UMA ABORDAGEM MULTI-VISÃO PARA A ESTIMATIVA AUTOMÁTICA DA QUALIDADE DE CONTEÚDO COLABORATIVO NA WEB 2.0

DANIEL HASAN DALIP

UMA ABORDAGEM MULTI-VISÃO PARA A ESTIMATIVA AUTOMÁTICA DA QUALIDADE DE CONTEÚDO COLABORATIVO NA WEB 2.0

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Marcos André Gonçalves
Coorientador: Marco Cristo

Belo Horizonte

Junho de 2015

DANIEL HASAN DALIP

A MULTI-VIEW APPROACH FOR ASSESSING THE QUALITY OF COLLABORATIVELY CREATED CONTENT ON THE WEB 2.0

Thesis presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Marcos André Gonçalves
Co-advisor: Marco Cristo

Belo Horizonte
June 2015

Ficha catalográfica elaborada pela Biblioteca do ICEx - UFMG

Dalip, Daniel Hasan

D143a Uma abordagem multi-visão para a estimativa automática da qualidade de conteúdo colaborativo na web 2.0 / Daniel Hasan Dalip. — Belo Horizonte, 2015

xxv, 104 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais Departamento de Ciência da Computação.

Orientador: Marcos André Gonçalves
Coorientador: Marco Antônio Pinheiro de Cristo

1. Computacão - Teses. 2. World Wide Web (Sistema de recuperacão da informacão) - Teses. 3. Aprendizado do computador - Teses. I. Orientador. II. Coorientador. III. Título.

519.6*73(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Uma abordagem multi-visão para a estimativa automática da qualidade de conteúdo colaborativo na Web 2.0

DANIEL HASAN DALIP

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Marcos André Gonçalves

PROF. MARCOS ANDRÉ GONÇALVES - Orientador
Departamento de Ciência da Computação - UFMG

Marco Antonio P. de Cristo

PROF. MARCO ANTÔNIO PINHEIRO DE CRISTO - COORIENTADOR
Departamento de Ciência da Computação - UFAM

PROF. ALBERTO HENRIQUE FRADE LAENDER
Departamento de Ciência da Computação - UFMG

PROF. ESTEVAM RAFAEL HRUSCHKA JÚNIOR
Departamento de Computação - UFSCAR

Fabrício Benevenuto de Souza

PROF. FABRÍCIO BENEVENUTO DE SOUZA
Departamento de Ciência da Computação - UFMG

PROF. JOSÉ PALAZZO MOREIRA DE OLIVEIRA
Instituto de Informática - UFRGS

Pável Pereira Calado

PROF. PÁVEL PEREIRA CALADO
Instituto Superior Tecnico - Portugal

Belo Horizonte, 19 de junho de 2015.

Acknowledgments

Agradeço a todos que direta ou indiretamente ajudaram a realizar o meu trabalho, em especial:

À minha família pelo total apoio.

À Daiane Marques por todo amor e carinho e pelo apoio em todas as horas durante o doutorado.

Aos professores Marcos André Gonçalves, Marco Cristo e Pável Calado pela orientação, apoio e dedicação

Aos professores da banca de avaliação: Alberto Laender, Fabrício Benevenuto, Estevam Hruschka, José Palazzo e Renato Assunção pelas valiosas sugestões.

À todos os integrantes do grupo do Laboratório de Banco de Dados pela amizade, troca de ideias e açaís na FACE, em especial: Thiago Salles, Peterson, Allan Jones, Anderson, Evandrino, Guilherme Tavares, Moisés, Harlley, Thiago Cardoso, Carol Bigonha, Vitor Oliveira, Cristiano, Rafael Odon, Bruno Leite, Isac, Daniel Xavier, Sérgio Canuto, Clebson, Gabriela, Vitor Mangaravite, Thiago Henrique, Rodrigo, Rodrygo, Michelle Brito, Michele Brandão, Luciana Mauron, Laís, entre outros LBDistas. Aos amigos que ganhei na UFMG: Ismael, Denise, Glívia, Lídia, Lucas, Luís, Natália Sales, Simone, Sandra Ávila, Emiliana e Ana Paula. Além disso, à todos os demais amigos que me ajudaram (e me acalmaram) nesta jornada, em especial: Arthur, Matheus, Isabela, Rafael, Henise, Igor, Juliane.

Às funcionárias da secretaria: Sheila, Linda, Sonia, Cida, Renata, Juliana por sempre me auxiliarem com documentação, viagens e outras questões administrativas do curso.

À Magali Araújo, minha orientadora de graduação no Uni-BH, pelo incentivo, orientação e amizade.

Ao Google, CNPQ e FAPEMIG pelo auxilio financeiro destinado a realização desta pesquisa.

“A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê”

(Arthur Schopenhauer)

Resumo

A Web contitui um novo tipo de repositório do conhecimento humano em que o usuário não é apenas consumidor mas também produtor de conteúdo. Porém, tal liberdade traz consigo uma importante questão: como o usuário pode determinar a qualidade da informação que ele acessa? Nesta tese, propomos uma abordagem multi-visão para a estimativa da qualidade de conteúdo colaborativo, ou seja, aplicamos técnicas de aprendizado de máquina para combinar avaliações independentes de qualidade realizadas por diferentes conjuntos de indicadores semanticamente relacionados (visões) em um único valor representando a qualidade do conteúdo. Com isso, foi realizada uma análise profunda de nossa abordagem em dois domínios (Fórum de Perguntas e Respostas e Enciclopédias Colaborativas), na qual foi possível uma maior compreensão de quando e como nossa abordagem consegue melhorar a predição automática da qualidade do conteúdo. Também estudamos o impacto das visões e de seus atributos em cada domínio, além de propor novos atributos. As principais contribuições desta tese são: (1) proposta de uma abordagem multi-visão que utiliza-se de grupos (i.e., visões) de indicadores de qualidade; (2) proposta de indicadores para estimativa da qualidade em Fóruns de Perguntas e Respostas; (3) aplicação desta abordagem em Fóruns de Perguntas e Respostas e Enciclopédias Colaborativas onde obtemos uma melhoria de até 30% da estimativa da qualidade em comparação com os melhores *baselines* encontrados na literatura; (4) uma análise profunda do impacto, informatividade e correlação dos atributos e das visões dos domínios estudados.

Palavras-chave: Avaliação de qualidade, Wiki, Fóruns de Perguntas e Respostas, Aprendizado de Máquina, Qualidade da Informação.

Abstract

The Web contains a new type of repository for the human knowledge where users are able not only to consume, but also to produce content in a much faster and easier manner. However, such freedom also carries concerns about the quality of this content. In this thesis, we propose an automatic quality approach to assess the quality of collaborative generated content. To accomplish this, we adopt a multi-view approach to assess the quality of content, in other words, we apply machine learning (ML) techniques to combine independent assessments regarding different sets of semantically related quality indicators (i.e., *views*) into a single quality value. Then, we perform a thorough analysis of our approach in two different domains (Questions and Answer Forums and Collaborative Encyclopedias), which allowed us to better understand when and how the proposed multi-view approach is supposed to improve quality assessment. We also study the impact of the views and the features that compose them in each domain. To summarize, our main contributions include: (1) the proposal of a general multi-view approach that takes advantage of groups (i.e., views) of quality indicators;(2) the proposal of new features in Q&A Forum domain; (3) the application of this approach in 2 domains where we could achieve an improvement of up to 30% in quality assessment over the best baselines methods found in the literature; (4) a throughout feature and view analysis regarding impact, informativeness and correlatedness, considering both domains.

Keywords: Quality Assessment, Wiki, Q&A Forums, Machine Learning, Information Quality.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Preliminaries

The Web 2.0 has brought deep changes to the Internet, as users are now able not only to consume, but also to produce content. This change gave rise to new ways for creating knowledge repositories, to which anyone can freely contribute. Some examples of these repositories include blogs, forums, and collaborative encyclopedias (hereafter called Wikis, for short), whose collections of documents are maintained by the Web community itself [Krowne, 2003; Dondio et al., 2006b].

The proliferation of collaboratively generated content leads us to think that, in the near future, this content will be predominant on the Web. Currently, there is a very large number of hosting services that allow the free editing of its content by the end users [Fogg et al., 2003; Rubio et al., 2010]. Each of these services hosts numerous collections dedicated to specific communities and subjects, such as geographic information, sports, technology, science, TV shows, science fiction, books, general knowledge, among others.

The most popular of such services, Wikia[1], has grown from one hundred to several thousands of collections in just a few years, containing more than four million pages of rich content. Another example of how communities can produce collaborative content on a large scale is that of Wikipedia[2]. This online encyclopedia took only two years to reach as many articles as the Encyclopedia Britannica. It currently contains more than twenty five million articles, written in hundreds of different languages [3]. There are

[1] http://www.wikia.org

[2] http://www.wikipedia.org

[3] http://en.wikipedia.org/wiki/Wikipedia

also many blogs and micro-blogs, such as Twitter[4], where users can exchange opinions about diverse subjects, such as politics, daily life, culture, among others [Maged et al., 2006].

Another example of collaborative content are questions and answers forums (also known as Q&A Forums) which are specific forums where users can collaborate asking and answering questions regarding topics such as programming, math, and English language. A good example of this kind of forum is the Stack Overflow[5] which contains 3.3 millions of questions and 6.6 millions of answers about programming.

However, such freedom also carries an important issue: *given the rhetoric of democratic access to everything, by everyone, at any time, how can a user determine the quality of the information provided?* Currently, content generated in a more traditional, centralized manner, published using physical media, such as books or journals, is still naturally seen as being of higher quality and more trustworthy [Dondio et al., 2006b]. Nevertheless, the growth and dissemination of collaboratively created content is such that mechanisms to assess the quality and trust of this type of material should be provided.

