac{min(\|R\|, r - count(r))}{min(\|R\|, \|N\|)})$
MRR	$\frac{1}{\|Q\|} \sum_{i=1}^{\|Q\|} \frac{1}{rank_i}$
nDCG	$\frac{DCG_{\|R\|}}{IDCG_{\|R\|}} = \frac{\sum_{i=1}^{\|R\|} \frac{2^{rel_i} - 1}{log_2(i+1)}}{\sum_{i=1}^{\|REL\|} \frac{2^{rel_i} - 1}{log_2(i+1)}}$

For calculating the aforementioned metrics, we used the condense result lists for each query, i.e. the retrieved results only for those (c_i, q_i) pairs for which we have human judgements. This allows to deal with our incomplete collections (not all possible (c_i, q_i) have true relevance labels).

Now Table 5.8 shows the evaluation results (average value of metrics over all queries), while Figure 5.2 shows the same results graphically.

The CW-aware WMD seems to have a positive effect on the recall, precision and ranking quality of the models, as shown by the differences of R-Precision, AveP, Recall, Bpref and nDCG metrics between the pairs of models $W2V$-$W2V_{cw}$ and CMB-CMB_{cw}. However, this does not hold for the top-2 results. In the $W2V$-$W2V_{cw}$ cases the Precision@2 between the models remains the same, whereas the MRR is slightly better (+0.013) when applying the CW-aware WMD. In CMB-CMB_{cw} case though the Precision@2 and MRR drops (-0.083 and -0.028 respectively) when applying the same technique. Note that in the last case, the WordNet-based method has more weight in the final score (0.6 versus 0.4). As regards the SQE technique, in most cases it affects negatively the models in terms of recall, precision and ranking quality, compared to the $W2V$-$W2V_{sqe}$ and CMB-CMB_{sqe} methods. It is interesting to note thought that when both SQE and CW-aware WMD are applied, we have a significant improvement to all evaluation metrics (see $W2V_{cw+sqe}$, CMB_{cw+sqe}). This observation indicates that the duplicates of informative words, as described in the CW-aware method, affects positively the produced mean vector when applying SQE and leads to a better set of expansion terms. Finally, note that all models outperformed the baseline.

Table 5.8: Evaluation results over FRUCE-v2

Method	Precision@2	R-Precision	AveP	Bpref	nDCG	MRR
BSL	0.583	0.266	0.243	0.253	0.396	0.666
WQE	0.833	0.466	0.401	0.43	0.625	0.916
$W2V$	0.833	0.549	0.476	0.506	0.865	0.875
$W2V_{cw}$	0.833	0.583	0.511	0.538	0.881	0.888
$W2V_{sqe}$	0.750	0.550	0.464	0.498	0.844	0.875
$W2V_{cw+sqe}$	**1.00**	0.633	**0.598**	0.606	**0.933**	**1.00**
CMB	0.833	0.583	0.501	0.536	0.870	0.916
CMB_{cw}	0.750	0.616	0.530	0.566	0.878	0.888
CMB_{sqe}	0.666	0.566	0.469	0.508	0.841	0.875
CMB_{cw+sqe}	0.916	**0.650**	**0.598**	**0.616**	0.923	**1.00**

5.4 Evaluation over the Collection WebAPP

Although our context if CFS and the process described in §4.1, meaning that the set of reviews that will be considered will not be big (since it will contain reviews referring only to the objects of the focus, or first bucket), we decided to evaluate the retrieval of the more relevant reviews in a larger collection in order to investigate

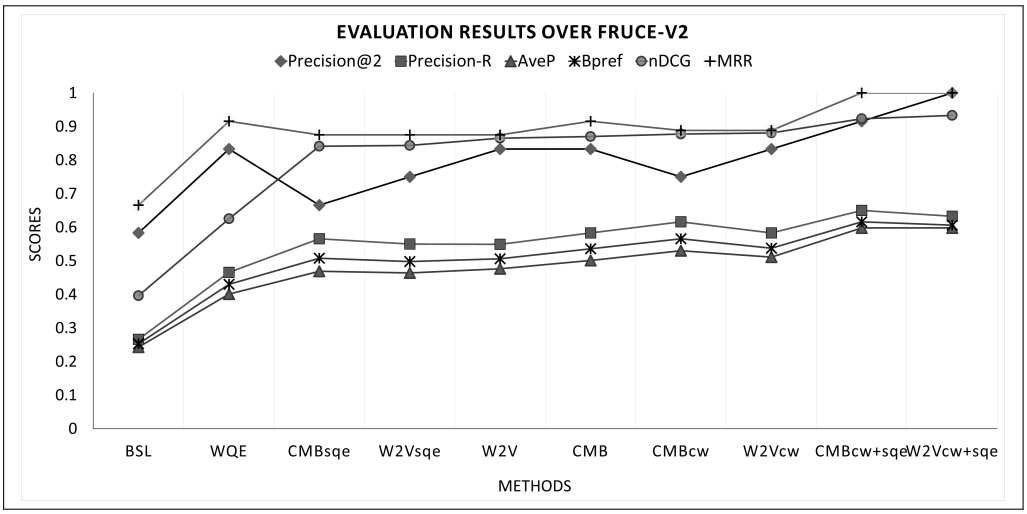

Figure 5.2: Evaluation results over FRUCE-v2

how it behaves, and to obtain results that are comparable to other words in the literature.

For this purpose we evaluated the proposed methods over the real dataset called WebAP [12], that consists of 82 TREC queries based on the 2004 TREC Terabyte Track GOV2[6] collection and has 8,027 manually annotated passages in total graded with a level of relevance between a scale of 4 levels, perfect (lvl. 4), excellent (lvl.3), good (lvl.2) and fair (lvl.1). Among all the annotated passages for all queries, 43% of them are perfect answers, 44% are excellent, 10% are good and the rest are fair answers. Table 5.9 shows some statistics as regards the number and the length of the passages of WebAP. More specifically the first 4 rows refer to passages that are annotated as relevant, while the last row refers to the not annotated ones.

