Metadata-Aware Measures for Answer Summarization in Community Question Answering

Mattia Tomasoni
Abstract

Metadata-Aware Measures for Answer Summarization in Community Question Answering

Mattia Tomasoni

My thesis report presents a framework for automatically processing information coming from community Question Answering (cQA) portals. The purpose is that of automatically generating a summary in response to a question posed by a human user in natural language. The goal is to ensure that such answer be as trustful, complete, relevant and succinct as possible. In order to do so, the author exploits the metadata intrinsically present in User Generated Content (UGC) to bias automatic multi-document summarization techniques toward higher quality information. The originality of this work lies in the fact that it adopts a representation of concepts alternative to n-grams, which is the standard choice for text summarization tasks; furthermore it proposes two concept-scoring functions based on the notion of semantic overlap. Experimental results on data drawn from Yahoo! Answers demonstrate the effectiveness of the presented method in terms of ROUGE scores. This shows that the information contained in the best answers voted by users of cQA portals can be successfully complemented by the proposed method.
I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person except where due acknowledgment has been made in the text.

Uppsala University Library is allowed to provide academic copies of the present Masters Thesis without need for further permission from the author.

Mattia Tomasoni, February 14, 2011
Acknowledgments

The research behind this thesis work was carried out in Beijing, China: my thanks go to Tsinghua University in the persons of Dr. Minlie Huang and Prof. Xiaoyan Zhu. They welcomed me in their laboratory, guided me in the choice of a topic and offered their suggestions; more precious than anything, I was always allowed to reason independently and to decide the direction of my research. Dr. Huang’s experience was invaluable and his feedback allowed me to turn the concoction of intuitions and experimental results that were in my head into a publishable piece of text: our joint effort culminated in a publication at the conference ACL 2010. My thanks to them stretch outside of lab hours, for the multiple dinners, be it hotpot or homemade dumplings. I feel we mutually learned about each other’s cultures and approaches to research and work.

I thank my lab-mates and my closest friends in Beijing: Aurora, Sasan, Robert, Ben, Ryan, Tang Yang, Evelyn, Gloria, Jiaxu Li, Tim. The food we shared: baozi and jianbing, doujia and erguotou; the places: Andingmen, Jishuitan, Xizhimen, Dongzhimen, WudaoKou, Sanlitun. Special thanks to my friend Zoe who received me, offered me shelter, fed me and found me a home. I also thank my impavid traveling companions Thomas and Mary, who hitchhiked with me through Southeast Asia and are still on the road while I am back to my desk; the first in Thailand, the latter in Mongolia.

I thank my parents and sister who always give anticipating any of my requests; they have shown to trust me more than I would trust myself. Paola, Cristina and Giuseppe: I thank you and love you.

I would also like to thank Emory University in the person of Dr. Eugene Agichtein for sharing their precious dataset and for their high quality research on User Generated Content. I would like to thank Uppsala University, the Student Nations and the Swedish people for welcoming me in their country and offering me education and health care. I thank my supervisor Prof. Roland Bol and to Prof. Ivan Christoff for their professional and personal support. For their thesis-writing advices, I would like to thank Prof. Joe Wolfe from the University of New South Wales, Prof. William D. Shoaff from the Florida Institute of Technology and Prof. John W. Chinneck from Carleton University.

Finally, I would like to acknowledge the Chinese Natural Science Foundation and the International Development Research Center in Ottawa for their Economical support.

Mattia Tomasoni, February 14, 2011
In memory of the time spent together, I dedicate my work to my friend Linus Sundbom, who has been with me in thought and written word ever since we met. To many a hike, many a campfire, many a hug; peace, brotherhood and tolerance.
## Contents

Acknowledgments iii

1 Introduction 1

2 UGC, cQA, BE, ILP and Other (Obscure) Acronyms 3
   2.1 Introductory Definitions ..................................... 3
   2.2 Question Answering ........................................ 4
   2.3 Community Question and Answering .......................... 5
   2.4 Regression .................................................. 6
   2.5 Concept representation ..................................... 6
   2.6 Integer Linear Programming .................................. 7

3 Trusting, Summarizing and Representing Information 9
   3.1 Summarization in Community Question Answering ............... 9
   3.2 Quantifying Coverage in Information Retrieval .................. 10
   3.3 Information Trustfulness of User Generated Content .......... 10
   3.4 Query-dependent Summarization Techniques ..................... 11
   3.5 Concept Representation ...................................... 11

4 Our Goal: Answer Summarization in cQA 13

5 Metadata-Aware Measures 15
   5.1 The Summarization Framework .................................. 15
   5.2 Semantic Overlap ............................................. 15
   5.3 Quality ....................................................... 16
   5.4 Coverage ..................................................... 18
   5.5 Relevance ..................................................... 19
   5.6 Novelty .......................................................... 20
   5.7 The concept scoring functions .................................. 20
   5.8 Fusion as a Summarization Problem ............................ 21
6 Experiments
   6.1 Datasets and Filters ........................................ 23
   6.2 Quality Assessing ........................................... 24
   6.3 Evaluating Answer Summaries ............................... 25

7 Conclusions and Future Work ................................. 27

Bibliography ............................................................ 29

A Example Run ...................................................... 33

B The Yahoo! Answers API ........................................... 35

C Prototype Installation Manual ................................. 37
Chapter 1

Introduction

“What about Bafut?” he asked.
“Is that a good place? What are the people like?”
“There is only one person you have to worry about in Bafut [...]”

[GERALD DURRELL, The Bafut Beagles]

Community Question Answering (cQA) portals are an example of Social Media where the information need of a user is expressed in the form of a question posed in natural language; other users are allowed to post their answers in response and after a period of time a best answer is picked among the ones available according to mechanisms specific to the cQA portal. cQA websites are becoming an increasingly popular complement to search engines: overnight, a user seeking a particular information can expect a human-crafted, natural language answer tailored to her or his specific needs without having to surf the Internet and its vast amount of information. We have to be aware, though, that User Generated Content (UGC) is often redundant, noisy and untrustworthy (see [1, 2, 3]) and can sometimes contain Spam or even malicious and intentionally misleading information. Interestingly and to our advantage, though, a great amount of information that can be used to assess the trustfulness of the available sources is embedded in the metadata generated as a byproduct of users’ action and interaction on Social Media. By exploiting such metadata, we can extract and make fruitful use of much valuable information which is known to be contained in answers other than the chosen best one (see [4]). Our work shows how such information can be successfully distilled from cQA content.

To this end, we casted the problem to an instance of the query-biased multi-document summarization task, where the question expressed by the user was seen as a query and the available answers generated by other users as documents to be summarized. We agreed on four characteristic and ideal answer should present: it should be as trustful, complete, relevant and as succinct as possible. We then mapped each characteristic that an ideal answer should present to a measurable property that we wished the final summary would maximize:

- Quality to assess trustfulness in the source,
- Coverage to ensure completeness of the information presented,
CHAPTER 1. INTRODUCTION

• Relevance to keep focused on the user’s information need and
• Novelty to avoid redundancy.

Quality of the information in the user-generated answers was assessed via Machine Learning (ML) techniques: a vector space consisting of linguistic and statistical features about the answers and their authors was build and populated with real-world data instances and a classifier was trained under best answer supervision. In order to estimate Coverage, a corpus of answers to questions similar to the one to be answered was retrieved through the Yahoo! Answers API \(^1\); this corpus was chosen under the assumption that it could approximate all the knowledge available about the question to be answered: the Coverage was then calculated as the portion of information in the total knowledge that was covered by the user-generated answer. The same notion of information overlap is at the base of the Relevance measure, that was computed as overlap between an answer and its question; in a similar fashion, Novelty was calculated as inverse overlap with all other answers to the same question.

In order to generate a summary, a score was assigned to each concept in the answers to be merged according to the above properties; A score-maximizing summary under a maximum coverage model was then computed by solving an associated Integer Linear Programming problem (see \([5, 6]\)).

We chose to express concepts in the form of Basic Elements (BE), a semantic unit developed at ISI\(^2\); We modeled semantic overlap between two concepts that share a same meaning as intersection in their equivalence classes (formal definitions will be given in Chapter 5).