To deal with this problem, many collaborative sites adopt quality control mechanisms, where the users can indicate the quality and appropriateness of the content and even the reputation of the editors. However, such manual assessment does not scale to the current rate of growth and change of these systems. As a consequence, several strategies to automatically estimate the quality of collaborative generated content have been proposed in the last few years.

To understand such strategies, it is worth to properly define *information quality* (IQ). According to Wang and Strong [1996], IQ is the information which fits on the consumers use. As observed in Ge and Helfert [2007], since information consumers are not very capable of finding errors in information and altering the way they use the information, an alternative definition of IQ takes a data perspective, i.e., IQ is the information that meets the specifications or requirements. For instance, in Wikipedia there are some requirements to an article be considered good, for example: it needs to have citations and an appropriate structure. In this sense, quality assessment algorithms attempt to estimate quality by means of the combination of statistical indicators that try to measure how well the information meets different requirements. For instance, it is expected that a good article has a length large enough to properly discuss a topic and it is cited as it provides reputable information. Thus, it is possible to learn how to combine these indicators to predict a quality level. Also, from a theoretical point

[4] `http://www.twitter.com`

[5] `http://stackoverflow.com`

of view, as highlighted by Wang and Strong [1996]; Tejay et al. [2006]; Ge and Helfert [2007], quality synthesizes the measurement of *various dimensions*.

This view of quality as a multi-dimensional concept suggests that it should be thought of as a combination of independent assessments where, as before, each assessment can be estimated from several statistical indicators. For example, the quality of a textual document can be viewed as a composition of dimensions such as clarity, factual accuracy and importance. In other words, quality is a multifaceted concept, in which each facet corresponds to an aspect that can be individually analyzed by an automated "expert." The "opinions" of these experts can then be combined for a final decision. We refer to this method as a *multi-view approach*. In such an approach, each view corresponds to a partition of the set of indicators where the indicators of a particular view are naturally seen as a group. Further, defined as partitions, the views cannot share indicators which implies that views are designed to be as much independent as possible from each other. In Machine Learning, this approach is somewhat similar to an ensemble technique of learning using multiple experts in two levels [Wolpert, 1992].

In this context, we here propose a general approach for combining statistical indicators to assess content quality in collections created collaboratively. It is based on a machine learning multi-view approach that takes advantage of the natural organization of the statistical indicators for several quality dimensions. We evaluated our approach by using indicators extracted from two domains, namely, Wikis and Q&A Forums. These indicators were grouped into specific sets of views that, when combined, led to improvements in quality evaluation.

1.2 Motivation

We identified some motivation topics for this thesis which will be discussed in this section.

1.2.1 Automatically Assessing the Quality of Content

Collaborative communities have already proposed manual techniques to treat the problem of quality of the content by means of the human judgment. Examples include:

- **Wikipedia** (`http://www.wikipedia.org`): In order to improve the quality of the Wikipedia articles, members of this collaborative encyclopedia created guidelines on how to properly write an article, and on what can be considered a good article[6,7,8,9]. Furthermore, using these guidelines, Wikipedia articles can be labeled with a rating regarding the perceived quality of its content (e.g., "stub" or "featured article") [Wikipedia, 2015a].

- **Q&A Forums:** To help users to find best answers for questions, Q&A forums hosted by Stack Exchange[10] (cf. Figure 1.1 for an example), sort answers according to its quality, as voted by the users. According to the Stack Overflow guide[11], a good answer, besides being correct, should be clear, provide examples, quote relevant material, be updated, and link to more information and further reading. Thus, since "quality" is a subjective feature, it is inferred from the opinion of the asker and from the votes received from other users.

  To accomplish this, the asker can choose which answer was the best, by her point of view, marking it with a "tick". In addition, each answer has a rating defined by the users which can down-vote or up-vote an answer (i.e., increase or decrease the rating). Thus, the system shows first the answer chosen by the asker as the best one, followed by the others ordered according to its rating.

  Note that only registered users can up-vote and down-vote questions and answers. Then, in order to obtain feedback from unregistered users, this system presents the following question "Was this post useful to you?", which anyone can answer.

- **Planet Math** (`http://planetmath.org`): As described by Krowne [2003], this is a math community where each article belongs to its author. Then, other people can suggest changes on the text and the author may accept these suggestions or not. If the author does not respond the users' suggestions, he may lose the ownership of the article, passing to any other member of the community who expresses interest in it.

- **Cucumis** (`http://www.cucumis.org`): as described by the website, it is a community of text translations. Each user sets their language skills by registering on the site. Then she can request translations of texts, using points gained when translating texts. The community itself assesses the translations done leaving comments whether the translation has any error and then, if the translation has a good quality by the point of view of the community, the translator earns points.

[6] `http://starwars.wikia.com/wiki/Wookieepedia:Featured_article_nominations`

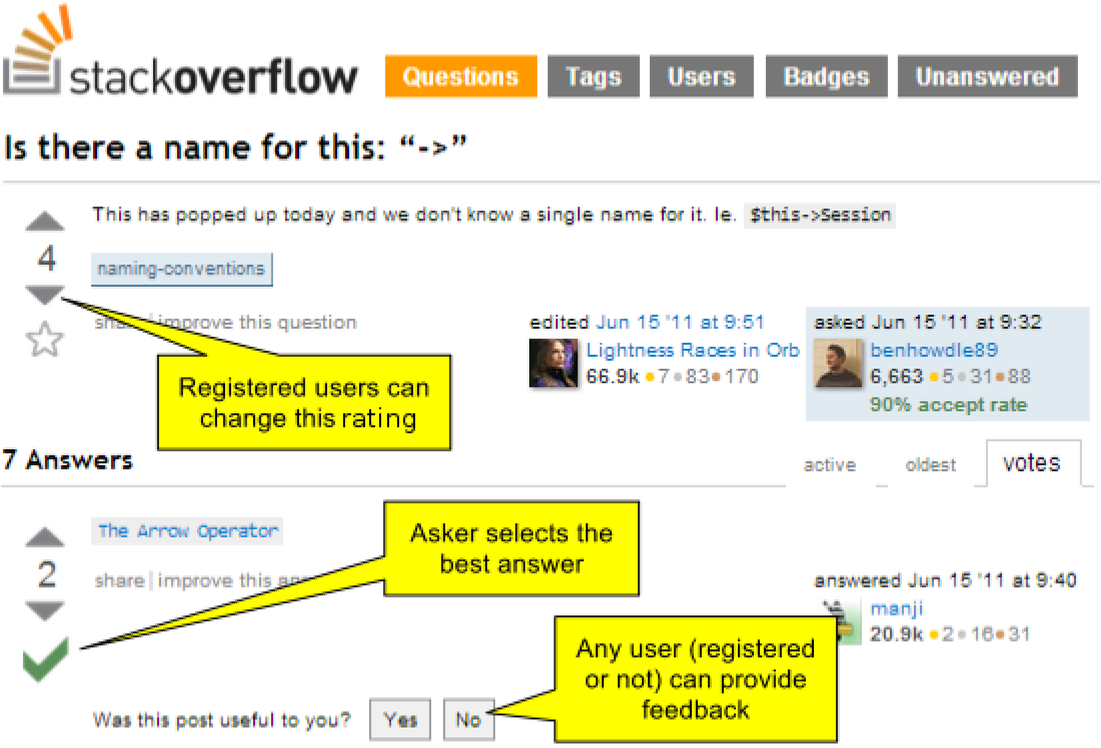
[7] `http://starwars.wikia.com/wiki/Wookieepedia:Good_article_nominations`

[8] `http://starwars.wikia.com/wiki/Wookieepedia:Featured_article_nominations`

[9] `https://en.wikipedia.org/wiki/Wikipedia:1.0/Criteria`

[10] `http://www.stackexchange.com`

[11] `http://meta.stackoverflow.com/questions/7656/how-do-i-write-a-good-answer-to-a-question`



Is there a name for this: "->"

This has popped up today and we don't know a single name for it. Ie. `$this->Session`

edited Jun 15 '11 at 9:51 Lightness Races in Orb 66.9k •7 •83 •170

asked Jun 15 '11 at 9:32 benhowdle89 6,663 •5 •31 •88 90% accept rate

7 Answers

The Arrow Operator

answered Jun 15 '11 at 9:40 manji 20.9k •2 •16 •31

Was this post useful to you? Yes No

Registered users can change this rating

Asker selects the best answer

Any user (registered or not) can provide feedback

Figure 1.1. The Stack Exchange Q&A Forums layout highlighting the tools which the user can use to manually assess the quality of a question and answer.

Even with policies to manually assess the quality, collaborative collections face problems, since given the current rate of growth and change of these systems [Voß, 2005], a manual revision process will eventually cease to be feasible. Moreover, in these collections, there is content less popular than others (e.g., from a specific subject), then it will take a longer time to receive ratings of quality by the user. Then, by using an automatic approach, we are able to predict the quality of the content at same the time that it was posted.

In addition, specifically in Q&A Forums, most strategies for automatic quality assessment found in the literature expect that the answer to be ranked has already received votes from the users [Suryanto et al., 2009; Shah and Pomerantz, 2010]. Thus, they are unable to assess the quality of answers to new or unpopular questions, which often do not contain such information.

1.2.2 Feature Combination

Taking into account the motivations presented in Section 1.2.1, some approaches have been proposed to facilitate automatic quality assessment [Dondio et al., 2006b; Rassbach et al., 2007; Dalip et al., 2011a], specially for Wikipedia. These approaches define quality features extracted from the article, and propose ways to combine them, to produce a final rating capturing its global quality. In Wiki, an article usually has multiple sources of evidence from which to extract such features. Examples are the history of reviews, the network of links among the articles, and the textual content and its structure. Particularly, in the approach that we propose [Dalip et al., 2009, 2011a], 68 quality indicators were extracted and combined using Support Vector Regression [Vapnik, 1995]. Experimental results showed significant gains over previous state of the art approach.