Table 5.9: WebAP collection statistics

Relevancy Level	Num of Passages	Avg words	Min words	Max words
Perfect (lvl.4)	2278	27.8	1	137
Excellent (lvl.3)	2133	29.0	1	267
Good (lvl.2)	792	25.3	1	115
Fair (lvl.1)	106	29.4	1	106
Not annotated	33395	92.6	1	10852

The collection contains documents and only some of the passages of these documents are annotated. It is not clear whether the passages that are not annotated are irrelevant or not judged.

For this reason we constructed our dataset by retrieving from WebAP only the

[6] www-nlpir.nist.gov/projects/terabyte/

judged query-passage pairs along with their level of relevance. Note that 2 out of 82 queries (queries with id 715 and 752) are not associated with any passage with any level of relevance, thus we filter them out, since there is no known correct answer for those queries. That results to a collection of 80 queries with 8,027 annotated passages in total and an average of about 97 annotated passages per query. However, note that the distribution of annotated passages per query is not uniform, meaning that there are queries with too few annotated passages and others with too many.

Since the derived WebAP dataset contains only graded relevant (c_i, q_i) pairs, it is reasonable to evaluate the methods using only the nDCG metric, testing whether each model returns relevant comments in the order of their graded relevance, i.e. more relevant comments should appear higher in the results list. However, in order to compute other metrics also, we made the assumption that only the comments that are annotated as perfect ($relevance = 4$) are considered relevant to a specific query. The rationale for this decision is that we focus on *high quality review retrieval*, and we want to see which methods are better on that. Apart from that, since it is more realistic for a collection to contain fewer relevant than irrelevant passages, the above selection gives such statistics: 44% relevant over 66% irrelevant.

We also filtered out those queries which are associated only with passages with perfect level of relevance. Since we use the condensed result lists in the evaluation, i.e. the subset of all pairs such that each pair is associated with a human judgment, such a subset of the systems response for each query cannot return any other passage than a perfect one. To this end, such cases would incorrectly improve the performance.

In the same manner, we also filtered out those queries that do not have any perfect answer, since that means that these queries would not have any relevant answer to be considered in the computation of the metrics.

These modifications reduced the final set of queries to 72 queries instead of the original 80 queries. Table 5.10 shows the statistics of the derived collection after the aforementioned modifications. The table also contains the respective statistics when we assume both perfect ($relevance = 4$) and excellent ($relevance = 3$) passages as relevant. Table 5.11 shows some indicative query-passage pairs alongside with their level of relevance. Note that the example of pairs are chosen to show that not all queries have passages with all levels of relevance (e.g. q709 have only perfect and excellent judged passages) and thus our modifications are necessary. Also note that the passage *id* presented in the first column of the table is not provided by WebAP. We set a unique *id* for each passage. Bellow we present a span of a judged document with its associated judged passage (text enclosed in EXCEL tags). For readability reasons text is omitted. We present the full text only for the judged passage, in order to keep its consistency.

<TARGET_QID>701</TARGET_QID>
<ORIGINAL_DOCNO>GX068−83−6288039</ORIGINAL_DOCNO>

```
<TEXT>
<NONE>
<SENTENCE>Committee on Resources Subcommittee... </SENTENCE>
<SENTENCE>Opening Statement:</SENTENCE>
<SENTENCE>Chairman Hansen, Members of Congress... </SENTENCE>
<SENTENCE>We strongly believe that Pennsylvania's... </SENTENCE>
<SENTENCE>For those of you unfamiliar with ... </SENTENCE>
<SENTENCE>History of the Birth of the Oil Industry... </SENTENCE>
<SENTENCE>Author: Barbara Zolli, PHMC Drake... </SENTENCE>
<SENTENCE>Presenter: Judith Downs, Vice... </SENTENCE>
<SENTENCE>Introduction</SENTENCE>
</NONE>
<EXCEL>
<SENTENCE>Pennsylvania oil industry pioneers carried their
knowledge, tools, and skills to other parts of the country
and the world.</SENTENCE>
<SENTENCE>Newer fields opened around the globe from Ohio
to Russia! But, after the Texas gusher wells came in at
Spindletop in 1901, world attention shifted away from
the Oil Creek Valley in Pennsylvania.</SENTENCE>
</EXCEL>
    .
    .
    .
</TEXT>
```

It is interesting to note here, that if we had used all available text as well as the unjudged text as non relevant (like other related work), then we should be able to score sentence "History of the Birth of the Oil Industry..." as non relevant for the query "q701: describe history of oil industry", despite the big overlap between the two aforementioned utterances terms, since it is not enclosed in an EXCEL tag. This is quite more challenging than our task, i.e. finding mentions about a simple utterance in a user review. That explains the choice of treating the collection in a different manner than other related works in the same collection.

We then apply the retrieval on the set of judged only passages in an attempt to rank them and most importantly to retrieve only perfect passages in the top two ranked ones. Table 5.12 shows the evaluation results of our methods (average value of metrics over all queries) while Figure 5.4 shows the same results graphically.

5.4.1 Quality.

Note that for the CW-aware approach we constructed the set of context words by checking the queries and identifying words like describe, state, etc., that do not have any information regarding the query, and are used in the flow of speech. For example the query "describe history oil industry" uses the word "describe"

Table 5.10: WebAP collection statistics after the modifications.

Levels considered relevant	$\{4\}$	$\{4,3\}$	$\{4,3,2\}$
Num of queries	72	68	31
Num of pairs	7305	6872	3772
Total Num of passages	7035	6872	3772
Avg Num of passages per query	101.45	101	121.67

Table 5.11: Indicative WebAP Queries and Passage Pairs. (*ID*: the paasage id, and *Rel*: query-passage relevance)

ID	Passage Text	Rel
q709: limits regulations concerning jockey weight horse racing		
560	If a jockey weighs in two or more pounds less than the weight at which he weighed out, the horse shall be disqualified.	4
464	Five (5) pounds is the limit of overweight any horse is permitted to carry	3
q722: ways iran support terrorism		
1469	With regard to Islamic Jihad, Iran has a slightly closer relationship (hardliners in Iran).	1
1489	Since the outbreak of the intifadah, support has intensified for Palestinian groups that use violence against Israel.	2
1536	Rezai said that, "Iran will continue its campaign against Zionism until Israel is completely eradicated."	3
1582	Iran continues its firm support for Hizballah, HAMAS, and the Palestine Islamic Jihad.	4
q729: revelations whistle blowers concerning department defense		
2598	And yet, this knowledge was being withheld from the American Public.	3
2600	He was unsuccessful in getting the Army to conduct live-fire tests to expose these weaknesses.	3