We would like to point out that the objective of our work was to present what we believe is a valuable conceptual framework; if further time and funding were available, more advance machine learning and summarization techniques could be investigated; this would most likely improve the performances.

The remaining of this thesis report is organized as follows. In the next chapter an overview of the necessary background information is given; in Chapter 3 we present the related literature, presenting the state of the art in Information Trust, Automatic Summarization and textual information representation. Chapter 4 contains our question statement, the objective of our research. Chapter 5, the core of this thesis report, presents the theoretical framework for answer summarization that we developed and the prototype that implements it; Chapter 6 contains the dataset on which the prototype was tested, the nature of the experiments and the results obtained. Finally, Chapter 7 presents our conclusions and ideas for future developments of our work; it is followed by four appendixes: the experiments manual, the prototype documentation, some relevant extracts of source code and a number of meaningful example runs.

Throughout the rest of this report I will use the first singular person whenever referring to work that I carried out on my own and the plural form for those parts that where devised in cooperation or published together with my external supervisor.

\(^1\)http://developer.yahoo.com/answers
\(^2\)Information Sciences Institute, University of Southern California, http://www.isi.edu
Chapter 2

UGC, cQA, BE, ILP and Other (Obscure) Acronyms

“Masa like dis kind of beef?” asked the hunter, watching my face anxiously.
“Yes I like um too much” I said, and he grinned.

[GERALD DURRELL, The Bafut Beagles]

In this chapter we give the background information regarding Question Answering and Automatic Summarization that is needed to understand the work that is presented in the following chapters. A basic knowledge in the areas of Computer Science and Information Technology is assumed. When appropriate, links to external resources for further reading are provided to complete the material presented in the chapter.

2.1 Introductory Definitions

In order introduce the reader to the field, the following section gives some brief definitions of general concepts of importance.

**Definition 1 (Machine Learning)** Machine Learning is a branch of computer science that studies the ability of a machine to improve its performance based on previous results. [7]

**Definition 2 (Computational Linguistics)** Computational linguistics is a field concerned with the processing of natural language by computers. [...] It is closely related to Natural Language Processing and Language Engineering. [8]

**Definition 3 (Natural Language)** A natural language is the systematic result of the innate communicative behavior of the human mind and its learning is biologically driven. It is manipulated and understood by humans as opposed to formal languages used to communicate orders to machines or express logical/mathematical statements.
Definition 4 (User-Generated Content (UGC)) User-Generated Content is any portion of publicly available material generated with creative intent by the end-users of a system without professional or commercial intent by their interaction of an web-application. It is also referred to as “Conversational Media”, “Consumer Generated Media”, “Collaborative Authored Content” or “Social Media Content”.

Definition 5 (Metadata) Metadata, literally “beyond the data”, is commonly defined as “data about data”: it is usually structured according to a specific scheme and can provide the time and date of creation, the author, the purpose and other similar information about the data it is describing.

2.2 Question Answering

Definition 6 (Question Answering (QA)) Question Answering is the task of automatically formulating an answer to satisfy the need of a user.

Although our goal is the summarization of User Generated Content to effectively answer human questions, which is a much more modest goal than the creation of a Question Answering System, the two are closely related. Question Answering is a particularly challenging subfield of Information Retrieval in that both the question posed by the user and the answer provided by the system are in natural language.

A Question and Answering system is firstly a natural language user interface used to retrieve information from a corpus of data; but is more than a natural language Search Engine: it does not merely determine the location of the desired information, but it distills it, processes it and presents it in the most human-compatible way: natural language text. The amount of information on the Internet is increasing exponentially and much research is being devoted in this direction. A fusion between question and answering systems and natural language search engines is regarded as one of the likely directions in which present day keyword-based engines might evolve in the future.

The first attempts to build such systems date back to the ’60: they were focused on answering domain specific questions and relied on an underlying expert system. Many comprehensive theories in Computational Linguistics and reasoning have been developed in the last decades and while Closed-Domain Question Answering is still a studied problem, the focus has nowadays shifted to Open-Domain Question Answering, where questions are not restricted to a specific area of human knowledge. The present state of the art in Open-Domain Question Answering is far from yielding satisfactory results. Many complex problems remain unsolved: given the intrinsic ambiguous of natural languages, the context in which a question is asked has to be taken in consideration in order to answer it correctly. Even after the information that is believed to be adequate to answer the question has been found, much work remains to be done: pieces of text coming from different sources must be merged through the use of answer fusion techniques. Furthermore, reasoning, common-sense and ability to perform inference might be required.

The problem that is being tackled by our work is a QA in the convenient case where the information in the corpus to be searched comes from cQA Websites. Under this restriction, many of the above problems are much easier to solve than in the general setting. For this reason information contained in cQA regarded as particularly valuable and the scientific community is starting to devote energies to devise methods that can exploit. Our claim is that it could be used to integrate the general QA systems.
2.3 Community Question and Answering

Definition 7 (Community Question Answering (cQA)) A Community Question and Answering (cQA) portal is a website where users can post and answer questions; questions and relative answers are subdivided into categories and are made available to other Internet users.

It is crucial to notice that the content of Community Question Answering websites is intrinsically different from the content of an on-line newspaper or a personal web-page; a term has been coined to capture its nature: User Generated Content.

Community Question and Answering portals are an instance of Social Media. Examples of Social Media include, blogs, micro-blogs, Social Networks, Forums, Wikis, Social News Services, Social Bookmarking (folksonomies), and Sharing websites for pictures, videos or music. User Generated Content is shaping the nature of the Internet so dramatically that the term “Web 2.0” was introduced to mark the evolution from the early-years static collection of interlinked home-pages to the dynamic and richer Internet that we know nowadays. User Generated Content though, is often redundant, noisy and untrustworthy [1, 2, 3]; this rises a trustfulness issue. Luckily, though, a post on a Community Question Answering website is much more than an anonymous string of characters appearing on the Internet: it has associated with it information about the user who posted it, the time and category under which it was filed and the opinions of other users about it; furthermore users make up a community, the structure of which can be studied and analyzed to better address trustfulness concerns.

The cQA service we selected for our experiments is Yahoo! Answers. Anybody that has a Yahoo! account can post a question at http://answers.yahoo.com/question/ask and answer...
questions that have been posted. Users can also express their opinion regarding the correctness of answers posted by others with their vote; after a fixed time (usually a week) the answer that was most voted by the community is chosen as the best answer. A more immediate and practical objective of our work than the one of attempting to build a module to a QA system, is that of complementing such best answer with a machine generated summary of valuable information coming from other answers.

2.4 Regression

As mentioned, our goal is to mine information from the metadata associated with each answer in order to assess trustfulness. A number of meaningful statistical properties need to be extracted and analyzed from the metadata associated with answers. The statistical instrument we picked to obtain as estimate of the degree of trust to be assigned is called Regression; we adopted its simplest form, where trust is modeled as a linear function of the properties in input as exemplified in the figure to the right

\[ y = a + bx \]

**Definition 8 (Linear Regression)** Linear Regression is a statistical technique that defines a line that best fits a set of data points and predicts the value of an outcome variable \( y \) from the values of one or more continuous variables \( X \) focusing on the conditional probability distribution of \( y \) given \( X \). [9]

2.5 Concept representation

A sentence can be represented in many ways, depending on the task: for instance as a bit vector, as a bag of words, as an ordered list of words or as a tree of sub-sentences. For text summarization purposes, the most common approach is to use so called n-grams to represent concepts; we intuitively refer to a concept as the smallest unit of meaning in a portion of written text, the semantic quantum of a sentence.

**Definition 9 (N-gram)** An N-gram is a tuple of \( n \) adjacent (non-stop) words from a sentence. Typical values of \( N \) are the natural numbers 2 and 3.