Common to all these studies is the fact that only a single training model, based on all available sources of information, is generated to predict quality. Then, in order to improve the prediction of quality, we propose to group our set of indicators [Dalip et al., 2009, 2011a] into semantically meaningful *views of quality*, i.e., groups of attributes, each representing a different type of evidence (textual information, link information, etc.). The quality predictions produced with such views are then combined, by means of meta-learning techniques, into one single quality value. This idea was motivated by the work of Kakade and Foster [2007], which demonstrated that the combination of views may improve the performance of machine learning methods. Since views represent different perceptions of a same concept (in our case, the relative quality of an article or answer), the combination of models created specifically for each view may improve results in a way similar to the combination of the opinions of different experts. Moreover, organizing features in such way, allows us to better exploit their different properties, thus improving the final prediction.

Note that, we can use the same approach in other domains. For example, in Q&A Forums we have multiple sources of information such as the ones derived from the textual content (e.g., structural, length, style and readability features), user and history of review features[12].

1.3 Hypothesis and Goals

As discussed previously, quality is a multifaceted problem in which each facet corresponds to a quality aspect (e.g., readability, style, organizational structure, link/citation coverage, review history). Then, our main hypothesis is that each aspect can be individually analyzed by an automated “expert” (learner) and the “opinions” of these experts can be combined for a final decision about the overall quality of a

[12] This is the case of Stack Exchange Q&A forums which someone can edit a question or answer and propose a modification which has to be authorized to the user.

particular item. Thus, the main goal of this research is to develop and evaluate advanced multi-view machine learning approaches for automatically combining several groups of semantically related quality indicators (aka *views*) in order to return a value representing the quality of an information item within a predetermined quality scale.

Thus, to accomplish this goals and test this hypothesis, our specific goals are:

1. Identify the “best” features and quality indicators that influence the users perception of quality of collaborative content in some specific domains (i.e., Wikis such as Wikipedia and Wikia, and Q&A Forums such as Stack Overflow[13]);

2. Determine the best way of grouping such features into semantically related groups (views);

3. Propose machine-learning-based approaches to derive quality “opinions” based on these groups and determine the best way for combining them;

1.4 Contributions

Our contributions are divided in (1) a feature representation and its impact for each tested domain; (2) a general multi-view approach and an in-depth study of views; (3) the usage of a feature selection approach in order not only to reduce the number of features without losing performance but also to do a feature analysis; (4) an application that helps the user to infer the quality in Wikipedia. Following we describe each of these contributions.

1. **Feature proposal and its impact**: We first studied the impact of features and quality indicators in Wikis and Q&A Forums domains. To accomplish this, we first did a thorough analysis of the capability of an automatic method to estimate content quality in Wikis. First, we extended our previous work [Dalip et al., 2009] which assessed the quality in Wikipedia, to assess the quality of two others Wikis, namely *Wookieepedia*[14], about the Star Wars universe, and *Muppet*[15], regarding the TV series “The Muppet Show”. Our consistent results throughout a large body of experiments and analyses allow us to make more generalizable conclusions than any previous work [Dalip et al., 2011a].

   In Q&A Forums we studied which, out of the 68 features previously used [Dalip et al., 2009, 2011a], could be useful in Q&A Forums together with others features

[13] http://stackoverflow.com

[14] http://starwars.wikia.com/

[15] http://muppet.wikia.com/

previously proposed in literature for this domain [Agichtein et al., 2008; Shah and Pomerantz, 2010; Burel et al., 2012], and new proposed features. Using our proposed approach, we were able to outperform a state of the art baseline with gains of up to 12% in NDCG, a metric used to evaluate rankings. We also conducted a comprehensive study of the features showing that, user and review features are the most important in the Q&A Forums domain [Dalip et al., 2013].

We detail the features (from Wiki and Q&A Forum) in Chapter 3. Some results of those work (adapted for multi-view) are presented in Chapter 5, for Wikis, and in Chapter 6 for Q&A Forums.

2. **General multi-view approach and view analysis**: In order to study better ways to combine the used features, motivated by the issues raised in the Section 1.2.2, we proposed an approach to assess the quality of collaboratively created content by organizing quality indicators into semantically related views and combining these views by means of meta-learning. With that, we did an in-depth analysis of this approach and of the impact of the views on quality assessment of collaborative content in Wikis and Q&A Forums. Our experimental results show that the proposed meta-learning approach is able to improve quality assessment over a state-of-the-art approach in five out of the six tested collections, with gains of up to 30.9%. In addition, we were able to reach more generalized conclusions and a better understanding from a qualitative point of view why some features performed well and others not, in order to better comprehend certain theoretical aspects of multi-view learning (e.g., when and why it is supposed to work) applied to quality estimation. Furthermore, we propose a general multi-view approach that takes advantage of groups (i.e., views) of quality indicators – this new approach generalizes and allows to better comprehend several previous solutions including some proposed by ourselves.

   The approach and preliminary results were published in the *Journal of Information and Data Management* [Dalip et al., 2012a] and a more detailed explanation of the approach together with an in-depth analysis of it were presented at the *2012 International Conference on Theory and Practice of Digital Libraries (TPDL)* [Dalip et al., 2012b]. We detail the approach in the Chapter 3, results of these work are presented in the Chapter 5, for Wikis, and Chapter 6 for Q&A Forums.

3. **Feature selection and analysis**: We also studied the impact of feature selection on our multi-view approach for assessing quality in all the studied collections. We

were motivated not only by the possibility of decreasing the complexity of the learned models but also by the opportunity of analyzing the importance of views and features. To accomplish this, we modeled the problem as a multi-objective search (using genetic algorithms) for the *smallest* set of features that is able to simultaneously *reduce* the quality assessment error. Results show that we can reduce the feature set to a fraction of 15% through 25% of the original set, while obtaining error rates comparable to the state of the art. We also investigated the impact and redundancy of different features and views for the Wikis domain. This work received the JCDL 2014 best student-paper prize [Dalip et al., 2014]. Chapter 2 presents the feature selection approach and results are presented in Chapter 5 and 6.

4. **Implemented Tool**: We were able to propose a tool, called GreenWiki[16], using some of the proposed metrics. This is a Wiki with some articles collected from Wikipedia and a panel of quality indicators about the article being read. Note that GreenWiki does not intend to evaluate the quality of an article, but rather, its goal is to present indicators that will help users get to their own conclusions about its quality [Dalip et al., 2011b].

Part of our work was also used to help other domains such as search query expansion [Brandão et al., 2014], to infer detractors and evangelists on Twitter [Bigonha et al., 2010], polarity detection on foursquare tips [Moraes et al., 2013] and sentiment analysis.

1.5 Thesis Organization

This thesis is organized as follows. Chapter 2 covers background. Chapter 3 presents our multi-view approach as well as the features we explore to represent the content for Wikis and Q&A Forums. Chapter 4 describes our datasets and the evaluation methodology. The experiments and their results are presented in Chapter 5, for Wikis, and in Chapter 6 for the Q&A Forums. Finally, Chapter 7 concludes this thesis and discusses future work.

[16] `http://www.dcc.ufmg.br/projetos/greenwiki`

Chapter 2

Background

In this chapter we detail the previous work related to this thesis. First, we explain quality criteria generally used to assess the quality of collaborative content. After that, we explain the supervised machine learning approach applied in this work. Then, we detail the feature selection approach which we have adapted to use in our context. Finally, we detail previous work on quality assessment of content specially in Q&A Forums and Wikis.

2.1 Quality Dimensions

Generally, in a quality estimation problem, we need to know which quality criteria a given domain takes into account. Several authors have proposed conceptual quality frameworks where quality is viewed as a multi-dimensional concept. Ge and Helfert [2007] classify such frameworks as hierarchical, ontological, based on semiotics, based on sources for metadata, based on products & services, and based on the sequence in which information is used. In this work, we adopt the semiotic framework proposed by Tejay et al. [2006]. This framework was derived from an extensive review of previous literature on information quality and is based on a data perspective where quality is related to the satisfaction of requirements. To accomplish this, Tejay et al. [2006] organized quality concepts in dimensions of quality, grouped by semiotic levels.

According to Tejay et al. [2006], semiotic can help to organize dimensions since it studies how a sign is created, processed and used. Sign, in the context of structured data quality, is the data itself. Then the study of signs can be divided in 4 semiotic levels: syntactic, semantic, pragmatic and empiric.

The Syntactic level is regarded about how the data is structured and formatted. The Semantic is concerned with the data meaning and interpretation while the pragmatic level is concerned with how people use the data. Finally, the Empiric level is about how the data is used/transmitted and what is the risk of being used in an unappropriated way. Thus, they organized quality dimensions according to each semiotic level. As examples we cite the syntactic dimension *conciseness* which captures how compact is the presentation of the data, the semantic dimension *ambiguity* that is concerned with how many interpretations are allowed by the data, the pragmatic dimension *relevancy* which is about the applicability of the data to the user needs and the empiric dimension *security* that is associated with usage data rights and level of protection against natural disasters.

Note that we can directly apply these semiotic levels in collaborative texts. Then, in this context, the syntactic level regards how the text was written and its structure. The semantic level concerns the meaning of the written text and the pragmatic level focuses on the relationship between the text and the behavior of its author in a given context. In this context, we do not use the empirical level since in our work we are interested in the content itself and not in the ways which the content is published.

Thus, based on previous work and on the publishing guides provided by collaborative free editing repositories[1,2,3,4], we adapted and expanded the list of quality dimensions presented in Tejay et al. [2006]. Table 2.1 presents all the dimensions and its meanings which are detailed in this section. As in Tejay et al. [2006], we also organize these dimensions in semiotic levels.

2.1.1 Syntactic Quality Dimensions

The syntactic quality dimensions are those related to how the text is presented. For instance, *clarity* assesses how tools and resources (e.g., images, examples) were used in order to facilitate the text understanding. *Organization* is the dimension associated with how the text is structured using, for example, sections and paragraphs.