Table 5.12: Evaluation results over WebAP. $rels = \{4\}$, $nonrels = \{1,2,3\}$

Method	Precision@2	R-Precision	AveP	Bpref	nDCG	MRR
BSL	0.423	0.376	0.245	0.279	0.648	0.517
WQE	0.437	0.395	0.260	0.288	0.821	0.562
$W2V$	0.451	0.423	0.277	0.305	0.894	0.567
$W2V_{cw}$	0.402	0.398	0.253	0.282	0.886	0.511
$W2V_{sqe}$	0.416	**0.438**	**0.294**	**0.323**	**0.895**	0.574
$W2V_{cw+sqe}$	0.402	0.406	0.258	0.287	0.887	0.510
CMB	0.451	0.415	0.279	0.306	0.894	0.572
CMB_{cw}	**0.472**	0.399	0.271	0.296	0.892	**0.589**
CMB_{sqe}	0.395	0.413	0.275	0.304	0.892	0.535
CMB_{cw+sqe}	**0.472**	0.399	0.271	0.296	0.892	**0.589**

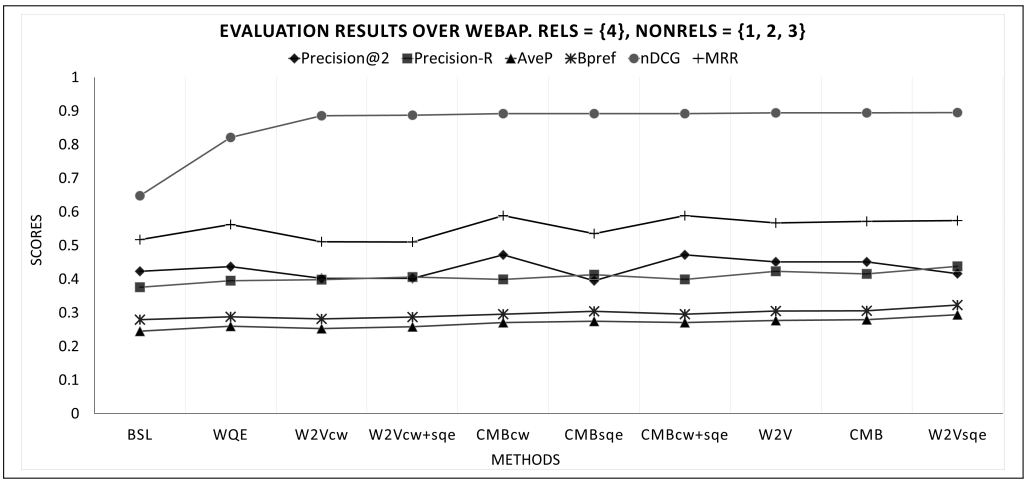

Figure 5.3: Evaluation results over WebAP when only *Perfect* are considered *Relevant*

as a command to the system, whereas the words "history", "oil" and "industry" constitute the focus of the question. Having said that, we created the following set of CW for this dataset $W_{context} = \{describe, state, give, information\}$. However, note that since WebAP is an open-domain (GOV data) collection, it is not straight forward how to manually or automatically choose the appropriate context words. This assumption is also validated by the gathered results.

Specifically, the CW-aware WMD affects model $W2V$ negatively in every evaluation metric (see model $W2V_{cw}$ Table 5.12). On the other hand SQE over model $W2V$ seems to have a positive impact over all evaluation metrics except Precision@2 (see model $W2V_{sqe}$ Table 5.12). When both CW-aware WMD and SQE are applied, the performance of model $W2V$ drops. These observations also hold for the CMB model, with two differences. First, the SQE technique does not boost the model to a better performance. Second, CW-aware WMD improves only the first ranks of the result list (check Precision@2 and MRR for CMB_{cw} in Table 5.12). Finally, models CMB_{cw} and CMB_{cw+sqe} perform identically. Having said that, it seems that CW-aware WMD is not easily applicable over open-domain datasets, since words may have different meanings over different domains. For example the word "state" which is currently in the $W_{context}$ set representing the verb "state", appears in the passages dataset also in the form of "people puerto rico state", denoting a country. In this case, we will devalue this passage since state is included in the context words set, ranking lower a relevant comment and affecting the performance of the CW-aware method. Additionally, CW of low quality are responsible for introducing noise to the SQE, leading to the construction of a low-quality query vector. This is observable in the performance differences between $W2V_{sqe}$ and $W2V_{cw+sqe}$, and CMB_{sqe} and CMB_{cw+sqe} in Table 5.12, where the addition of CW before the SQE lowers the performance in most of the evaluation metrics in both cases.

Table 5.13: Evaluation results over WebAP. $rels = \{4, 3\}$, $nonrels = \{1, 2\}$

Method	Precision@2	R-Precision	AveP	Bpref	nDCG	MRR
BSL	0.779	0.609	0.520	0.413	0.650	0.858
WQE	0.786	0.778	0.664	0.458	0.818	0.865
$W2V$	0.786	0.803	0.693	0.454	0.889	0.879
$W2V_{cw}$	0.764	0.797	0.683	0.461	0.881	0.850
$W2V_{sqe}$	0.801	0.810	0.702	0.457	**0.891**	0.879
$W2V_{cw+sqe}$	0.772	0.795	0.684	0.461	0.882	0.848
CMB	**0.830**	**0.814**	**0.703**	**0.465**	0.889	0.890
CMB_{cw}	0.823	0.804	0.694	0.460	0.887	**0.900**
CMB_{sqe}	0.801	0.811	0.701	0.458	0.887	0.880
CMB_{cw+sqe}	0.823	0.804	0.694	0.460	0.887	**0.900**