Note that bag-of-words is a particular instance of the N-gram representation where \( N \) is equals to 1. The N-gram approach is the most widely used and currently the most successful. It is to be noted, though, that since the text is being treated from a purely syntactic point of view, the meaning that the words express is being overlooked. As an example let’s consider the two bigrams “ferocious-bear” and “fierce-Ursidae”: although totally unrelated syntactically, those two constructs are known to have the same meaning to any English-speaking person with a knowledge of zoology. The N-gram approach suffers greatly from what is known as Semantic Gap; formally:

\[ \text{Semantic Gap} = \left| \text{semantic meaning of } \text{ferocious-bear} - \text{semantic meaning of } \text{fierce-Ursidae} \right| \]

\[ \text{Semantic Gap} = 0 \]

Footnote: [1] Created using macanova by Wikipedia’s user Michael Hardy and released into the public domain
Definition 10 (Semantic Gap) The Semantic Gap is the loss of informative content between a powerful language and a formal language.

For our purposes we explored an alternative Concept representation that could be defined bag-of-BEs. A BE (Basic Element) is “a head|modifier|relation triple representation of a document developed at ISI” [10]. BEs are a strong theoretical instrument to tackle the ambiguity inherent in natural language that find successful practical applications in real-world query-based summarization systems. Different from n-grams, they are variant in length and depend on parsing techniques, named entity detection, part-of-speech tagging and resolution of syntactic forms such as hyponyms, pronouns, pertainyms, abbreviation and synonyms. To each BE is associated a class of semantically equivalent BEs as result of what is called a transformation of the original BE; the mentioned class uniquely defines the concept. What seemed to us most remarkable is that this makes the concept context-dependent. A sentence is defined as a set of concepts and an answer is defined as the union between the sets that represent its sentences.

How the use of BEs for concept representation help filling the Semantic Gap will be clarified in Chapter 5 where a formal definition of Equivalence Class is given and a related semantic operator is defined.

2.6 Integer Linear Programming

Our final goal is text summarization. Many techniques exist that can reduce the volume of a human-written text retaining only the most important information and discarding the rest. The one that best suited our task, as will be described in Chapter 3, is based on the optimization technique known as Integer Linear Programming. We will now give a definition.

Definition 11 (Integer Linear Programming) An Integer Linear Program is a problem expressible in the following form. Given an \( nm \) real matrix \( A \), \( m \)-vector \( b \) and \( n \)-vector \( c \), determine \( \min_x \{ c \cdot x \mid Ax \geq b \land x \geq 0 \} \) where \( x \) ranges over all \( n \)-vectors and the inequalities are interpreted component-wise, i.e., \( x \geq 0 \) means that the entries of \( x \) are nonnegative. Additionally, all of the variables must take on integer values. [11]

Given a set of variables that can assume integer values, an Integer Linear Program is the problem of assigning values to such variables so as to maximize a certain objective function within the feasible area defined by a series of constraints. Solving such problem is known to be NP-hard. In the figure, an objective function in two dimensions is being optimized under three linear constraints; lattice points indicate feasible integer values for the variable (labeled \( x_1 \) and \( x_2 \)). Intuitively, it can be applied to text summarization where each unit of text (word, concept or sentence) is assigned a binary variable that indicates inclusion in the summary and a series of constraints are set up together with some meaningful function whose output is to be optimized. Details are given in the next chapter.
Chapter 3

Trusting, Summarizing and Representing Information

“If we go meet bad beef we go catch um, no kill um” I said firmly.
“Eh! Masa go catch bad beef?”
“Na so, my friend. If you fear, you no go come, you hear?”

[GERALD DURRELL, The Bafut Beagles]

In this chapter we present the literature that, directly or indirectly, relates to the work presented in this thesis. Recent research and state of the art in Information Trust in User Generated Content, Automatic Summarization and representation of Textual Information are presented. Each piece of literature that is briefly summarized in this chapter has been chosen because part of the theoretical foundation upon which I build my own work.

3.1 Summarization in Community Question Answering

The starting point of our research was the study by Liu, Li et. al. named “Understanding and Summarizing Answers in Community-based Question Answering Services”[4]. They pointed out that for many types of questions it is not possible to identify which answer is the correct one; example are non-factoid questions asking for opinions, suggestions, interpretations and the like. We believe that answers to questions of this kind are precisely what makes User Generated Content so valuable in that they provide access to information that cannot be simply looked up in an encyclopedia. For questions of the mentioned kind, portions of relevant and correct information will be spread among multiple answers other than the chosen best one; Liu and Li’s idea is to use summarization techniques to collect it and present it at once. Although this intuition has a strong potential and interesting possibilities of practical application in the future, we argue that the techniques presented in their paper fail to take in consideration the peculiarities of the input domain; in our work we exploited the properties associated with content on Social Media (i.e. the available metadata) to devise custom measures that would address challenges that are specific to Question Answering.
such as information trust, completeness of the answer and originality; additionally we explored novel means of representing the information and summarization technique that could support such measures to devise and ah hoc solution to the specific problem.

### 3.2 Quantifying Coverage in Information Retrieval

In their paper “Essential Pages” [12], Ashwin, Cherian et. al., defined query coverage as the portion of relevant information provided by a link out of the hypothetical total knowledge available on the Web. The Objective of their work was to “build a search engine that returns a set of essential pages that maximizes the information covered” [12]. The Coverage measure was based on the familiar Term Frequency and on a score called Term-relevance based on the popularity of a term in the total knowledge. We adapted the idea of Coverage to our scenario and our representation in order to evaluate the completeness of the answers to be summarized; please refer to Section 5.4 for details.

### 3.3 Information Trustfulness of User Generated Content

Information trustfulness laid at the core of our research. The work “Finding High-Quality Content in Social Media” [13] by Agichtein, Castillo et. al. underlines that the “quality of user-generated content varies drastically from excellent to abuse and spam [...] and that] as the availability of such content increases, the task of identifying high-quality content in sites based on user contributions - social media sites - becomes increasingly important.” Their research shows how this goal can be achieved with accuracy close to that of humans by exploiting the metadata that accompanies the content of Community Question and Answering websites: “in addition to the content itself, there is a wide array of non-content information available, such as links between items and explicit quality ratings from members of the community”. Among the many results, they presented an ordered list of quality features for answers in Yahoo! Answers; we select the most representative among those features that were available in our dataset and design a feature space in which answers can be represented and similar quality estimates can be produced. The design of our Quality feature space is presented in Section 5.3.

Other related studies include “Wisdom: a Web Information Credibility Analysis System”[14] by Akamine, Susumu et. al., a general approach to the broad problem of evaluating information credibility on the Internet by making use of semantic-aware Natural Language Preprocessing techniques. The following papers have analogous goals, but a focus on User Generated Content and Wikipedia in particular: “Assessing Information Quality of A Community-Based Encyclopedia”[15] by Stvilia, Twidale et. al., “Investigation Into Trust for Collaborative Information Repositories: A Wikipedia Case Study”[16] by MeGuinness, Zeng et. al., “Measuring Article Quality in Wikipedia: Models and Evaluation”[17] by Hu, Meiqun et. al. and finally “Computing Trust from Revision History”[18] by Zeng, Honglei et. al. “A Framework to Predict the Quality of Answers with Non-textual Features”[1] by Jeon, Croft et. al., is a study in the specific domain of Community Question Answering which presents a framework that uses Maximum Entropy for answer quality estimation with non-textual features similar to the ones proposed by Agichtein, Castillo et. al. [13] in the article introduced above. Another method that shares this same objective is the one more recently published in “Quality-aware Collaborative Question Answering: Methods and Evaluation” by Suryanto, Lim et. al., which is based on the expertise of answerers. Finally, Wang, Xin-Jing et. al., in their article “Ranking Community Answers by Modeling Question-answer Relationships via Analogical Reasoning”[2], introduce the interesting idea of ranking answers taking their relation to questions in consideration.
3.4 Query-dependent Summarization Techniques

As mentioned, summarization techniques are central to our work; in Section 2.6 we briefly mentioned how Integer Linear Programming could provide means of solving text summarization problems at the concept level. The idea was presented in the article “A Scalable global Model for Summarization”[5] by Gillick and Favre; it presented an extractive summarization method which addresses redundancy globally at the concept level and makes use of an Integer Linear Program for exact inference under a maximum coverage model; “an ILP formulation is appealing because it gives exact solutions and lends itself well to extensions through additional constraints”[5]. We adapted the automatic model they proposed to our needs by incorporating measures such as trust and completeness. Please refer to Section 2.6 for details on Integer Linear Programming and to Section 5.8 for details on our implementation of Gillick and Favre’s method.