Some others dimensions takes the amount of text into account, such as *level of detail* and *conciseness*. The first one regards to how much specific information is provided while the second regards to whether this information is presented in a compact way. For example, given a Wikipedia article, *level of detail* indicates how much detailed is each topic discussion and *conciseness* deals with how compact is the presentation of each topic.

[1] `http://starwars.wikia.com/wiki/Wookieepedia:Featured_article_nominations`

[2] `http://starwars.wikia.com/wiki/Wookieepedia:Good_article_nominations`

[3] `http://starwars.wikia.com/wiki/Wookieepedia:Featured_article_nominations`

[4] `https://en.wikipedia.org/wiki/Wikipedia:1.0/Criteria`

Quality Dimensions
Syntactic Level
Appearance: CI presentation aesthetic
Clarity: use of resources and patterns that favor understanding
Conciseness: whether content is presented in a compact way
Consistency: presentation of same (kind of) content in the same way
Correctness: whether content is free of lexical and grammar errors
Level-of-detail: how much specific information is provided
Organization: how content is structured regarding presentation
Readability: complexity of grammar and lexical usage
Semantic Level
Ambiguity: whether content allows multiple interpretations
Coherence: whether ideas are logically connected
Factual Accuracy: whether facts are correct
Informativeness: whether content conveys information or instruction
Meaningfulness: whether content has value, significance, purpose
Opinative or factual: whether information represents beliefs or facts
Redundancy: same information appearing multiple times
Reliability: whether content is trustworthy and free of editorial and systemic bias
Understandability: whether the information is easily comprehensible
Validity: if the content is supported by reliable sources
Pragmatic Level
Appropriateness: whether content is suitable for a particular use
Completeness: whether it provides the necessary depth, breadth and scope
Engagement: ability to influence or affect user
Importance: whether content is significant, influencing and worthy
Maturity and Stability: whether improvements are necessary
Neutral Point of View: whether information is unbiased and impartial
Relevancy: whether information is applicable and pertinent to the task at hand
Reputation: how well thought of is the content
Sufficiency: whether the adequate amount of information is provided
Timeliness: whether content is up to date
Usefulness: whether the content is beneficial to the user

Table 2.1. Quality dimensions and its meaning

Finally, the two others syntactic dimensions, *readability* and *correctness*, regards on how the text is written. *Readability* is concerned with how (unnecessarily) complex is the writing in terms of word and grammar usage. *Correctness* is related to how many typos and grammar mistakes can be found in the text.

2.1.2 Semantic Quality Dimensions

As discussed earlier, semantic level deals with the relationship between the collaborative text contents and its meaning. For example, *informativeness* dimension indicates if the content provides or discloses information, if it is instructive [Collins, 2003]. *Meaningfulness* is regarded to how much the content conveys meaning, function, or purpose, how valuable or significant it is [Collins, 2003].

Reliability indicates if the text is trustworthy. Wikipedia authors usually tries to improve the *reliability* by providing sources and confirming the validity of the information. A dimension related to *reliability* is *validity*, which is concerned with how much the content can be verified as true by, for example, being supported by reliable sources.

Identifying if a content is a personal opinion or a fact can be important depending on the type of text being written. And if the content conveys a fact, it is important to identify its correctness, which is captured by the *factual accuracy* dimension.

Ambiguity, *redundancy* and *coherence* are dimensions related to aspects that normally impact on the *understandability* of the text. According to Tejay et al. [2006], *ambiguity* is related to the possibility of the text allowing multiple interpretations. *Redundancy* indicates that the same information appear multiple times in the text. *Coherence* is observed when the ideas expressed by the text are logically connected without contradictions. Finally, *understandability* indicates the degree the text is clear, comprehensible and free of ambiguity. Note that, syntactic *readability* is similar to *understandability*, but a content can provide a good *readability* being simple and easy to read and, at the same time, difficult to understand.

2.1.3 Pragmatic Quality Dimensions

The pragmatic level focuses on the relationship between the collaborative text and the behavior of its author in a given context, that is, it attempts to grasp the intentions of the author. This way, these dimensions are subjective as their interpretation is dependent on their authors and contexts. As consequence, we expect much more disagreement regarding pragmatic dimensions (such as *sufficiency* or *relevance*) than about syntactic and semantic dimensions (such as *correctness* and *factual accuracy*).

Sufficiency and *completeness* are related to how much information there is for the task at hand. While the prior indicates if the content has *enough* information, the later is concerned with if it has *all* information. For instance, given the question "How to know the size of data types in C" in a Q&A forum, an answer indicating the use of the sizeof operator can be sufficient. However this answer would be certainly more complete if also includes information such as the header file to be included, compatibility issues, and example code.

Proposed by Dondio et al. [2006b], a *mature* and *stable* collaborative text is the one which authors came into consensus and is considered almost complete. When a text is immature, it tends to be unstable since it changes more over time in order to have its content completed. Furthermore, depending on the subject, it can evolve towards an "edit war" when editors argue about an issue and they keep reverting the revisions of each other. *Timeliness* is concerned with if the collaborative text is up to date. This dimension is specially important in texts about new topics.

A *neutral point of view*, as defined by the Wikipedia guidelines [Wikipedia, 2015b], configures the situation in which the information is expressed in an unbiased and impartial way. In addition, when it is necessary to express a certain point of view, the author is required to also express all alternative points of view fairly, without privileging any of them.

Engagement is associated with how much the text is able to make the user interested in its content. As an example of an engaging prose, Rosenzweig [2006] compares the article about Abraham Lincoln found in two online reference sources: Wikipedia and American National Biography Online. He shows that in spite of the Wikipedia article being as accurate and complete as the American National Biography Online one, the Wikipedia prose is less engaging, as illustrated by the following concluding quotes:

> Lincoln's death made the President a martyr to many. Today he is perhaps America's second most famous and beloved President after George Washington. Repeated polls of historians have ranked Lincoln as among the greatest presidents in U.S. History [5].

> The republic endured and slavery perished. That is Lincoln's legacy [6].

The second one clearly provides a more engaging and concise prose. In this case, in particular, the lack of engagement of Wikipedia can be attributed to its enforcement of a neutral point of view that leads to a verbose and faltering writing characterized by inconclusive and unemphatic affirmatives.

Relevancy and *usefulness* are similar dimensions but, according to [Tejay et al., 2006], while relevancy is concerned with how pertinent is the information to the task at hand, usefulness has to do with how beneficial is the information to the user such that she would really use it somehow. For instance, among two relevant answers given to the question "how to open a file in Python?", in a Q&A forum, one of them can be more useful if it provides a snippet of Python code illustrating how to open a file than if it only describes the relevant API. In addition, *importance* is related to whether the content has a special relevance for the topic, person or task at hand (e.g., "Natural Selection" is normally regarded as more important than "Taxidermy" to the topic Biology).

Content *appropriateness* has to do with whether the content is suitable for a particular use and audience. Note that, content can be at the same time relevant and inappropriate. For example, in an article about "violent deaths", a graphic video

[5] `http://www.anb.org/articles/04/04-00631.html`

[6] `http://en.wikipedia.org/w/index.php?title=Abraham_Lincoln&oldid=27007980`

depicting a violent death can be relevant to the topic, although inappropriate for screening depending on venue (e.g., a major News portal), audience (e.g., children) etc.

Reputation is concerned with how presumed good and trustworthy is the content or its author(s). This dimension differs from semantic *reliability* since *reliability* is related to whether the text is written in a way that make people trust in the content, while *reputation* captures what people really think about the content (or its authors).

2.2 Supervised Machine Learning Methods

Supervised Machine learning techniques aims at learning, through data patterns, a way to reach some target result with less possible error [Mitchell, 1997].

To accomplish this we assume that we have access to some *training data* of the form $A = \{(\mathbf{a}_1, r_1)$, $(\mathbf{a}_2, r_2)$, ..., $(\mathbf{a}_n, r_n)\}$ where a_i has one or more features and a target value r_i. In this work, the term *feature* describes a statistic value that represents a measurement in order to help to predict a_i. For instance, if a_i is an Wikipedia article, one feature could represent its length. Then, machine learning methods tries to combines these features in order to the predicted result reach as closer as possible of r_i. Depending on the problem which we have at hand, r_i can have different values.

In some cases, r_i can represent finite values, for example in a problem of defining if an email is spam or not r_i can assume only two values, "yes" or "no". In this case, we say that it is a classification problem. In other cases, r_i can represent infinite real numbers and the goal is to predict this number. For example, predict the score of a football match or to predict the company month profit. Then, it is a regression.

In addition, there is other method to use when r_i is a real number representing just the order of the instance a_i among others. This is called Learning to Rank (L2R) which is a successful approach for the task of web search result ranking [Mohan et al., 2011]. In this approach, a_i is a query-document pair (d_i, q_i) which is also represented by a set of features and r_i represents numerical score indicating how relevant document d is to query q. These features are usually related to the similarity between the content of the document d and the query q.

In this work we are going to deal with regression and Learning to Rank. Thus, this section explains the two methods used in this work, namely SVR, which is for regression, and SVMRank which is for Learning to Rank. [Drucker et al., 1996; Joachims, 2002].