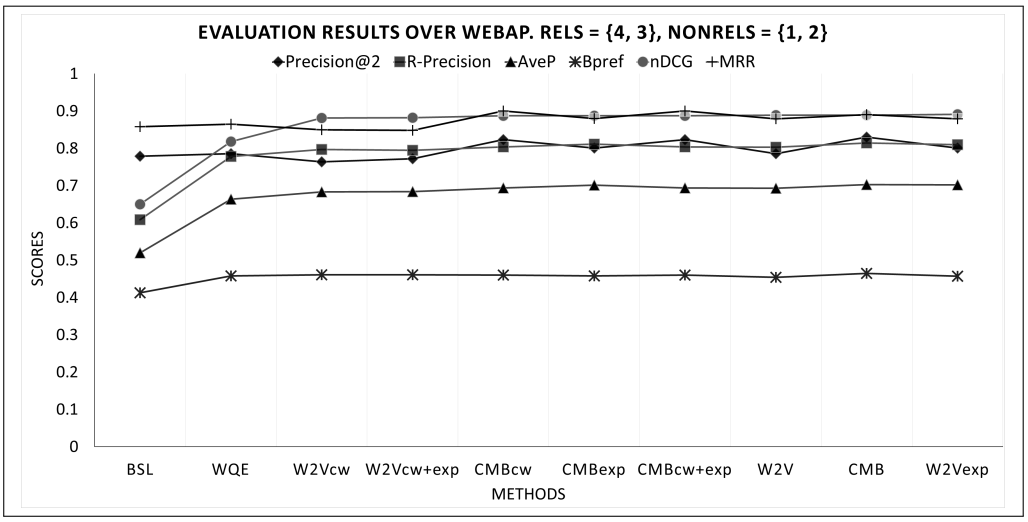

Figure 5.4: Evaluation results over WebAP when *Perfect* and *Excellent* are considered *Relevant*

Since the consideration of only "perfect" hits as relevant, can be considered as very strict, we peformed experiments also for the case where perfect and excellent are considered as relevant. The results are shown in Table 5.13 and Figure 5.4. We can see that Precision@2 now raises to 83% for CMB. Moreover, as expected all of the evaluation metrics are improved, since the ratio between relevant and irrelevant passages, now favors the relevant ones (87% relevant over 23% irrelevant). Finally, we experimented also with the case only fair comments are considered irrelevant. In that setting, as expected, the values of all metrics are increased. The results are show in Table 5.14.

Table 5.14: Evaluation results over WebAP. $rels = \{4, 3, 2\}$, $nonrels = \{1\}$

Method	Precision@2	R-Precision	AveP	Bpref	nDCG	MRR
BSL	0.887	0.582	0.548	0.370	0.609	0.946
WQE	0.870	0.824	0.774	0.428	0.779	0.946
$W2V$	**0.903**	**0.938**	**0.888**	0.451	0.874	**0.951**
$W2V_{cw}$	**0.903**	0.927	0.871	**0.486**	0.865	0.919
$W2V_{sqe}$	**0.903**	0.936	0.886	0.434	**0.876**	0.919
$W2V_{cw+sqe}$	0.870	0.925	0.869	0.480	0.867	0.919
CMB	**0.903**	0.937	0.886	0.450	0.874	0.946
CMB_{cw}	0.887	0.929	0.874	0.453	0.871	0.946
CMB_{sqe}	0.854	0.937	0.885	0.436	0.871	0.946
CMB_{cw+sqe}	0.887	0.929	0.874	0.453	0.871	0.946

5.5 Synopsis of Experimental Findings and Comparison to Related Work

In brief we have seen that the proposed method achieves precision 100% in top-2 in a small closed domain collection, FRUCE-v2, while in the large government domain collection WebAP, the same method achieves precision 47% in top-2, even when we assume relevant only the perfect answers, and note that this percentage is higher than the state-of-the-art (i.e. *Word Movers Distance* achieved 45%). If we assume as relevant the perfect and excellent answers, then we achieve 82% (over 78% that *Word Movers Distance* achieved). The approach presented in [2] performed worse in MRR (51%) and Precision@1 (35%) in the same collection. However, since the evaluation setting differs due to training, different data preparation (they have also used unjudged passages as non-relevant) and different testing methodology (using passages from the first 100 results based on a BM25 search, including a correct answer passage in the last rank if BM25 fail to retrieve any correct results for the query at hand) a direct comparison cannot be considered accurate.

Figures 5.5, 5.6 and 5.7 show the derivation history of the methods described in §4.3 and their corresponding values for Precision@2 and MRR when using FRUCE-v2 and WebAP for the two assumed scenarios respectively.

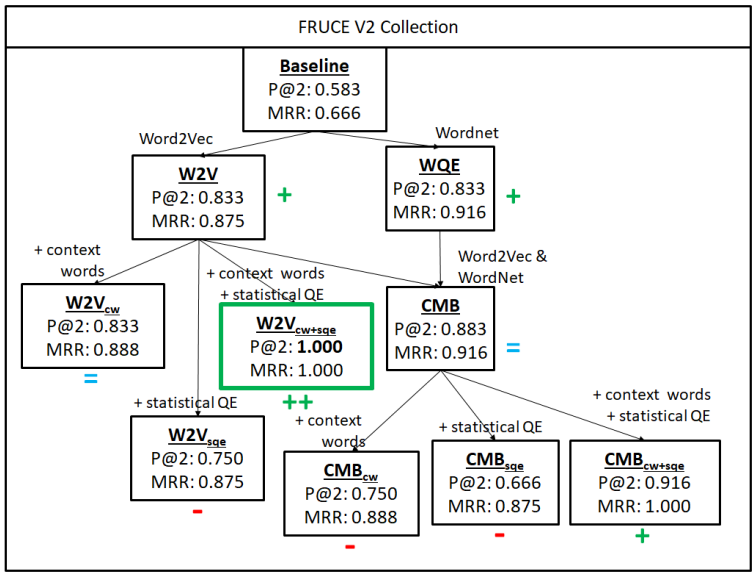

Figure 5.5: The derivation history of the methods and their evaluation results over FRUCE-v2