3.5 Concept Representation

It was said above that the summarization method devised by Gillick and Favre works at the concept level; in their work concepts were expressed as N-grams, but we opted for a more powerful representation called Basic Elements. Both N-grams and Basic Elements have been described in Section 2.5 to which the reader can refer for details. Basic Elements were presented in the short paper “Summarizing Answers for Complicated Questions”[10] by Zhou, Lin et. al., where they illustrate the functioning of their own query-based summarizer based on the Basic Elements paradigm and they give references to the online framework that implements it. The purpose of the framework is to evaluate machine summaries, but I modified it so that it would perform Equivalence Classes extraction on our dataset.
Chapter 4

Our Goal: Answer Summarization in cQA

“You lie, my friend. You no be ole man. You done get power too much [...]”

He chuckled, and then sighed.

“No, my friend, you no speak true. My time done pass.”

[GERALD DURRELL, The Bafut Beagles]

In this chapter I formally state the objective of my thesis project: proposing a series of metadata-aware measures to score concepts in Community Question Answering according to their importance and testing their effectiveness in guiding the task of summarizing the answer form which they come from.

In order to do so I will have to investigate the following: how can User Generated Content from Community Question Answering websites be used to answer questions? More specifically:

Is it possible to devise a procedure to automatically process Community Question Answering information with the purpose of generating a satisfactory answer in response to an arbitrary user question by making use of the metadata intrinsically available in User Generated Content?

And if this is the case, is it possible to ensure correctness? To what degree? Furthermore, can such answer be guaranteed to be complete in the information it provides, maximally relevant to the question and as succinct as possible? Could a heuristic estimate of the mentioned properties be given? What would the precise mathematical formulation be? Can a compromise between those desirable but often conflicting properties be established? Should an incomplete but relevant answer be preferred to a more complete but less relevant one? But what if it also appears to be less trustworthy? Would the more complete one be preferable in that case? Would that be true even if it turns out that the information it carries is available in many other answers and potentially
redundant? It also needs to be determined to what entities these properties should apply: to a whole answer? To a paragraph, rather than a sentence or a sub-branch of its parsing tree? Maybe to a each single concept or word? Once this is all settled: how can the actual summary be generated? Moreover a number of practical concerns arise: will the resulting algorithms have reasonable time and space complexities so that it could be run in practice on large sets of real-world data? Where can a suitable dataset be found? User Generated Content brings in a number of privacy issues: how can these be addressed? What pre-processing would data need? In case supervised Machine Learning techniques were used, what sources of supervision are available? Finally: can the validity of the proposed framework be demonstrated in a series of repeatable experiments?

To the best of my abilities, I will try to give satisfactory answers to the questions above in the following chapters.

CHAPTER 4. OUR GOAL: ANSWER SUMMARIZATION IN CQA

Yahoo! Answers

Resolved Question

How to protect yourself from a bear?

Fenks

I'm serious. I'm going to the Bieszczady mountains where there are apparently 200 bears and not many people. I've just read that a bear in that area killed 3 years old European biion. I know that bears usually don't attack people, but if I come across such hungry beast, what is the best way to save your life? They run better than me and can climb the trees. I don't have a gun permission. Anyone experienced?

summarized answer

In addition if the bear actually approaches you or charges you.. still stand your ground. Many times they will not actually come in contact with you, they will charge, almost touch you than run away

The actions you should take are different based on the type of bear. for example adult Grizzlies can t climb trees, but Black bears can even when adults. They can not climb in general as thier claws are longer and not semi-retractable like a Black bears claws.

I truly disagree with the whole play dead approach because both Grizzlies and Black bears are opportunistic animals and will feed on carrion as well as kill and eat animals. Although Black bears are much more scavenger like and tend not to kill to eat as much as they just look around for scraps. Grizzlies on the other hand are very accomplished hunters and will take down large prey animals when they want.

I have lived in the wilderness of Northern Canada for many years and I can honestly say that Black bears are not at all likely to attack you in most cases they run away as soon as they see or smell a human, the only places where Black bears are agressive is in parks with visitors that feed them, everywhere else the bears know that usually humans shoot them and so fear us.
Chapter 5

Metadata-Aware Measures

The first sip of the liquid nearly burnt my throat out: it was quite the most filthy raw spirit I have ever tasted. [...] He coughed vigorously and turned to me, wiping his streaming eyes. “Very strong” he pointed out.

[GERALD DURRELL, The Bafut Beagles]

This chapter, which constitutes the core of my thesis report, presents the theoretical framework for answer summarization that we developed: the metadata-aware measure, the scoring functions and the summarization method.

5.1 The Summarization Framework

As stated in previous chapters, the objective of our work is to devise a procedure to automatically process Community Question Answering information with the purpose of generating a satisfactory answer in response to an arbitrary user question \( q \). To do so, we make use of metadata intrinsically available in User Generated Content. The following are given:

- \( q \): question (to be answered)
- \( TA^a \): set of all answers to \( q \) (to be summarized)
- \( u \): profile of the user who authored answer \( a \), \( \forall a \in TA^a \)
- \( TA^u \): set of all answers ever given by the user associated with \( u \)
- \( \vartheta, \varsigma, \varpi \) and \( \varrho \): various metadata as explained in Section 5.3
- \( TA^q \): “Total Knowledge” set (“everything” that can possibly be known about \( q \))

5.2 Semantic Overlap

This section gives a formal definition of our model of concept representation based on Basic Elements (BEs) (see 2.5) and semantic overlap.
From a set-theoretical point of view, each concept $c$ was uniquely associated with a set of related concepts $E^c = \{c_1, c_2 \ldots c_m\}$ such that:

$$\forall i, j \quad (c_i \approx^L c) \land (c_i \neq c)$$

(5.1)

In our model, the “$\approx$” relation indicated syntactic equivalence (exact pattern matching), while the “$\approx^L$” relation represented semantic equivalence under the convention of some language $L$ (two concepts having the same meaning). $E^c$ was defined as the set of semantically equivalent concepts to $c$, called its equivalence class; each concept $c_i$ in $E^c$ carried the same meaning ($\approx^L$) of concept $c$ without being syntactically identical ($\equiv$); furthermore, (as implied by the definition of set) no two concepts $i$ and $j$ in the same equivalence class were identical. Given two concepts $c$ and $k$:

$$c \Rightarrow k \rightarrow E^c \cap E^k \neq \emptyset$$

**Definition 12 (Semantic Overlap ($\Leftrightarrow$))** We define semantic overlap as occurring between two concepts $c$ and $k$ if the corresponding equivalence classes $E^c$ and $E^k$ had at least one element in common.

Given the above definition of equivalence class and the transitivity of the “$\equiv$” relation, we have that if the equivalence classes of two concepts are not disjoint, then they must bear the same meaning under the convention of some language $L$; in that case we said that $c$ semantically overlapped $k$ (which is trivially true when they are syntactically identical, $c \equiv k$). It is worth noting that relation “$\Leftrightarrow$” is symmetric, transitive and reflexive; as a consequence all concepts with the same meaning are part of a same equivalence class. BE and equivalence class extraction were performed by modifying the behavior of the BEwT-E-0.3 framework $^1$. The framework itself is responsible for the operative definition of the “$\approx^L$” relation and the creation of the equivalence classes.

### 5.3 Quality

Quality assessing of information available on Social Media had been studied before mainly as a binary classification problem with the objective of detecting low quality content. We, on the other hand, treated it as a ranking problem and made use of quality estimates with the novel intent of successfully combining information from sources with different levels of trustfulness. This is crucial when manipulating UGC, which is known to be subject to particularly great variance in credibility $[1, 2, 3]$.