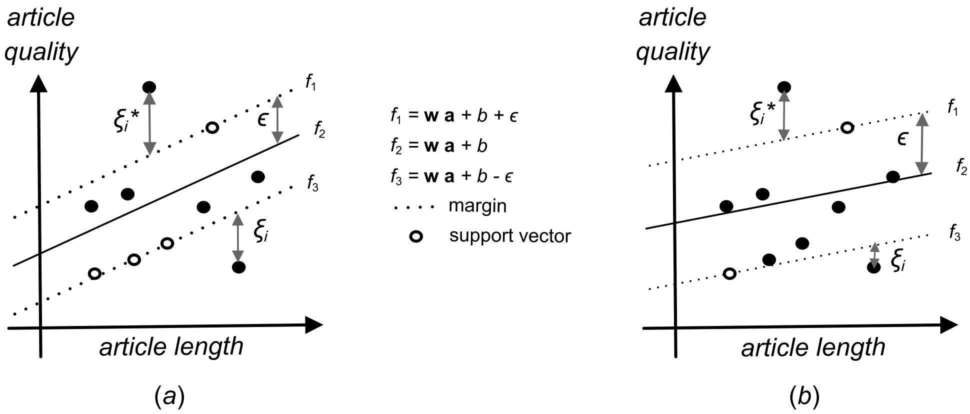


Figure 2.1. Regression problem with one numeric target (article quality) and ten articles (points), represented by a single feature (article length). Note that two articles, in both graphics, are considered examples of error because they lie outside the area delimited by the margins. Their distances to the margins are given by ξ_i^* and ξ_i, respectively. Figures (a) and (b) represent regressions performed using two different ϵ values.

2.2.1 SVR

Then, given the training data defined on the previous section $A = \{(\mathbf{a}_1, r_1),\ (\mathbf{a}_2, r_2),$ ..., $(\mathbf{a}_n, r_n)\}$ the problem here is to find a function f which approximates the mapping between input and output variables. In our case, the input variables are given by the features used to represent a_i and the output variable is the target value (r_i). We refer as *error* to the difference between the prediction and the true value $(f(\mathbf{a}) - r), r \in \mathbb{R}, \mathbf{a} \in A_v$. The magnitude of the error is measured by a loss function. The main idea behind SVR is to use a loss function (called *ϵ-insensitive*) that does not consider error values situated within a certain distance of the true value. One way of visualizing this method is to consider a region of size $\pm\epsilon$ around the hypothesis function, where ϵ denotes a margin. Any training point lying outside this region is considered an example of an error, as illustrated by Fig. 2.1(a), where f_1 and f_3 represent the margins around hypothesis function f_2. In this work, our goal is to find a function $f : A \rightarrow \mathbb{R}$ that has at most ϵ deviation from the output variable $r \in \mathbb{R}$, for all the training data.

In SVR, the input a is first mapped onto an m-dimensional feature space using some nonlinear mapping Φ. A linear model is then constructed in this feature space. More formally, the linear model $f(\mathbf{a}, \mathbf{w})$ is given by $f(\mathbf{a}, \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{a}) \rangle + b$, where $\mathbf{w}$ is a weight vector of m feature values, b is the bias term, and $\langle \mathbf{w}, \Phi(\mathbf{a}) \rangle$ denotes the inner product between $\mathbf{w}$ and $\Phi(\mathbf{a})$. The quality of estimation is measured by the ϵ-insensitive loss function $L^\epsilon(r, f(\mathbf{a}, \mathbf{w}))$ defined in Eq. 2.1.

$$L^{\epsilon}(r, f(\mathbf{a}, \mathbf{w})) = \begin{cases} 0 & \text{if } |r - f(\mathbf{a}, \mathbf{w})| \leq \epsilon \\ |r - f(\mathbf{a}, \mathbf{w})| - \epsilon & \text{otherwise} \end{cases} \tag{2.1}$$

SVR performs a linear regression in the high-dimension feature space using the ϵ-insensitive loss function while it tries, at the same time, to reduce the model complexity by minimizing the norm of $\mathbf{w}$. The linear regression of the loss function is performed by minimizing error estimates $(r - f(\mathbf{a}, \mathbf{w}))$ and $(f(\mathbf{a}, \mathbf{w}) - r)$, measured, respectively, by non-negative slack variables ξ_i^* and ξ_i. If we consider f_1 the margin above f and f_3 the margin below f, then ξ_i^* measures deviations above f_1 whereas ξ_i measures deviations below f_3, as shown in Fig. 2.1. Thus, SVR can be formulated as the convex optimization problem of minimizing:

$$\frac{1}{2}\|\mathbf{w}\ \|^2 + C\sum_{i=1}^{n} (\xi_i + \xi_i^*) \tag{2.2}$$

subject to:

$$\begin{aligned} |r_i - f\left(\mathbf{a}_i, \mathbf{w}\right)| &\leq \epsilon + \xi_i^* \\ |f\left(\mathbf{a}_i, \mathbf{w}\right) - r_i| &\leq \epsilon + \xi_i \\ \xi_i, \xi_i^* &> 0, 0 < i \leq n \end{aligned}$$

where $C > 0$ is a constant parameter. This optimization problem can be transformed into its dual problem, whose solution is given by Eq. 2.3:

$$f(\mathbf{a}) = \sum_{i=1}^{n_{\mathrm{SV}}} (\alpha_i + \alpha_i^*)\, \kappa\left(\mathbf{a}_i, \mathbf{a}\right), \text{ subject to } 0 < \alpha_i, \alpha_i^* \leq C \tag{2.3}$$

where n_{SV} is the number of support vectors (vectors lying on the margins, depicted as white circles in Fig. 2.1) and κ is an inner product function (*kernel function*) defined as $\kappa(\mathbf{a}_i, \mathbf{a}) = \sum_{j=1}^{m}(\Phi_j(\mathbf{a_i})\Phi_j(\mathbf{a}))$.

Note that SVR estimation accuracy depends on a good setting for C, ϵ and the kernel parameters. C determines the trade-off between model complexity (flatness) and the degree to which deviations larger than ϵ are tolerated. If C is large, the objective becomes simply to minimize the equation $\frac{1}{n}\sum_{i=1}^{n} L^{\epsilon}\left(r_i, f\left(\mathbf{a}_i, \mathbf{w}\right)\right)$. Parameter ϵ controls the width of the ϵ-insensitive zone, used to fit the training data. Bigger ϵ-values use fewer support vectors, at the expense of providing more “flat” estimates, as we can see in Fig. 2.1(a) and Fig. 2.1(b).

We have chosen SVR due to its advantages over other methods, such as the presence of a global minimum solution resulting from the minimization of a convex programming problem, relatively fast training speed and the capability of dealing with a sparse feature space [Chu et al., 2001]. In here, we solve the quadratic optimization problem given by Eq. 2.3 using the SVMLIB package [Chang and Lin, 2001]. In our experiments we have used a *radial basis function* (RBF) as κ. Other parameters, were chosen using cross-validation within the training set [Mitchell, 1997], with the data scaling and parameter selection tool provided by the SVMLIB package [Hsu et al., 2000].

2.2.2 SVMRank

In SVMRank we assume that we have access to the same *training data* $A = \{(a_1, r_1), ..., (a_n, r_n)\}$ but now a_i is a query-document pair with their associated relevance ratings $r_i \in \mathcal{R}$ such that if $r_i > r_j$ then $\mathbf{a}_i$ should be ordered before $\mathbf{a}_j$ ($\mathbf{a}_j \succ \mathbf{a}_i$). Thus, we are interested in learning ranking function $f(\mathbf{a}_i)$ such that $f(\mathbf{a}_i) > f(\mathbf{a}_j)$ if $\mathbf{a}_j \succ \mathbf{a}_i$. We can use this training data to learn a linear $f(\mathbf{a}_i, \mathbf{w})$, by observing that, for all pairs of answers $(\mathbf{a}_i, \mathbf{a}_j)$ where $r_i > r_j$, $f(\mathbf{a}_i, \mathbf{w}) > f(\mathbf{a}_j, \mathbf{w})$ iff $\mathbf{w} \cdot \mathbf{a}_i > \mathbf{w} \cdot \mathbf{a}_j$. The solution for this problem can be approximated by minimizing:

$$\frac{1}{2} \|\mathbf{w}\ \|^2 + C \sum \xi_{ij} \tag{2.4}$$

subject to:

$$\forall \{(\mathbf{a_i}, \mathbf{a_j}) : r_i < r_j \in \mathbb{R}\} \colon w \cdot \mathbf{a}_i \geq w \cdot \mathbf{a}_j + 1 - \xi_{ij}$$
$$\forall_{ij} \colon \xi_{ij} \geq 0$$

where $C > 0$ is a constant parameter. Note that $\mathbf{w} \cdot \mathbf{a}_i \geq \mathbf{w} \cdot \mathbf{a}_j + 1 - \xi_{ij}$ can be rewritten as $\mathbf{w} \cdot (\mathbf{a}_i - \mathbf{a}_j) \geq 1 - \xi_{ij}$, that is, this optimization is equivalent to the classification of the difference between the vectors $\mathbf{a}_i$ and $\mathbf{a}_j$.

Thus, SVMRank solves the classification problem of determining if a given pair of instances is correctly or incorrectly ordered. As before, we solve the optimization problem given by Eq. 2.4 using the SVM^{rank} package[7]. In our experiments we have used a linear kernel.

[7] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

2.3 SPEA2 Feature Selection Approach

Generally, machine learning approaches achieve good results by using a large number of features. However, extracting and computing all these features implies in a large processing cost. We argue that such cost can be avoided, since many of the features are likely to be redundant. In addition, using a small set of features will also be useful to achieve a better understanding of our approach and their impact on quality assessment. Because of that, we applied a genetic algorithm (GA) based approach for feature selection, described in this section. GAs are based on the theory of natural selection. Possible solutions are represented by individuals that compete to solve the problems and reproduce according to their success (usually called their *fitness*). The surviving individuals after many generations correspond to the solutions closer to optimal.

Although there are many studies regarding feature selection in traditional approaches, as far as we know, there is no studies of feature selection in multi-view approaches. The closer works are those from Opitz [1999] and Tsymbal et al. [2003]. The first applies a genetic algorithm to boosting and bagging ensemble methods. However, it uses a fitness function with just one objective, dividing the accuracy of the classifier by the number of features, which can lead to the loss of some good solutions. On the other hand, Tsymbal et al. [2003] exploit a hill climbing solution along with bayesian classifiers. However, greedy algorithms tend to be less robust than genetic algorithms [Vafaie and Imam, 1994]. Furthermore, greedy algorithms tend to discover just a local good solution, while genetic algorithms search for the best global solution.