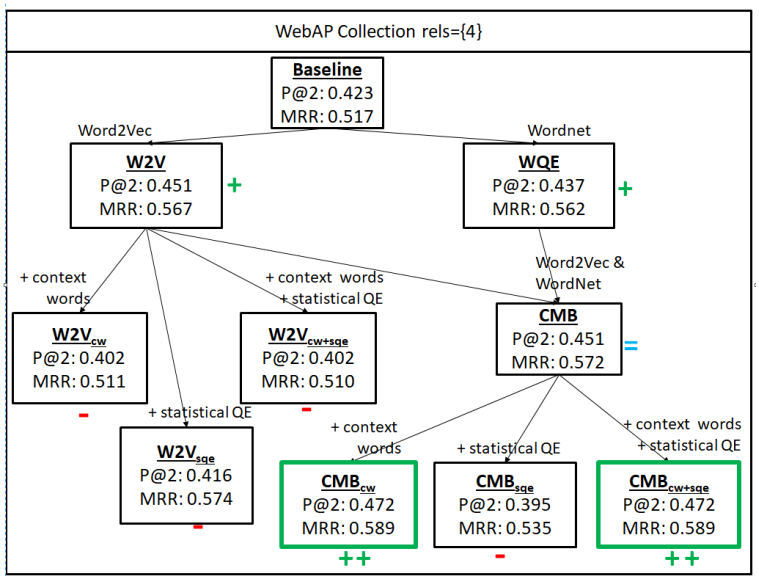

Figure 5.6: The derivation history of the methods and their evaluation results over WebAP when $rels = \{4\}$.

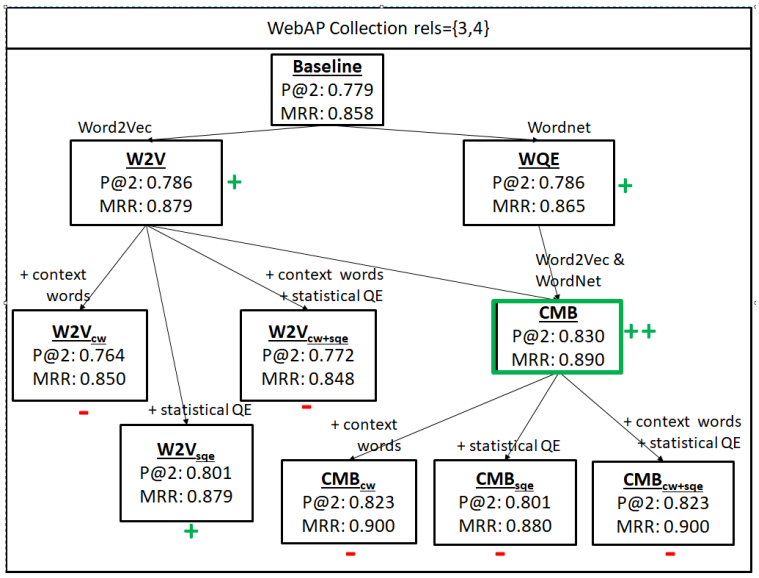

Figure 5.7: The derivation history of the methods and their evaluation results over WebAP when $rels = \{3, 4\}$.

Chapter 6

Implementation and Applications

The proposed approach is implemented in the context of research prototype LD-SDS system that is developed by TOSHIBA Research Europe and FORTH. The rest of this chapter is organized as follows, §6.1 describes the implementation, while §6.2 presents possible applications of the system. Figure 6.1 shows the main components of the architecture.

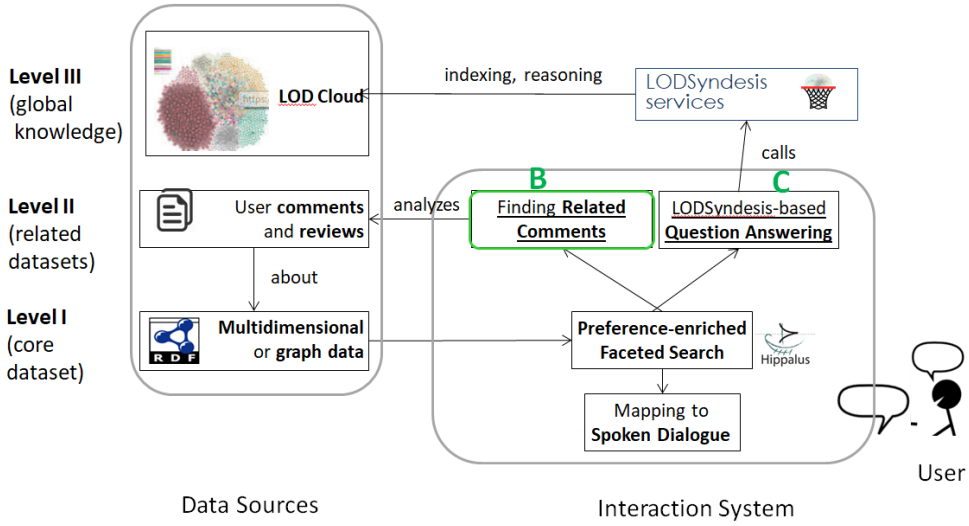

Figure 6.1: Architecture of the LD-SDS System

6.1 Implementation

Generally, the main component of LD-SDS system is the CFS component which operates over *Level I* of knowledge. That is, the core dataset (multidimensional or graph data) of the objects at hand (e.g. hotels, restaurants, cell phones etc). If

a user's question cannot be answered by the core dataset then the input questions as well as the objects on focus are passed to the component B, which is the User Review Retrieval component. Note that this is the main contribution of this work. Finally, if a question is not possible to be answered by the reviews, or if the user is not satisfied with the gathered information at that point, the question is passed to the component C, which is an open domain Question Answering System over hundreds of linked datasets, which operates over the LODSyndesis.

Practically, the component B, which is essentially the contribution of this work, is a standalone service. Particularly, it receives as input a set of product URIs, i.e. products on focus based on the CFS component and a question posed in natural language that concerns the products on focus, i.e. the input URIs. Thereafter, the service retrieves the reviews that are associated with the input products through a SPARQL query and the proposed approach is applied only over the retrieved reviews. This way the retrieval process expected to perform better in terms of precision and execution time, since the number of reviews to be scored is much lower than the total number of existing reviews and they are also product specific. We have used the *jwi version 2.2.3*[1] for the *Wordnet* component and the *dl4j version 0.9.1*[2] for the *Word2vec* component implementations. As regards the RDF graph, for retrieving the reviews on focus, *blazegraph version 2.1.4*[3] was used. The whole work was implemented in *JAVA* [4] as a *Maven* project. The input/output interaction between the CFS and User Review Retrieval components is based on *JSON* format.