$^1$The authors can be contacted regarding the possibility of sharing the code of the modified version. Original version available from [http://www.isi.edu/publications/licensed-sw/BE/index.html](http://www.isi.edu/publications/licensed-sw/BE/index.html).
An answer $a$ was given along with information about the user $u$ that authored it, the set $TA^q$ (Total Answers) of all answers to the same question $q$ and the set $TA^u$ of all answers by the same user. Making use of results available in the literature [13], we designed a Quality feature space to capture the following syntactic, behavioral and statistical properties:

- $\vartheta$, length of answer $a$
- $\varsigma$, number of non-stopwords in $a$ with a corpus frequency larger than 5
- $\varpi$, points awarded to user $u$ according to the Yahoo! Answers’ points system
- $\varrho$, ratio of best answers posted by user $u$

The features mentioned above determined a space $\Psi$; An answer $a$, in such feature space, assumed the vectorial form:

$$\Psi^a = (\vartheta, \varsigma, \varpi, \varrho)$$

Following the intuition that chosen best answers ($a^*$) carry high quality information, we used supervised ML techniques to predict the probability of $a$ to have been selected as a best answer $a^*$. We trained a Linear Regression classifier to learn the weight vector $W = (w_1, w_2, w_3, w_4)$ that would combine the above feature. Supervision was given in the form of a training set $Tr^Q$ of labeled pairs defined as:

$Tr^Q = \{(\Psi^a, isbest^a) \ldots\}$

$isbest^a$ was a boolean label indicating whether $a$ was an $a^*$ answer; the training set size was determined experimentally and will be discussed in Section 6.2. Although the value of $isbest^a$ was known for all answers, the output of the classifier offered us a real-valued prediction that could be interpreted as a quality score $Q(\Psi^a)$:

$$Q(\Psi^a) \approx P( isbest^a = 1 \mid u, TA^u, TA^q )$$
$$\approx P( isbest^a = 1 \mid \Psi^a )$$
$$= W^T \cdot \Psi^a$$

The Quality measure for an answer $a$ was approximated by the probability of such answer to be a best answer ($isbest^a = 1$) with respect to its author $u$ and the sets $TA^u$ and $TA^q$. It was calculated as dot product between the learned weight vector $W$ and the feature vector for answer $\Psi^a$.

Our decision to proceed in an unsupervised direction came from the consideration that any use of external human annotation would have made it impracticable to build an actual system on larger scale. An alternative, completely unsupervised approach to quality detection that has not undergone experimental analysis is discussed in Chapter 7.

**Example**

Consider Question $Q00$: “How do I protect myself from a bear?”

and Answer $A00$: “Protect yourself by climbing up the highest tree: that will do.”

Suppose that auxiliary information about the author is available together with all the answers he/she ever posted and all answers to the same question available on the cQA website; it would then be possible to compute the Quality properties for the answer. Suppose their values are:

$\vartheta = 11$, $\varsigma = 3$, $\varpi = 5212$, $\varrho = 0.58$.

Answer $A00$ would be injected in the quality feature space $\Psi$ in the form of the vector$^2$:

$$\Psi^{A00} = (11, 3, 5212, 0.58)$$

$^2$Unlike shown in this example, the values were in practice normalized before populating the Quality Feature Space
Also suppose that $A_{00}$ was chosen by the users of the community as best answer for question $Q_{00}$. As a result $isbest^{A_{00}} = 1$ and training example $\langle (11, 3, 5212, 0.58), 1 \rangle$ would be added to training set $Tr^{Q_{00}}$ and used to train the classifier.

5.4 Coverage

In the scenario we proposed, the user’s information need is addressed in the form of a unique, summarized answer; information that is left out of the final summary will simply be unavailable. This raises the concern of completeness: besides ensuring that the information provided could be trusted, we wanted to guarantee that the posed question was being answered thoroughly. We adopted the general definition of Coverage as the portion of relevant information about a certain subject that is contained in a document [12]. We proceeded by treating each answer to a question $q$ as a separate document and we retrieved through the Yahoo! Answers API a set $TK^q$ (Total Knowledge) of 50 answers to questions similar to $q$: the knowledge space of $TK^q$ was chosen to approximate the entire knowledge space related to the queried question $q$. We calculated Coverage as a function of the portion of answers in $TK^q$ that presented semantic overlap with $a$.

$$C(a, q) = \sum_{c_i \in a} \gamma(c_i) \cdot tf(c_i, a) \quad (5.3)$$

The Coverage measure for an answer $a$ was calculated as the sum of term frequency $tf(c_i, a)$ for concepts in the answer itself, weighted by a concept importance function, $\gamma(c_i)$, for concepts in the total knowledge space $TK^q$. $\gamma(c)$ was defined as follows:

$$\gamma(c) = \frac{|TK^q|}{|TK^q|} \cdot \log_2 \frac{|TK^q|}{|TK^q,c|} \quad (5.4)$$

where $TK^q,c = \{d \in TK^q : \exists k \in d, k \bowtie c\}$.

The function $\gamma(c)$ of concept $c$ was calculated as a function of the cardinality of set $TK^q$ and set $TK^q,c$, which was the subset of all those answers $d$ that contained at least one concept $k$ which presented semantical overlap with $c$ itself. A similar idea of knowledge space coverage is addressed by [12], from which formulas (5.3) and (5.4) were derived.

Example

Consider again Question $Q_{00}$; thanks to the Yahoo Answers API method mentioned above, it would be possible to retrieve the set “Total Knowledge” of answers to similar questions:

$$TK^{Q_{00}} = \{A_{01}, A_{02}, A_{03}, \ldots A_{50}\}$$

such limit was imposed by the current version of the API
In order to calculate the Coverage $C(A00, Q00)$, all answers would be turned into their concept representation (bag-of-BEs). Each concept from $A00$ would then be treated separately. Let’s consider the first one, $c00_1$: protect|yourself”; to calculate the corresponding concept importance $\gamma(c00_1)$ we would consider all answers contained in $TK^{Q00}$ in turn; the equivalence classes of each concept of each answer would be intersected with the equivalence class of $c00_1$ (comparing for semantic overlap); out of 50, let’s suppose that 10 answers from $TK^{Q00}$ contained concepts which semantically overlapped. Then:

$$\gamma(c00_1) = \frac{10}{50} \cdot \log_2 \frac{50}{10} \approx 0.4644$$

The term frequency of $c00_1$ would also be calculated. Suppose $tf(c00_1, A00) = 0.1$. We would now be able to calculate the product between concept importance ($\gamma$) and term frequency ($tf$); similar values would be calculated for all other concepts $c00_j$ in $A00$ and added together to compute the final Coverage value for the whole answer: $C(A00, Q00) = 0.0464 + \ldots \gamma(c00_j) \cdot tf(c00_j, A00) \ldots$

## 5.5 Relevance

To this point, we have addressed matters of trustfulness and completeness. Another widely shared concern for Information Retrieval systems is Relevance to the query. We calculated relevance by computing the semantic overlap between concepts in the answers and the question. Intuitively, we reward concepts that express meaning that could be found in the question to be answered.

$$R(c, q) = \frac{|q^c|}{|q|} \quad (5.5)$$

where $q^c = \{k \in q : k \equiv c\}$

The Relevance measure $R(c, q)$ of a concept $c$ with respect to a question $q$ was calculated as the ratio of the cardinality of set $q^c$ (containing all concepts in $q$ that semantically overlapped with $c$) normalized by the total number of concepts in $q$.

### Example

Consider once again $c00_1$ ("protect|yourself"), the first concept in answer $A00$; its equivalence class would be compared to the equivalence classes of each of the concepts contained in question $Q00$ and the number of overlaps would be stored as $|Q00|^{c00_1}|$. The fraction of semantically overlapping concepts over the total number of concepts in $Q00$ is the measure of the relevance of concept $c00_1$ with regards to $Q00$. Now suppose $|Q00| = 3$, meaning that the question contained three concepts; $c00_1$ semantically overlaps with exactly one concept from $Q00$: “protect|myself”


5.6 Novelty

In the scenario we proposed, the user’s information need is addressed in the form of a unique, summarized answer; information that is left out of the final summary will simply be unavailable. This raises the concern of completeness: besides ensuring that the information provided could be trusted, we wanted to guarantee that the posed question was being answered thoroughly. Another property we found desirable, was to minimize redundancy of information in the final summary. Since all elements in $TA^q$ (the set of concepts in all answers to $q$) would be used for the final summary, we positively rewarded concepts that were expressing novel meanings.