Then, a good approach to address this problem is to consider it as a multi-objective one since we need to minimize the number of features while maximizing performance. Thus, the ideal set of features to assess article quality has to satisfy two basic criteria: (1) be small and (2) improve an evaluation metric. As these objectives may be conflicting, we do not expect to find a set that simultaneously optimizes both criteria. Thus, we are interested in a solution where none of the objectives can be improved without degrading the other one, i.e., a *nondominated* or *Pareto optimal* solution.

More formally, let the objectives x and y be improving an evaluation metric and minimizing the number of features used, respectively. We say that a solution i dominates a solution j when i is better than j in, at least, one objective and is not worse in the other. For example, if the evaluation metric is the regression error which we want to minimize, then, i dominates j when $((x_i < x_j) \land (y_i \leq y_j)) \lor ((x_i \leq x_j) \land (y_i < y_j))$. In addition, if, for a given solution i, there is no other solution j that dominates i, we say that i is a *nondominated solution* and it belongs to the *Pareto front*, as is illustrated

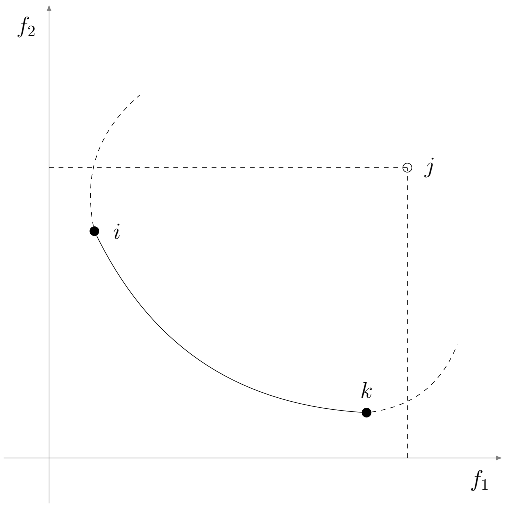


Figure 2.2. Two-dimensional example of the Pareto front, where the goal is to minimize along both the f_2 and f_1 axes. Solutions i and k do not dominate each other. However, solution j is dominated by solution i and k.

in Figure 2.2.

Among algorithms designed to find nondominated solutions, evolutionary approaches, such as genetic algorithms (GA), have become paramount. When applied to solve multi-objective problems, the set of solutions generated by GA allows the approximation of the entire Pareto front. Thus, we chose a well known multi-objective genetic algorithm, SPEA2 [Zitzler et al., 2001], to find a good set of features. Furthermore this multi-objective approach achieved good performance in several tasks [Zitzler et al., 2001]. In addition, in our case which we are also interested in analysing the solutions, another important advantage is the capability of maintaining the best solutions found. Then, to accomplish this, SPEA2 operates with a population of fixed size. This population starts with random individuals that represent different sets of features. The fitness of these individuals is calculated according to how well they meet their objective. In our case, the individual is successful if it is able to predict content quality with minimum error, while using as few features as possible. The most promising individuals are more likely to be chosen as parents of the next generation. They then breed new individuals to the next generation through genetic operations such as crossover and mutation. These steps repeat until a number of generations is reached, in a process described in Algorithm 1.

In Algorithm 1, $P_g = \{i_0, ..., i_n\}$ is a population set with n individuals per generation g. Each individual is represented by its chromosome, in which we perform the genetic operations. In our case, we model individual i chromosome as the vector

$c_i = \{f_0, f_1, ..., f_m\}$, such that $f_j = v$ if feature f_j is used in view v, and 0 otherwise. Each position f_j is called gene and an example of such representation is illustrated in Figure 2.3. We also implement elitism, that is, we allow some very successful individuals to be present unaltered in the next generation. To this effect, we use an archive $\overline{P_g}$ to hold all the best individuals identified during the execution of the method until generation g.

The algorithm itself works as follows. We first create an empty archive $\overline{P_0}$ and starts with a population P_0 of random individuals (line 6). We then assign a fitness value for each individual (line 7), using objectives (x_i, y_i) where x_i is the number of features and y_i is the error rate of individual i[8]. To compute the fitness, we first compute the *strength* S_i of each individual $i | i \in P_g \cup \overline{P_g}$ in the current generation. S_i counts how many individuals are dominated by i. The fitness value of i is then given by Eq. 2.5:

$$fitness(i) = R(i) + D(i) \tag{2.5}$$

where

$$R(i) = \sum_{j \in (P_g \cup \overline{P_g}) \wedge j \succ i} S_j \tag{2.6}$$

and $j \succ i$ means that j dominates i. In other words, $R(i)$ is the sum of the strength of the individuals who dominates i (individuals extracted from the archive or the population in generation g). To break ties, we use the density estimate $D(i)$, calculated according to Eq. 2.7:

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{2.7}$$

where σ_i^k is the distance for the k-th nearest individual in the solution space (x, y) using the K-nearest neighbor algorithm [Silverman, 1986]. k is defined according to Zitzler et al. [2001] and Silverman [1986] as $\sqrt{|P_g| + |\overline{P_g}|}$. The value 2 is used to ensure that $D(i) < 1$ and to keep the denominator greater than zero.

Note that $fitness(i)$ is optimized by minimizing $R(i)$. When $R(i) = 0$, no individual dominates i. Thus, the individuals $\{i | R(i) = 0\}$ are the best solutions, that is, they belong to the Pareto front. From this, it is clear that the smaller is $R(i)$, the closer i is to the Pareto front.

Also note that $D(i)$ is computed to promote a large variety of solutions since the

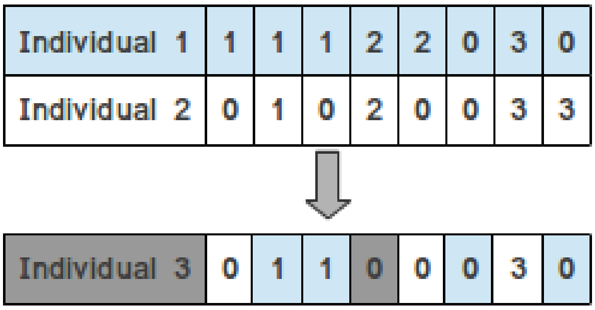
[8] To estimate the error we divide the training set in two random partitions $\mathcal{T}_1$ (with 80% of the instances) and $\mathcal{T}_2$ (with the remaining 20%). The individual is trained using $\mathcal{T}_1$ and its error is estimated using $\mathcal{T}_2$. In next generation, a new hold-out is performed such that a same individual (or its descendants) is never evaluated using the same data. By doing this we expect to minimize overfitting.

value of $D(i)$ decreases as the region of i in the solution space is less dense. By choosing a solution in a more sparse region, more diverse neighbors have a greater chance to be found by genetic operators. For instance, suppose that most of the individuals in a population have objective values (x, y) around $(0.85, 80)$ and that for two particular individuals, i and j, $R(i) = R(j) = 5$. Also suppose that i achieved error 0.85 using 80 features while j achieved a higher error, 0.9, using only 10 features. Since $D(i)$ prefers individuals in sparse regions, the tie will be break with j selected. Thus, j will have a greater chance of surviving to the next generation and evolving (through mutation and crossover). This strategy is also useful to avoid overfitting since it fosters the genetic pool diversity.

Algorithm 1 SPEA2 Genetic Algorithm

```
Require: Population size $N$
Require: Archive size $A$
Require: Number of generations $G$
 1: Let $P_g$ = pop. of individuals $\{i_0, ..., i_n\}$ of generation $g$
 2: Let $\overline{P_g}$ = the best individuals of all generations until $g$
 3: Let $\mathcal{D}_g$ = dominated individuals of $P_{g-1}$ and $\overline{P_{g-1}}$
 4: Let $\mathcal{N}_g$ = non-dominated individuals of $P_{g-1}$ and $\overline{P_{g-1}}$
 5: $\overline{P_0} \leftarrow \emptyset$
 6: Initialize $P_0$ with random individuals
 7: For each individual in $P_0$ assign its fitness value
 8: for $g = 1$ to $G$ do
 9:     Add $\mathcal{N}_g$ to $\overline{P_g}$
10:     if $|\overline{P_g}| > A$ then
11:         $\triangleright$ Truncate $\overline{P_g}$ in order to have $|\overline{P_g}| = A$
12:         $truncate(\overline{P_g})$
13:     else
14:         if $|\overline{P_g}| < A$ then
15:             $k = A - |\overline{P_g}|$
16:             Fill $\overline{P_g}$ with the $k$ best individuals in $\mathcal{D}_g$
17:     Let $R$ = rank of individuals from $\overline{P_g}$ sorted by fitness
18:     while $|P_g| < N$ do
19:         Select two random individuals $i_x$ and $i_y$ from $R$
20:         if $will\_perform\_crossover(p_c)$ then
21:             $new\_ind \leftarrow perform\_crossover(i_x, i_y)$
22:             Add $new\_ind$ in $P_g$
23:         else
24:             Add $i_x$ to $P_g$
25:             if $|P_g| < N$ then
26:                 Add $i_y$ to $P_g$
27:     for all $i \in P_g$ do
28:         if $will\_perform\_mutation(p_m)$ then
29:             $new\_ind \leftarrow mutate(i)$
30:             Replace individual $i$ by $new\_ind$ in $P_g$
31:     Assign fitness values to individuals in $P_g$ and $\overline{P_g}$
```

After initializing population P_0 and assigning fitness values, we create the archive $\overline{P_g}$ (line 9-16). To this end, we add all the nondominated individuals $\mathcal{N}_g$ of the last generation population and archive (i.e., $\mathcal{N}_g = \{i | fitness(i) < 1 \land i \in (P_{g-1} \cup \overline{P_{g-1}})\}$) to $\overline{P_g}$. We then ensure that $\overline{P_g}$ has at most A individuals. To accomplish this, if



Individual 1	1	1	1	2	2	0	3	0
Individual 2	0	1	0	2	0	0	3	3
Individual 3	0	1	1	0	0	0	3	0

Figure 2.3. Example of genetic operations. Individuals are represented by vectors of "genes". Each gene corresponds to a feature, whose value indicates its view. In this figure, individuals 1 and 2 "reproduce", by crossover, into individual 3. As result, each gene i of individual 3 was copied from one of its parents, with equal probability. Note the color of each gene in 3 is the same of its parent from whom it inherited the gene. The exception is the fourth gene that, due to a mutation operation, was replaced by 0.