6.2 Applications

A promising application domain of our work is e-Commerce. "IR for E-commerce" is an essential component of some of the largest web sites (such as eBay, Amazon, Airbnb, Alibaba, Taobao, Target, Facebook, and others) and lately we observe a trend of the research community to further study such tasks, as signified by workshops like started with the SIGIR 2017 Workshop on eCommerce (ECOM17) followed by the 2018 event [4]. Commonly all items in such sites are described by a structured description and are accompanied by user reviews, and the approach investigated in this paper is a Web 3.0 approach since it aims at advancing the way that users interact with web site (spoken language, session-aware, machine learning-derived word embeddings).

[1] `https://mvnrepository.com/artifact/edu.mit/jwi/2.2.3`

[2] `https://mvnrepository.com/artifact/org.deeplearning4j/deeplearning4j-core/0.9.1`

[3] `https://mvnrepository.com/artifact/com.blazegraph/bigdata-core/2.1.4`

[4] java version 1.8 plus is recommended for compatibility

Chapter 7

Discussion

Here we discuss related issues such as how the user session affects the review retrieval (in §7.1) and alternative approaches (in §7.2).

7.1 User Session and Review Retrieval

We wondered whether the session of the user, i.e. the user'As actions before making a review-related question, should be taken into account during the review retrieval. For the scenarios that we have seen and considered, that would not be useful, for instance during hotel finding, the user session contains hard and soft constraints related to the location, price, stars, hotel type etc. This information would not be useful in review retrieval since the reviews mainly focus on aspects that are not directly related to the key characteristics of the hotels. However we should stress, that the value that the user session brings to the review retrieval process is that it reduces the hotels whose reviews have to be considered. This is beneficial not only for reasons of efficiency but also of quality (it is easier to achieve high quality review retrieval from a smaller set than from a larger set of reviews).

We should note that we resort to reviews only if the structured dataset is not enough for answering a user question. For instance, if noisiness is a facet, then a related question or command of the user will be translated to hard/soft constraints that will be managed based on structured dataset for filtering our (or for ranking) the hotels according to their noisiness.

To synopsize, the user session indirectly affects the review retrieval: it affects the ordering of objects which in turn determines which reviews will be considered during review scoring. However the session of the user and the results from the review retrieval, could be considered together for improving and ordering of the objects, but is beyond the scope of the current work since where we focus on review retrieval.

7.2 Alternative Approaches

7.2.1 Paragraph Vectors

An alternative approach for review retrieval could be based on *doc2vec* [16], that is to come up with the representation of words that is based not on a general collection of texts, but on a set of user reviews. This however requires a very big collection of reviews for getting good results, which is not easy to obtain. This is verified by experiments that we performed, that are summarized next, which showed that such method returns worse results than the methods described in the previous section if trained only in the available collections of documents.

In more details, doc2vec is a word2vec based model for producing *paragraph embeddings* but instead of using just words to predict the next word, they use an additional feature vector, which is document-unique. When training the word vectors, say the vector of a word w_i, the document vector d_k is trained as well, and in the end of training, it holds a numeric representation of the document. While a word vector represents the concept of a word, the document vector intends to represent the concept of a document.

For the purposes of the user review retrieval we have used the reviews in FRUCE-v2 collection and we used each review id (e.g. c_k) as a label that represents the concept of the associated review. Then we trained doc2vec for producing a word vector for each id, where its spot in the vector space intends to reveal the semantic concept of the respective review. For training we set the learning rate to $\alpha = 0.025$ and minimum learning rate to $\alpha_{min} = 0.001$, while we used a number of epochs $\#epochs = 1000$. Thereafter, we calculate inferred vector for given question, using the default parameters as mentioned for training. Finally we compute the cosine similarity between the question inferred vector and each review inferred vector and sort the reviews in decreasing order based on this score.

As expected, the paragraph vectors are far inferior in comparison to the methods described in the previous sections, as regards the FRUCE-v2 collection, as shown in Table 7.1, since the reviews in this dataset are very few, in comparison to the Word2vec embeddings that have been trained on part of Google News dataset (about 100 billion words) and contains 300-dimensional vectors for 3 million words and phrases.

Method	Precision-2	R-Precision	AveP	Recall	Bpref	nDCG	MRR
d2v	0.333	0.366	0.196	0.366	0.266	0.529	0.597

Table 7.1: Evaluation of doc2vec over FRUCE-v2

7.2.2 Automatic Context Words Detection

As far as we concern, this is the first work that defines the Context Words (CW) and also propose an approach, that uses a collection of such words in order to

improve the retrieval results of user reviews. Yet, there is no method that automatically detect such terms. It is reasonable though, that since the idea of CW originate from irrelevant reviews, but with high relevance score with respect to a user question, such reviews are probably the ones that we should exploit in order to extract CW for the domain at hand. That means, that we should first be able to identify the question-review pairs with the aforementioned characteristics, i.e irrelevant but similar to the input question. It is interesting that in our context of CFS, these reviews could be provided in two ways. (1) Through Interactive Relevance Feedback Mechanism, or (2) through training collections. As regards (1), it easy for the user to decide weather a review-sentence is relevant to his/her own question, since we are not focusing on factoid questions where the answer is unknown to the user, but in opinion questions, that are facet-oriented (but for unknown facets). In such questions there is no single credible source (user review in our case) that contains the golden answer, but are rather collected from many, relevant to the input question reviewed opinions. That means that the user does not have to know if a relevant review sentence is correct or not, but if it is relevant to what he/she asked. As regards (2), it is worth researching how given a training collection of question-review pairs, marked with their level of relevance, we can simulate the Interactive Idea and thus collect the set of CW in training mode, and then use it as it is in the testing mode.

In general, many variations of this approach could be applicable. For example, it could be used both a training set for initializing a CW-list and then keep extending it as users identify irrelevant comments in the result list. Additionally to that, the approach could be user specific. In such a case the system could keep a CW-list per user. This is quite convenient, since different users are interested more about specific topics and for this reason tend to ask questions that are associated with a finite set of topics. To this end, a user-specific CW-list is expected to be more efficient for the associated user, since it could perform better to the questions of interest, i.e. the questions that the specific user asks more frequently. For example if a user considers the topic of *noise* very important for hotel booking, it is expected to ask about it whenever he/she wishes to book a hotel room. Additionally, for such a user, it is far more important to receive relevant information about *noise* than any other topic, thus a user-specific CW-list would be more pleasant.