The Novelty measure $N(c, q)$ of a concept $c$ with respect to a question $q$ was calculated as the ratio of the cardinality of set $TA^{q,c}$ over the cardinality of set $TA^q$; $TA^{q,c}$ was the subset of all concepts in all answers to $q$ that presented semantic overlap with $c$.

Example

Suppose we were to consider once again concept $c001$ (“protect|yourself”) from answer $A00$; the procedure to calculate its Novelty would be conceptually similar to the one for the calculation of Relevance. Suppose $Q00$ had a total of 50 concepts ($|TA^{Q00}| = 50$ counting all concepts in all answers other than $A00$) and that among those, 5 semantically overlapped with $c001$ ($|TA^{Q00},c001| = 5$). The concept would then be assigned to following Novelty value:

$$N(c001, Q00) = 1 - \frac{|TA^{q,c}|}{|TA^q|}$$

5.7 The concept scoring functions

We have now determined how to calculate the scores for each property in formulas (5.2), (5.3), (5.5) and (5.6); under the assumption that the Quality and Relevance of a concept are the same
of its answer, every concept \( c \) part of an answer \( a \) to some question \( q \), could be assigned a score vector as follows:

\[
\Phi^c = (Q(\Psi^a), C(a, q), R(c, q), N(c, q))
\]

What we needed at this point was a function \( S \) of the above vector which would assign a higher score to concepts most worthy of being included in the final summary. Our intuition was that since Quality, Coverage, Novelty and Relevance were all virtues properties, \( S \) needed to be monotonically increasing with respect to all its dimensions. We designed two such functions. Function (5.7), which multiplied the scores, was based on the probabilistic interpretation of each score as an independent event. Further empirical considerations, brought us to later introduce a logarithmic component that would discourage inclusion of sentences shorter than a threshold \( t \) (a reasonable choice for this parameter is a value around 20). The score for concept \( c \) appearing in sentence \( s^c \) was calculated as:

\[
S^\Pi(c) = \prod_{i=1}^{4} (\Phi_i^c) \cdot \log_2(\text{length}(s^c))
\] (5.7)

A second approach that made use of human annotation to learn a vector of weights \( V = (v_1, v_2, v_3, v_4) \) that linearly combined the scores was investigated. Analogously to what had been done with scoring function (5.7), the \( \Phi \) space was augmented with a dimension representing the length of the answer.

\[
S^\Sigma(c) = \sum_{i=1}^{4} (\Phi_i^c \cdot v_i) + \text{length}(s^c) \cdot v_5
\] (5.8)

In order to learn the weight vector \( V \) that would combine the above scores, we asked three human annotators to generate question-biased extractive summaries based on all answers available for a certain question. We trained a Linear Regression classifier with a set \( Tr^S \) of labeled pairs defined as:

\[
Tr^S = \{ (\Phi^c, \text{length}(s^c)), \text{include}^c \}
\]

\( \text{include}^c \) was a boolean label that indicated whether \( s^c \), the sentence containing \( c \), had been included in the human-generated summary; \( \text{length}(s^c) \) indicated the length of sentence \( s^c \). Questions and relative answers for the generation of human summaries were taken from the “filtered dataset” described in Section 6.1.

### 5.8 Fusion as a Summarization Problem

The previous sections showed how we quantitatively determined which concepts were more worthy of becoming part of the final machine summary \( M \). The final step was to generate the summary itself by automatically selecting sentences under a length constraint. Choosing this constraint carefully demonstrated to be of crucial importance during the experimental phase. We again opted for a metadata-driven approach and designed the length constraint as a function of the lengths of all answers to \( q \) (\( TA^q \)) weighted by the respective Quality measures:

\[
\text{length}^M = \sum_{a \in TA^q} \text{length}(a) \cdot Q(\Psi^a)
\] (5.9)

The intuition was that the longer and the more trustworthy answers to a question were, the more space was reasonable to allocate for information in the final, machine summarized answer \( M \).
$M$ was generated so as to maximize the scores of the concepts it included. This was done under a maximum coverage model by solving the following Integer Linear Programming problem:

\[
\text{maximize: } \sum_i S(c_i) \cdot x_i \tag{5.10}
\]

subject to:

\[
\sum_j \text{length}(j) \cdot s_j \leq \text{length}^M
\]

\[
\sum_j s_j \cdot \text{occ}_{ij} \geq c_i \quad \forall i \tag{5.11}
\]

\[
\text{occ}_{ij}, x_i, y_j \in \{0, 1\} \quad \forall i, j
\]

\[
\text{occ}_{ij} = 1 \text{ if } c_i \in s_j, \quad \forall i, j
\]

\[
x_i = 1 \text{ if } c_i \in M, \quad \forall i
\]

\[
y_j = 1 \text{ if } s_j \in M, \quad \forall j
\]

The integer variables $x_i$ and $y_j$ were equals to one if the corresponding concept $c_i$ and sentence $s_j$ were included in $M$. Similarly $\text{occ}_{ij}$ was equal to one if concept $c_i$ was contained in sentence $s_j$. We maximized the sum of scores $S(c_i)$ (for $S$ equals to $S^\Pi$ or $S^\Sigma$) for each concept $c_i$ in the final summary $M$. We did so under the constraint that the total length of all sentences $s_j$ included in $M$ must be less than the total expected length of the summary itself. In addition, we imposed a consistency constraint: if a concept $c_i$ was included in $M$, then at least one sentence $s_j$ that contained the concept must also be selected (constraint (5.11)). The described optimization problem was solved using lp_solve\(^4\).

We conclude with an empirical side note: since solving the above can be computationally very demanding for large number of concepts, we found performance-wise very fruitful to skim about one fourth of the concepts with lowest scores.

\(^4\)the version used was lp\_solve 5.5, available at \url{http://lpsolve.sourceforge.net/5.5}
Chapter 6

Experiments

“Dis medicine,” he said hoarsely
“e good for black man?”
“Na fine for black man.”
“Black man no go die?”
“At all, my friend.” […]
‘You like you go try dis medicine?’ I asked casually.

[GERALD DURRELL, The Bafut Beagles]

In this chapter we present the experimental results that demonstrate the degree of effectiveness of our methods. They are run on the prototype that has been presented in the previous chapters. The interested reader is encouraged to consult the experiments manual in Appendix C and contact the author for further help in reproducing the results that follow.

6.1 Datasets and Filters

In the following section I describe the dataset on which we conducted our experiments: how it has been obtained, what its statistical properties are and how it has been filtered and subdivided.

The initial dataset was composed of 216,563 questions and 1,982,006 answers written by 171,676 user in 100 categories from the Yahoo! Answers portal. We will refer to this dataset as the “unfiltered version”. The metadata described in Chapter 5 was extracted and normalized; quality experiments (Section 6.2) were then conducted. The unfiltered version was later reduced to 89,814 question-answer pairs that showed statistical and linguistic properties which made them particularly adequate for our purpose. In particular, trivial, factoid and encyclopedia-answerable questions were removed by applying a series of patterns for the identification of complex questions. The work by [4] indicates some categories of questions that are particularly suitable for

1The reader is encouraged to contact the authors regarding the availability of data and filters described in this Section.
summarization, but due to the lack of high-performing question classifiers we resorted to human-crafted question patterns. Some pattern examples are the following:

- {Why, What is the reason} [...]  
- How {to, do, does, did} [...]  
- How {is, are, were, was, will} [...]  
- How {could, can, would, should} [...]  