$|\overline{P_g}| < A$, we fill $\overline{P_g}$ with the best dominated individuals (i.e. those with best fitness and $fitness(i) \geq 1$) from $\overline{P_{g-1}} \cup P_{g-1}$. Otherwise, if $|\overline{P_g}| > A$, we truncate the archive.

To avoid loosing diversity, we truncate the archive (line 12) by removing individuals that are the most similar to the others. More specifically, we remove the individual that has the minimum distance to the i-th nearest individual, starting with $i = 1$. In case of ties, we analyze the $(i + 1)$-th nearest individual. This process continues iteratively until $|\overline{P_g}| = A$.

Once the best individuals of the previous generation are known, we proceed with the creation of a new population P_g as follows. The individuals in $\overline{P_g}$ are ranked according to their fitness (line 17) and while $|P_g| < N$, pairs of individuals are selected (line 19) for crossover. The probability of selection is proportional to the rank positions of the individuals, that is, the lesser is the fitness of the individuals, the larger is the chance of being selected. Once the pair is selected, the probability of crossover application is given by p_c (lines 20-22). After crossover has been applied to the entire population, individuals are randomly selected to mutation with a probability p_m (lines 27-30).

In order to perform a crossover between two individuals i_x and i_y, we iterate over vectors $c_x = \{f_0, f_1, ..., f_m\}$ and $c_y = \{f_0, f_1, ..., f_m\}$ such that, for each position p, we select f_p from c_x or c_y with probability 0.5. Mutation consists of selecting a single position p of individual i and change its value to 0 if it is not 0, or to v otherwise. Figure 2.3 illustrates these operations.

After that, fitness values are calculated for the new individuals (line 31) and a new generation takes place. After G generations, the evolution is interrupted and the non-dominated individuals from $\overline{P_G}$ and P_G are selected. From this pool of individuals, since non-dominated individuals can be alive for many generations, we chose the one with best average error. Note that, using a Genetic Algorithm approach can have its disadvantages as they can have a high computational cost. However, it allows us to analyze a diversity of solutions and to understand better the behavior and importance of each feature and views.

2.4 Related Work

Issues about the quality of collaborative content on the Web have motivated several previous studies. In this section, we review studies organized on (1) features/indicators associated with quality assessment or related topics and general quality assessment in (2) Wikis, (3) Q&A forums, and (4) other domains and applications. We finish this section highlighting the difference of our work to previously proposed studies.

2.4.1 Quality Assessment Features

The need for automatically estimating content quality has motivated many studies on quality indicators. For instance, in the past century, many authors proposed textual features to infer the readability degree of a textual content. That was useful to estimate the age or US grade level necessary to one comprehends a text (cf. Flesch [1948]).

With the web, many authors shifted their attention to web specific indicators. For instance, Alexander and Tate [1999] discuss some quality criteria for web resources such as authority (the editor's reputation), use of hyperlinks, accuracy of information, and coverage. Other studies, not really addressing quality issue problems, also focused on indicators that later would be applied to quality assessment. For instance, Argamon et al. [2003] focus on textual evidence such as using several classes of simple lexical and syntactic text features to analyze the difference between genres when writing formal texts. Similarly, Zheng et al. [2006] use four types of writing-style features (lexical, syntactic, structural, and content-specific features) to identify the authorship in online messages. Fogg et al. [2001] analyze which aspects affect people's perception of credibility. They have shown that the expertise, ease of use and trustworthiness are the aspects that most affect the credibility of a web site. In addition, Zhang et al. [2007] focused their work on expertise finding in online forums.

Wikipedia has motivated the study of many indicators not only for quality assessment but also for other quality related tasks, such as vandalism detection [Potthast et al., 2008; Chin et al., 2010; Gelbukh, 2011] and reputation [Anthony et al., 2005; Bigonha et al., 2010; Wöhner et al., 2011].

Review evidence is explored by Zeng et al. [2006], Wilkinson and Huberman [2007], Adler and de Alfaro [2007], Han and Wang [2011]. Wilkinson and Huberman [2007] discovered that high quality articles usually have a high number of edits, a high number of editors and an intense cooperative behavior. Adler and de Alfaro [2007] proposed a visualization scheme for Wikipedia articles, where the color of the text changes based on the reputation of an editor. This reputation is calculated using the history of their reviews. Similarly, Zeng et al. [2006] use review evidence to assess the trustworthiness of an article.

Network evidence is explored by Korfiatis et al. [2006], where two networks are defined: the *article network* and the *editor network*. In the first case, articles are seen as nodes and the edges are composed by the hyperlinks between them. In the second case, editors are seen as nodes and they are linked if they have reviewed the same article. The authors analyzed network metrics, such as in-degree and centrality, and show that these can help to assess the quality of an article. For example, they can provide a degree of consensus between editors. Furthermore, Benevenuto et al. [2009] used network features such as PageRank [Brin and Page, 1998] and assortativity in order to detect spam in Web pages and online video systems, such as YouTube.

Many studies are focused on specific knowledge domains. For instance, Flekova et al. [2014] study features specifically useful for quality estimating of Wikipedia biographies, while Conti et al. [2014] focus on Wikipedia medical articles.

Note that all these indicators are used to assess different *quality dimensions*. For instance, statistical indicators, such as the number of typos in an article or the number of recent edits, can be used to infer how syntactically correct the article is, i.e., its *correctness*. Review features can infer its *maturity* and *stability* and network features can help to infer *importance* and *reputation*. Given the current state-of-the-art in semantic analysis, semantic dimensions are little explored in literature. An example of study focused on such features is the one by Han et al. [2014] that use natural language processing techniques to identify facts in articles in order to predict its completeness, timeliness and factual accuracy. Their study was, however, restricted to Wikipedia biographies, historical Wikipedia articles, and newspapers.

Many of the indicators previously proposed in literature were carefully reviewed by Stvilia. For instance, Stvilia et al. [2005] propose indicators to assess articles in Wikipedia regarding reputation, completeness, complexity, informativeness, consistency, currency, and volatility. Later, Stvilia et al. [2007] analyzed much previously proposed indicators in the light of a general framework. Different from previous work, however, such framework was proposed as a knowledge resource and guide for developing quality measurement models. As such, it did not enforce any particular method for automatic assessment, consisting of comprehensive typologies of quality identification problems, related activities, and a taxonomy of quality dimensions organized in a systematic way. Their proposal was evaluated using two case studies based on Dublin Core records and Wikipedia. In the case of Wikipedia, an automatic combination strategy was suggested, based on two steps: in the first, the unsupervised step, 19 features were extracted and heuristically combined into seven quality metrics; in the second, the supervised step, these metrics were used as features in a C45 classifier with satisfactory results.

2.4.2 Quality Assessment in Wikis

Given the many indicators previously studied (cf. Section 2.4.1), some authors proposed to combine them into a unique value to represent overall quality, specially on Wikipedia. For instance, Dondio et al. [2006a,b] suggested a methodology to estimate the quality and credibility of articles in Wikipedia. Thus, several pieces of evidence are combined to build an article ranking that tries to capture certain aspects of quality, such as stability, editing quality, and importance. These pieces of evidence are extracted from the article revision history, textual content, and hyperlink structure and combined into a unique final ranking.

Hu et al. [2007] propose to measure the quality of an edit based on the quality of its reviewers. Recursively, the quality of the reviewers is based on the quality of the articles they reviewed. Authors like Cusinato et al. [2009] and Wöhner and Peters [2009] proposed a similar metric which assigns quality scores to both articles and contributors. Similarly, Suzuki and Yoshikawa [2013] proposed an approach of mutually evaluating reviewers and articles which the quality of the reviewers is not calculated by using the text quality. Instead they use their own quality score, claiming that their approach is more resilient to vandalism.

Differently from approaches proposed by Dondio et al. [2006a] and Dondio et al. [2006b], which used simple linear combination methods, a few other efforts were proposed to combine the available evidence using machine learning techniques. As an example, Rassbach et al. [2007] proposed the use of natural language features such as the number of phrases, auxiliary verbs, and the Kincaid readability index [Ressler, 1993], together with a Maximum Entropy Model [Borthwick et al., 1998] to estimate the quality of the articles. In another work, De la Calzada and Dekhtyar [2010] proposed a machine learning approach to estimate the quality of articles regarding two categories: stabilized articles and controversial articles. In Xu and Luo [2011], the authors use textual features and machine learning to predict the quality of an article using lexical clue words and a decision tree. In addition, Han and Wang [2011] use the evolution of the history of reviews to predict the quality of the article.

We have proposed to treat quality estimation as a regression problem [Dalip et al., 2009]. In other words, we estimated the quality of articles in Wikipedia as a grade in a continuous quality scale. To accomplish that, we used a Support Vector Regression method [Drucker et al., 1996; Vapnik, 1995]. Our main contribution in that work was a detailed study of the various sources of evidence and their impact on the prediction of the quality of a Wikipedia article. Furthermore, the proposed method was shown to achieve overall better results than the best approaches previously proposed in literature.