After detecting such a review, the task is to identify which of the terms that it consists of, are the ones that falsely increased it's relevance with the given question. That probably indicates that someone should search the terms in the irrelevant review, that increase it's similarity with the input question. A potential approach for this task could be the one used in SQE approach presented in §4.3, for removing the noise from the k-nearest neighbors of the query vector. It is reasonable to follow such an approach, since the task in both cases is to identify only the most relevant terms to an input question between a set of terms. Moreover, note that in the SQE case, all chosen words (k-nearest neighbors) are considered, at least, somehow relevant, whereas in the case of CW, only CW are expected to be relevant and thus the method is expected to perform better.

However, to the best of our knowledge, a collection for our purpose does not exist. To this end we set as future work the implementation and evaluation of the process described in this section, as well as the creation of such a collection.

Chapter 8

Conclusion

In the context of Faceted Search quite often the structured data are insufficient for answering a user's query. In such cases the system could resort to related textual reviews (expressed in natural language) for identifying those that could be exploited for helping the user. This requires finding the most relevant reviews that (a) are associated with the most preferred objects, and (b) are related to a user's question. Moreover, spoken dialogue interaction poses increased requirements on quality, in order to avoid wasting user's time by reading irrelevant comments.

To this end, we proposed methods that exploit lexical databases like the hand-crafted WordNet dictionary and the statistical Word2vec approach that transforms individual words into vectors of low dimensionality. Variations of the previous methods enriched with techniques that try to devalue words that describe the context of the dataset domain and exploit Word2vec statistical query expansion are also presented. We show the real-time applicability of the proposed methods.

Our experimental results over the hand-crafted closed-domain collection FRUCE-v2, that consists of 139 hand-crafted user reviews/comments and 6 queries, and the WebAP GOV2 TREC collection comprising more than 8000 passages and 80 queries, show the effectiveness of the proposed approaches. In particular, due to the specific requirements of the CFS, we give emphasis on the quality of top-2 results. In the case of the closed-domain FRUCE-v2, where it is easy to identify appropriate contex words, the Word2vec approach and its combination with Wordnet, enhanced with the context-words and statistical query expansion methods (i.e. methods III_{cw+sqe}, IV_{cw+sqe} respectively), perform better than the plain methods and much better than the baseline one, providing excellent results in all metrics. Specifically, III_{cw+sqe} method is able to achieve a 1.0 value for Precision@2 and in MRR, and the rather high nDCG value of 0.933. The IV_{cw+sqe} method also achieves 1.0 in MRR and gives better results in Precision-R, AveP, Recall and Bpref, but worse in Precision@2 (0.916).

On the other hand in the more open-domain WebAP dataset, the scores of all metrics for all methods are lower and the context-word approach seems to lower even further the performance of the Word2vec method (i.e. methods III_{cw},

III_{cw+sqe}). In the combined WordNet and Word2Vec approach (i.e. methods IV_{cw}, IV_{cw+sqe}), context-words are able to improve the top ranked results, achieving the best results regarding the Precision@2 and MRR metrics (0.472 and 0.589) for this dataset. The statistical query expansion methods III_{sqe}, IV_{sqe} perform better, especially the III_{sqe} that achieves the best results for the WebAP dataset for the rest of the metrics (e.g. nDCG of 0.895). The experimental results are rather promising, with better performance than other relevant works, although as discussed, a direct comparison is not appropriate.

Note that we might gain performance by training the Word2vec over the used collections. However, this is beyond the scope of this work, since the size and the quality of the underlying collection might not always guarantee a high-quality Word2vec model. Regarding the CW-aware WMD the key to better performance lies in the quality of the chosen CW.

8.1 Possible Future Work

List of Possible Future Directions: In the future it is worth researching methods to:

a) Continue the quality evaluation over the real dataset scraped from *booking.com*.

b) Investigate the applicability of comparative opinion mining techniques. [45].

c) Investigate how we could exploit external sources in cases where even the user comments/reviews are insufficient for answering a user's question.

d) Diversification if it is required by the system responses over the evaluation datasets.

e) Develop methods for automatically detecting the context-words in both closed and open domain datasets.

f) Develop methods for answering complex question that require both structured and unstructured information.

Bibliography

[1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051, 2017.

[2] Daniel Cohen and W Bruce Croft. A hybrid embedding approach to noisy answer passage retrieval. In *European Conference on Information Retrieval*, pages 127–140. Springer, 2018.

[3] Gerard De Melo and Gerhard Weikum. Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 513–522. ACM, 2009.

[4] Jon Degenhardt, Pino Di Fabbrizio, Surya Kallumadi, Mohit Kumar, Yiu-Chang Lin, Andrew Trotman, and Huasha Zhao. Sigir 2018 workshop on ecommerce (ecom18). In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1407–1409. ACM, 2018.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Mamadou Diao, Sougata Mukherjea, Nitendra Rajput, and Kundan Srivastava. Faceted search and browsing of audio content on spoken web. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1029–1038. ACM, 2010.

[7] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*, 2016.

[8] Eleftherios Dimitrakis, Konstantinos Sgontzos, Panagiotis Papadakos, Yannis Marketakis, Alexandros Papangelis, Yannis Stylianou, and Yannis Tzitzikas. On finding the relevant user reviews for advancing conversational faceted search. In *Proceedings of 4th Workshop on Sentic Computing, Sentiment Analysis, Opinion Mining, and Emotion Detection (EMSASW 2018) Co-located with the 15th Extended Semantic Web Conference 2018 (ESWC 2018), Heraklion, Greece, June 4, 2018.*, pages 22–31, 2018.

[9] P. Fafalios and Y. Tzitzikas. X-ENS: Semantic Enrichment of Web Search Results at Real-Time. In *SIGIR'13*, pages 1089–1090, Dublin, Ireland, 2013.