We also removed questions that showed statistical values outside of convenient ranges: the number of answers, length of the longest answer and length of the sum of all answers (both absolute and normalized) were taken in consideration. In particular we discarded questions with the following characteristics:

- there were less than three answers ²  
- the longest answer was over 400 words (likely a copy-and-paste)  
- the sum of the length of all answers outside of the (100, 1000) words interval  
- the average length of answers was outside of the (50, 300) words interval  

At this point a second version of the dataset was created to evaluate the summarization performance under scoring function (5.7) and (5.8); it was generated by manually selecting questions that arouse subjective, human interest from the previous 89,814 question-answer pairs. The dataset size was thus reduced to 358 answers to 100 questions that were manually summarized (refer to Section 6.3). From now on we will refer to this second version of the dataset as the “filtered version”.

### 6.2 Quality Assessing

In Chapter 5 we claimed to be able to identify high quality content. To demonstrate it, we conducted a set of experiments on the original unfiltered dataset to establish whether the feature space $\Psi$ was powerful enough to capture the quality of answers; our specific objective was to estimate the amount of training examples needed to successfully train a classifier for the quality assessing task. The Linear Regression³ method was chosen to determine the probability $Q(\Psi^a)$ of $a$ to be a best answer to $q$; as explained in Chapter 5, those probabilities were interpreted as quality estimates. The evaluation of the classifier’s output was based on the observation that given the set of all answers $TA^q$ relative to $q$ and the best answer $a^*$, a successfully trained classifier should be able to rank $a^*$ ahead of all other answers to the same question. More precisely, we defined Precision as follows:

$$\text{Precision} = \frac{|\{q \in T^Q \mid \forall a \in TA^q, Q(\Psi^a) > Q(\Psi^{a^*})\}|}{|T^Q|}$$

where the numerator was the number of questions for which the classifier was able to correctly rank $a^*$ by giving it the highest quality estimate in $TA^q$ and the denominator was the total number of examples in the training set $T^Q$. Figure 6.1 shows the precision values (Y-axis) in identifying

²Being too easy to summarize or not requiring any summarization at all, those questions would not constitute an valuable test of the system’s ability to extract information.
³Performed with Weka 3.7.0 available at [http://www.cs.waikato.ac.nz/~ml/weka](http://www.cs.waikato.ac.nz/~ml/weka)
6.3 Evaluating Answer Summaries

The objective of our work was to summarize answers from cQA portals. Two systems were designed: Table 6.1 shows the performances using function $S^\Sigma$ (see equation (5.8)), and function $S^\Pi$ (see equation (5.7)). The chosen best answer $a^*$ was used as a baseline. We calculated ROUGE-1 and ROUGE-2 scores against human annotation on the filtered version of the dataset presented.

<table>
<thead>
<tr>
<th>System</th>
<th>$a^*$ (baseline)</th>
<th>$S^\Sigma$</th>
<th>$S^\Pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUGE-1_R</td>
<td>51.7%</td>
<td>67.3%</td>
<td>67.4%</td>
</tr>
<tr>
<td>ROUGE-1_P</td>
<td>62.2%</td>
<td>54.0%</td>
<td>71.2%</td>
</tr>
<tr>
<td>ROUGE-1_F</td>
<td>52.9%</td>
<td>59.3%</td>
<td>66.1%</td>
</tr>
<tr>
<td>ROUGE-2_R</td>
<td>40.5%</td>
<td>52.2%</td>
<td>58.8%</td>
</tr>
<tr>
<td>ROUGE-2_P</td>
<td>49.0%</td>
<td>41.4%</td>
<td>63.1%</td>
</tr>
<tr>
<td>ROUGE-2_F</td>
<td>41.6%</td>
<td>45.9%</td>
<td>57.9%</td>
</tr>
<tr>
<td>ROUGE-L_R</td>
<td>50.3%</td>
<td>65.1%</td>
<td>66.3%</td>
</tr>
<tr>
<td>ROUGE-L_P</td>
<td>60.5%</td>
<td>52.3%</td>
<td>70.7%</td>
</tr>
<tr>
<td>ROUGE-L_F</td>
<td>51.5%</td>
<td>57.3%</td>
<td>65.1%</td>
</tr>
</tbody>
</table>

Table 6.1: Summarization Evaluation on filtered dataset (refer to Section 6.1 for details). ROUGE-L, ROUGE-1 and ROUGE-2 are presented; for each, Recall (R), Precision (P) and F-1 score (F) are given.

4ROUGE is currently recognized as the standard to evaluate machine summaries: implementation available at: [http://berouge.com/default.aspx](http://berouge.com/default.aspx)
Figure 6.2: Increase in ROUGE-L, ROUGE-1 and ROUGE-2 performances of the $S^{\Pi}$ system as more measures are taken in consideration in the scoring function, starting from Relevance alone (R) to the complete system (RQNC). F-1 scores are given.

In Section 6.1. The filtered dataset consisted of 358 answers to 100 questions. For each questions $q$, three annotators were asked to produce an extractive summary of the information contained in $TA^q$ by selecting sentences subject to a fixed length limit of 250 words. The annotation resulted in 300 summaries (larger-scale annotation is still ongoing). For the $S^{\Sigma}$ system, 200 of the 300 generated summaries were used for training and the remaining were used for testing (see the definition of $Tr$ $S$ Section 5.7). Cross-validation was conducted. For the $S^{\Pi}$ system, which required no training, all of the 300 summaries were used as the test set.

$S^{\Sigma}$ outperformed the baseline in Recall (R) but not in Precision (P); nevertheless, the combined F-1 score (F) was sensibly higher (around 5 points percentile). On the other hand, our $S^{\Pi}$ system showed very consistent improvements of an order of 10 to 15 points percentile over the baseline on all measures; we would like to draw attention on the fact that even if Precision scores are higher, it is on Recall scores that greater improvements were achieved. This, together with the results obtained by $S^{\Sigma}$, suggest performances could benefit from the enforcement of a more stringent length constraint than the one proposed in (5.9). Further potential improvements on $S^{\Sigma}$ could be obtained by choosing a classifier able to learn a more expressive underlying function.

In order to determine what influence the single measures had on the overall performance, we conducted a final experiment on the filtered dataset to evaluate (the $S^{\Pi}$ scoring function was used). The evaluation was conducted in terms of F-1 scores of ROUGE-L, ROUGE-1 and ROUGE-2. First only Relevance was tested (R) and subsequently Quality was added (RQ); then, in turn, Coverage (RQC) and Novelty (RQN); Finally the complete system taking all measures in consideration (RQNC). Results are shown in Figure 6.2. In general performances increase smoothly with the exception of ROUGE-2 score, which seems to be particularly sensitive to Novelty: no matter what combination of measures is used (R alone, RQ, RQC), changes in ROUGE-2 score remain under one point percentile. Once Novelty is added, performances rise abruptly to the system’s highest.
Chapter 7

Conclusions and Future Work

“You fit climb dat big stick?” I repeated, thinking he had not heard.
“Yes, sah” he said.
“For true?”
“Yes, sah, I fit climb um. I fit climb stick big pass dat one.” [...]
“Right, you go come tomorrow for early-early morning time.”

[GERALD DURRELL, The Bafut Beagles]

We presented a framework to generate trustful, complete, relevant and succinct answers to questions posted by users in cQA portals. We made use of intrinsically available metadata along with concept-level multi-document summarization techniques. Furthermore, we proposed an original use for the BE representation of concepts and tested two concept-scoring functions to combine Quality, Coverage, Relevance and Novelty measures. Evaluation results on human annotated data showed that our summarized answers constitute a solid complement to best answers voted by the cQA users.

We are in the process of building a system that performs on-line summarization of large sets of questions and answers from Yahoo! Answers. Larger-scale evaluation of results against other state-of-the-art summarization systems is ongoing.

We conclude by discussing a few alternatives to the approaches we presented. The \( length^M \) constraint for the final summary (Chapter 5), could have been determined by making use of external knowledge such as \( TK^q \); since \( TK^q \) represents the total knowledge available about \( q \), a coverage estimate of the final answers against it would have been ideal. Unfortunately the lack of metadata about those answers prevented us from proceeding in that direction. This consideration suggests the idea of building \( TK^q \) using similar answers in the dataset itself, for which metadata is indeed available. Furthermore, similar questions in the dataset could have been used to augment the set of answers used to generate the final summary with answers coming from similar questions. [25] presents a method to retrieve similar questions that could be worth taking in consideration for the task. We suggest that the retrieval method could be made Quality-aware. A Quality feature space for questions is presented by [13] and could be used to rank the quality of questions in a way similar to how we ranked the quality of answers.

The Quality assessing component itself could be built as a module that can be adjusted to the kind of Social Media in use; the creation of customized Quality feature spaces would make it possible
to handle different sources of UGC (forums, collaborative authoring websites such as Wikipedia, blogs etc.). A great obstacle is the lack of systematically available high quality training examples: a tentative solution could be to make use of clustering algorithms in the feature space; high and low quality clusters could then be labeled by comparison with examples of virtuous behavior (such as Wikipedia’s Featured Articles). The quality of a document could then be estimated as a function of distance from the centroid of the cluster it belongs to. More careful estimates could take the position of other clusters and the concentration of nearby documents in consideration.

Finally, in addition to the chosen best answer, a DUC-styled query-focused multi-document summary could be used as a baseline against which the performances of the system can be checked.
Bibliography


Appendix A

Example Run

Follows an interesting example run of the prototype presented in Chapter 5. For details on the prototype implementation of the presented framework, please refer to Appendix C: the source code and data are available on demand! The question here considered was taken from the Yahoo! Answers portal¹. The summarized answer is composed of five portions of text coming from different answers and was generated with the $S^2$ scoring function; the chosen best answer is presented for comparison. The richness of the content and the good level of readability made it a successful instance of metadata-aware summarization of information in cQA systems. Less satisfying examples include summaries to questions that require a specific order of sentences or a compromise between strongly discordant opinions; in those cases, the summarized answer might lack logical consistency.

***SUMMARIZED ANSWER***

[...] In addition if the bear actually approaches you or charges you... still stand your ground. Many times they will not actually come in contact with you, they will charge, almost touch you than run away. [...] The actions you should take are different based on the type of bear. for example adult Grizzlies can t climb trees, but Black bears can even when adults. They can not climb in general as thier claws are longer and not semi-retractable like a Black bears claws. [...] I truly disagree with the whole play dead approach because both Grizzlies and Black bears are opportunistic animals and will feed on carrion as well as kill and eat animals. Although Black bears are much more scavenger like and tend not to kill to eat as much as they just look around for scraps. Grizzlies on the other hand are very accomplished hunters and will take down large prey animals when they want. [...] I have lived in the wilderness of Northern Canada for many years and I can honestly say that Black bears are not at all likely to attack you in most cases they run away as soon as they see or smell a human, the only places where Black bears are agressive is in parks with visitors that feed them, everywhere else the bears know that usually humans shoot them and so fear us. [...]  

***BEST ANSWER***

Great question. I have done alot of trekking through California, Montana and Wyoming and have met Black bears (which are quite dinky and placid but can go nuts if they have babies), and have been half an hour away from (allegedly) the mother of all grizzley s whilst on a trail through Glacier National park - so some other trekkerers told me... What the park warden say is SING, SHOUT, MAKE NOISE...do it loudly, let them know you are there..they will get out of the way, it is a surprised bear wot will go mental and rip your little legs off..No fun permission: anything that will confuse them and stop them in their tracks...I have been told be an native american buddy that to keep a bottle of perfume in your pocket...throw it at the ground near your feet and make the place stink: they have good noses, them bears, and a mega concentrated dose of Britney Spears Obsessive Compulsive is gonna give em something to think about...Have you got a rape alarm? Def take that...you only need to distract them for a second then they will lose interest..Stick to the trails is the most important thing, and talk to everyone you see when trekking: make sure others know where you are.

¹http://answers.yahoo.com/question/index?qid=20060818062414AA7VldB
Appendix B

The Yahoo! Answers API

In Chapter 5.4 I mentioned how, given a question, a set of related answers (answers to similar questions) can be retrieved from the Question and Answering portal Yahoo! Answers through a dedicated API. To be granted use of the Yahoo! Answers API, a user is required to get a Yahoo! id and register an application id. The first can be obtained by signing up at http://yahoo.com; the latter, by filling in the registration form available at the following address: https://developer.apps.yahoo.com/wsregapp/. I created a yahoo id, “mattiatomasoni”, and registered an application by the name ”Qualitative Question Answering”: so doing I became part of the Yahoo! Developer Network (YDN). A developer can check his/her list of registered applications at https://developer.apps.yahoo.com/dashboard/. After the registration process, the API can be downloaded from http://developer.yahoo.com/download/.

The following resources of interest are available to YDN member:

- http://developer.yahoo.com, the YDN official website
- http://developer.yahoo.net/blog the YDN official blog
- http://developer.yahoo.net/forum, a discussion forum about YDN in general
- http://tech.groups.yahoo.com/group/ydn-answers, a mailing list for Yahoo! Answers API
- http://www.ygroupsblog.com/blog a blog affiliated with above Yahoo! Group

For further instructions on how to get the Yahoo! Answer API, please refer to http://developer.yahoo.com/answers/.
Appendix C

Prototype Installation Manual

In this chapter we give a short, operative manual which aims at guiding the reader through the process of repeating the experiments described in chapter 6 and published in “Metadata-aware Measures for Answer Formulation in Community Question Answering”1. What in the present chapter is referred to as QQA (Qualitative Question Answering), is a prototype that implements the conceptual framework presented in chapter 5 in the Java programming language. The reader is encouraged to contact the author regarding to possibility to obtain copies of the source code, data, Javadoc documentation and additionally required software in the versions listed below:

- “QQA”: prototype sources (Java)
- “qqa_dumb”: database dump (Mysql)
- Eclipse for Java Developer (suggested Helios Release)
- Java interpreter and compiler: java and javac (recommended 1.6 or higher)
- Mysql Server and Client
- mysql-connector-java-5.1.10 or higher
- Apache Ant (recommended 1.7.1 or higher)
- ROUGE (recommended 1.5.5 or higher)
- BEwT_E (recommended 0.3 or higher)
- Weka framework (recommended 3-7-0 or higher)

Create the qqa database

QQA works on a database of questions, answers and metadata. This can be obtained in two ways: a local empty database can be populate starting from the “qqa_dump.txt” file, or a database can be generate through the QQA prototype from text files (not recommended).

If you are planning to rebuild the database, the QQA/qqa_debug.xml (and the file build.xml in the same directory), need to be customized by adding the paths to the text files. Those files are property of Emory University. Please contact the author regarding their availability. The database can be rebuilt by invoking the following command from the QQA/build directory:

```
ant A.Upload_Database
```

1www.csai.tsinghua.edu.cn/~hml/papers/acl2010.pdf
Please note that this step can be very time consuming (a few hours with a modern PC at the time the research was carried out). The reader is instead encouraged to create a Mysql user:

```
GRANT usage ON *.* TO [user]@localhost IDENTIFIED BY '[password]';
```

and a database called “qqa” by invoking:

```
mysqladmin -uroot -p[password] create qqa
```

Populate then the database from the dump with:

```
mysql -uroot -p[password] qqa < qqa_dump.txt
```

Finally, grant access to the newly created user and configure QQA/src/qqa/db/MysqlConnect.java with the username and password used above.

Finally, in

**Build and Debug the QQA prototype in Eclipse**

Import the project in Eclipse: File → Import → General → Existing Project into Workspace. “Select root directory”: browse to QQA. Click on the Finish button.

Create a build configuration: Open the folder build inside the QQA project, right click on qqa_debug.xml → Properties. Under Run/Debug Settings → New → Ant Build → Ok → Targets; check desired targets and click Ok; “E_Answer_Questions_BA” will run the baseline, and “E_Answer_Questions_RQCN_Pie” complete system. Please read the following sections for a detailed descriptions of the other possible targets.

Run the build configuration: right click on qqa_debug.xml → Debug As → Ant Build. The console will pause waiting for the debugger to connect.

Create a debug configuration: Run → Debug Configurations. Double click on “Remote Java Application”: under Project use the Browse button to select QQA, under Host specify localhost and under Port specify 8000. Click on Debug button.

If you had set any breakpoints, the debugger will stop the execution and allow you to explore the code.