[10] Pavlos Fafalios, Michail Salampasis, and Yannis Tzitzikas. Exploratory patent search with faceted search and configurable entity mining. In *1st International Workshop on Integrating IR technologies for Professional Search (ECIRb••13 Workshop)*, 2013.

[11] Sébastien Ferré. Sparklis: an expressive query builder for sparql endpoints with guidance in natural language. *Semantic Web*, 8(3):405–418, 2017.

[12] Mostafa Keikha, Jae Hyun Park, and W Bruce Croft. Evaluating answer passages using summarization measures. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 963–966. ACM, 2014.

[13] Mostafa Keikha, Jae Hyun Park, W Bruce Croft, and Mark Sanderson. Retrieving passages and finding answers. In *Proceedings of the 2014 Australasian Document Computing Symposium*, page 81. ACM, 2014.

[14] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[15] Saar Kuzi, Anna Shtok, and Oren Kurland. Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1929–1932. ACM, 2016.

[16] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[17] Panagiotis Lionakis and Yannis Tzitzikas. Pfsgeo: Preference-enriched faceted search for geographical data. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 125–143. Springer, 2017.

[18] Betia Lizbeth López-Ochoa, José Luis Sánchez-Cervantes, Giner Alor-Hernández, Ma Antonieta Abud-Figueroa, Beatriz A Olivares-Zepahua, and Lisbeth Rodríguez-Mazahua. An architecture based in voice command recognition for faceted search in linked open datasets. In *International Conference on Software Process Improvement*, pages 174–185. Springer, 2017.

[19] Betia Lizbeth López-Ochoa, José Luis Sánchez-Cervantes, Giner Alor-Hernández, Mario Andrés Paredes-Valverde, and José María Álvarez-Rodríguez. *FASELOD: A Faceted Search Engine in Linked Open Datasets Using Voice Recognition*, pages 245–267. Springer International Publishing, Cham, 2019.

[20] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[21] Nikos Manolis and Yannis Tzitzikas. Interactive exploration of fuzzy rdf knowledge bases. In *Extended Semantic Web Conference*, pages 1–16. Springer, 2011.

[22] Julian McAuley and Alex Yang. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, pages 625–635. International World Wide Web Conferences Steering Committee, 2016.

[23] Florian Metze, Xavier Anguera, Etienne Barnard, Marelie Davel, and Guillaume Gravier. The spoken web search task at mediaeval 2012. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8121–8125. IEEE, 2013.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

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

[27] Panagiotis Papadakos, Nikos Armenatzoglou, Stella Kopidaki, and Yannis Tzitzikas. On exploiting static and dynamically mined metadata for exploratory web searching. *Knowledge and information systems*, 30(3):493–525, 2012.

[28] Panagiotis Papadakos and Yannis Tzitzikas. Comparing the effectiveness of intentional preferences versus preferences over specific choices: a user study. *International Journal of Information and Decision Sciences*, 8(4):378–403, 2016.

[29] Dimitra Papadimitriou, Georgia Koutrika, Yannis Velegrakis, and John Mylopoulos. Finding related forum posts through content similarity over intention-based segmentation. *IEEE Transactions on Knowledge and Data Engineering*, 29(9):1860–1873, 2017.

[30] Alexandros Papangelis, Panagiotis Papadakos, Margarita Kotti, Yannis Stylianou, Yannis Tzitzikas, and Dimitris Plexousakis. Ld-sds: Towards an

expressive spoken dialogue system based on linked-data. In *Search Oriented Conversational AI, SCAI 17 Workshop (co-located with ICTIR 17)*, 2017.

[31] Dae Hoon Park, Hyun Duk Kim, ChengXiang Zhai, and Lifan Guo. Retrieval of relevant opinion sentences for new products. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 393–402. ACM, 2015.

[32] Dae Hoon Park, Mengwen Liu, ChengXiang Zhai, and Haohong Wang. Leveraging user reviews to improve accuracy for mobile app retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 533–542. ACM, 2015.

[33] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[34] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[35] Dragomir R Radev, Hongyan Jing, and Malgorzata Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4*, pages 21–30. Association for Computational Linguistics, 2000.

[36] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*, 2016.

[37] Giovanni Maria Sacco and Yannis Tzitzikas. *Dynamic taxonomies and faceted search: theory, practice, and experience*, volume 25. Springer Science & Business Media, 2009.

[38] Evgeny Sherkhonov, Bernardo Cuenca Grau, Evgeny Kharlamov, and Egor V Kostylev. Semantic faceted search with aggregation and recursion. In *International Semantic Web Conference*, pages 594–610. Springer, 2017.

[39] Yueming Sun and Yi Zhang. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'18, pages 235–244, New York, NY, USA, 2018. ACM.

[40] Johanne R Trippas, Damiano Spina, Mark Sanderson, and Lawrence Cavedon. Results presentation methods for a spoken conversational search system.

In *Proceedings of the First International Workshop on Novel Web Search Interfaces and Systems*, pages 13–15. ACM, 2015.

[41] Y. Tzitzikas and E. Dimitrakis. Preference-enriched faceted search for voting aid applications. *IEEE Transactions on Emerging Topics in Computing*, PP(99):1–1, 2016.

[42] Yannis Tzitzikas, Nicolas Bailly, Panagiotis Papadakos, Nikos Minadakis, and George Nikitakis. Using preference-enriched faceted search for species identification. *International Journal of Metadata, Semantics and Ontologies*, 11(3):165–179, 2016.

[43] Yannis Tzitzikas, Nikos Manolis, and Panagiotis Papadakos. Faceted exploration of rdf/s datasets: a survey. *Journal of Intelligent Information Systems*, pages 1–36, 2016.

[44] Yannis Tzitzikas and Panagiotis Papadakos. Interactive exploration of multidimensional and hierarchical information spaces with real-time preference elicitation. *Fundamenta Informaticae*, 20:1–42, 2012.

[45] Kasturi Dewi Varathan, Anastasia Giachanou, and Fabio Crestani. Comparative opinion mining: a review. *Journal of the Association for Information Science and Technology*, 68(4):811–829, 2017.

[46] Ivan Vulić and Marie-Francine Moens. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 363–372. ACM, 2015.

[47] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep learning for answer sentence selection. *CoRR*, abs/1412.1632, 2014.