This thesis greatly extends the work of Dalip et al. [2009] by experimenting with two other collaborative digital libraries, with some interesting properties regarding estimation of quality (e.g., different criteria for quality evaluation), in addition to a different, much larger sample of Wikipedia than what was used before. Furthermore, a detailed study of the various sources of evidence and their impact on the prediction of the quality, allowing us to make a more generalized conclusions for the quality assessment of content in Wikis. Furthermore, our proposed approach was shown to achieve overall better results than the best approaches previously proposed in literature.

2.4.3 Quality Assessment in Q&A Forums

To facilitate the automatic assessment of question and answers, several works have been proposed in the literature. These can be classified in three groups according to three distinct objectives: (1) find the best answer; (2) rank the given answers; and (3) assess the quality of the question. Our approach is more related to the works of group 2. All these approaches have in common the fact that they are based on machine learning.

Works in group 1, which address the problem of finding the best answer to a given question, generally follow a straightforward classification strategy. A set of questions, for which the asker has already selected the best answer, is used for training. The answers are represented using a particular set of features and a classifier is applied, to label each answer as “best” or not, according to those features. Studies in this group have suggested the use of features related to expertise [Agichtein et al., 2008; Zhang et al., 2007], content length [Agichtein et al., 2008], grammar errors [Agichtein et al., 2008], question topics [Agichtein et al., 2008], user information [Shah and Pomerantz, 2010], comments [Burel et al., 2012], and vocabulary [Gkotsis et al., 2014]. From these, we highlight the work of Shah and Pomerantz [2010], which proposed nine different features related to the answers content and to user information, to predict the best answer in Yahoo Answers. The authors learned the best answers through a classifier based on Logistic Regression [Cessie and Houwelingen, 1992]. They identified features related to the users answering and asking the question are good indicators of the best answers. Similarly, Burel et al. [2012] proposed new features related to the answer and its follow-up comments. They used an Alternating Decision Tree method [Freund and Mason, 1999] to classify the best answers. As a result, they achieved accuracy levels of 84% to 87% in the samples used. Furthermore, they found out that length features are not correlated with best answers for the datasets used and a feature based on the rating of the answer can be a good predictor of the best answers.

Works in group 2, which address the problem of ranking answers, focus on matching questions to answers, using some sort of similarity measure. Examples of studies in this group are the work of Surdeanu et al. [2008], who used an L2R method with only relevance functions as evidence, Jeon et al. [2006], who proposed a ranking model which takes into account an answer quality estimate, and Suryanto et al. [2009], who explored user expertise in the ranking. This last work deserves a more detailed description since, among those we studied, it achieved the best performance. Suryanto et al. [2009] argue that a user can have different expertise levels for different topics. Thus, they proposed quality-aware methods to rank answers. First, they learn good answers by using a manually annotated corpus, where answers are identified as good or bad. Then, they use this information, combined to relevance features, to calculate an expertise value that is used to rank the answers. Their intuition is that a good answer will be provided by a user that has provided good answers to similar questions in the past. To evaluate their method, they performed a manual annotation of a set of answers regarding their relevance ("relevant" or "not relevant"). By using the expertise based method, along with traditional relevance features, they were able to outperform all the previously described work. In addition, Ponzanelli et al. [2014] divide Stack Overflow votes into quality levels in order to predict the answers quality in the Stack Overflow Q&A Forum but, without ranking them.

A characteristic shared by all the previous methods was the use of discrete quality taxonomies as ground truth ("bad / good", "bad / medium / good", and "best / no best"). Such approach, however, ignores the fact that, among the good answers (and among the bad answers), some are better and some are worse. Furthermore, approaches which uses taxonomies which annotates one best answer per question ignores the fact that there are others good answers besides the one chosen by the asker [Sakai et al., 2011]. To avoid this, Sakai et al. [2011] proposed the use of a continuous scale to evaluate answers in Q&A Forums. This not only allows for a more accurate analysis of the answer rating systems, but also the discovery of what the authors call *hard questions*, i.e., questions that are handled poorly by the answer rating systems. The proposed solution, however, this still requires some expensive manual annotation.

Finally, works in group 3 address the problem of assessing the quality of the question itself. This problem was first addressed by Li et al. [2012], where the authors represented questions using a combination of three features: number of tags, number of answers, and the amount of time necessary for the question to be answered. In a following work, Anderson et al. [2012] highlight the importance of old questions which attract people via search engines. Thus, the authors proposes a method to predict whether a question has already been sufficiently answered and predicting whether it would attract attention in future.

In Q&A Forum domain, our work is closer to those in group 2: ranking answers. However, unlike the previous methods, we do not require explicit quality rating annotations. Instead, we use the number of positive and negative votes (rating) available on a different set of questions as an implicit quality assessment. This assessment can then be used to train an L2R method, which can later be applied on new questions, even if their answers were not voted yet. Furthermore, like Sakai et al. [2011], we use a continuous scale for answer quality. We also improve on previous proposals by studying a new set of topic-based features and textual features which we use to assess the quality of Wikis content.

2.4.4 Quality Assessment in Others Domains and Applications

There are other studies focused in assessing quality in other domains. For instance, Weimer et al. [2007] used textual features in order to predict post quality in a regular forum. In addition, Bethard et al. [2009] presented a method to estimate document quality in educational digital libraries. Since quality can change according to the user's perspective, they defined different dimensions of quality and created an indicator for each dimension. For instance, for the dimension "appropriate pedagogical guide" the used indicators were "contains instructions?" and "identifies the learning objects?". Once these indicators were defined, a Support Vector Machine was trained to classify the library article.

Bendersky et al. [2011] proposed an approach incorporating features of quality of content in web documents to rank search engine results. The authors used 10 features (e.g., number of visible terms, number of terms in the title, links percentage, percentage of the information in table, etc.) and combined them using a Markov Model method. By using several collections, authors shown that quality features improved the retrieval performance of text and link based retrieval methods.

2.4.5 Our Approach

All these methods apply a single learned model for quality prediction, which uses all the available information. In order to improve that, in this work we propose to organize sets of related features into *views*, learn a model for each view, and then combine the quality predictions produced with such views are then combined, by means of meta-learning techniques, into one single quality value.

More specifically, in this thesis we propose a general approach for combining statistical indicators to assess content quality in collections created collaboratively using a fully supervised approach. Moreover, we show that we could apply this approach in Wiki and Q&A Forum domain with gains over our baselines. In particular, we apply it to six datasets, three from each domain, to perform an in-depth original analysis of view performance and correlation between views. This study has provided new insights on the impact of different views (and features within views) in the final result, more specifically, on how correlations among views can impact on the accuracy of the final quality estimation. After that, using the SPEA2 algorithm described in Section 2.3, we are able to maintain an error rate comparable to the original approach, with a lesser computational cost. In addition, we also provide a study of the importance of features and views after feature selection.

To accomplish our multi-view approach, we use a meta-learning technique based on stacking [Wolpert, 1992]. Traditionally, stacking techniques use a meta-classifier to learn the relation between the output of distinct learning algorithms and the target class. In our case, instead of using models generated by distinct algorithms, we will use models generated from the distinct views. In this sense, our proposed technique is slightly different from the stacking method as originally proposed. This approach was already successfully used in other domain such as relationship extraction [Zhou et al., 2009].

In our evaluation, the baseline for the Wiki domain is the SVR regressor proposed by Dalip et al. [2011a] and described in Section 3.3.1. As far as we know, this is the best non-multi-view approach proposed for this domain. In this thesis, we refer to this method as SVR. For the Q&A Forum domain, we use two baselines. The first is a traditional Learning to Rank approach based on SVM-rank, similarly to Pal and Konstan [2010]. From now on we refer to this method as SVM-RANK. The second baseline is the method proposed by Suryanto et al. [2009]. This method was also used as baseline by us and, as far as we know, this is the best method previously proposed in the literature not based on the use of ensembles. The intuition behind this method is that the expertise of the users is not the same for all topics and good answers will be given by users who have provided good answers to similar questions in the past. Thus, to predict the rank of answer a given by user u to question q, we need to combine the estimated quality of a using the quality of other answers of u given to questions similar to q, weighted by the respective similarities. Note that the other answers of u capture the expertise of u in the topic of a. To learn the answer quality, Suryanto et al. [2009] manually annotate a set of answers as good or bad. This information was then combined with relevance features in order to calculate a quality value used to rank the answers. From now on, we refer to the method of Suryanto et al. [2009] as EX_QD.

Chapter 3

Proposed Approach

In this chapter, we present our proposal to infer the quality of web collaborative items. We call *collaborative item* (CI) any item (e.g., a document or an answer to a question) that is open for edition on the Web. We describe quality indicators and how they relate to quality dimensions, and the sources they are extracted. We then describe our method to combine the assessments derived from each quality view.

3.1 Quality Dimensions, Indicators and Sources

Given the quality dimensions in Section 2.1, we now discuss its importance in the domains we study in this thesis and which *indicators* could estimate them. An indicator[1] is a statistic value containing a measurement that is probably correlated with a quality dimension. For instance, the number of characters in the text *indicates* how conciseness is a CI.

In the following, we present indicators grouped by the sources from which they were extracted, that is, *content structure*, *text content*, *content relevance*, *edit history*, *CI graph*, *user/editor information*, and *user/editor graph.* Figure 3.1 shows an overview of how dimensions, indicators and sources are related to each other. In Table 3.1 we show the relationships between all dimensions and sources, while the specific indicators are better explained in the Section 3.2.

Among these dimensions, the most enforced by the website publishing guides are *correctness* (no misspellings), *appropriateness* (use of appropriate language and content), *conciseness*, *factual accuracy*, *reliability*, *organization*, *clarity* and *understandability.* Some dimensions are emphasized according to the policies enforced by a specific

[1] In this text, we use the terms *indicator* and *feature* interchangeably.

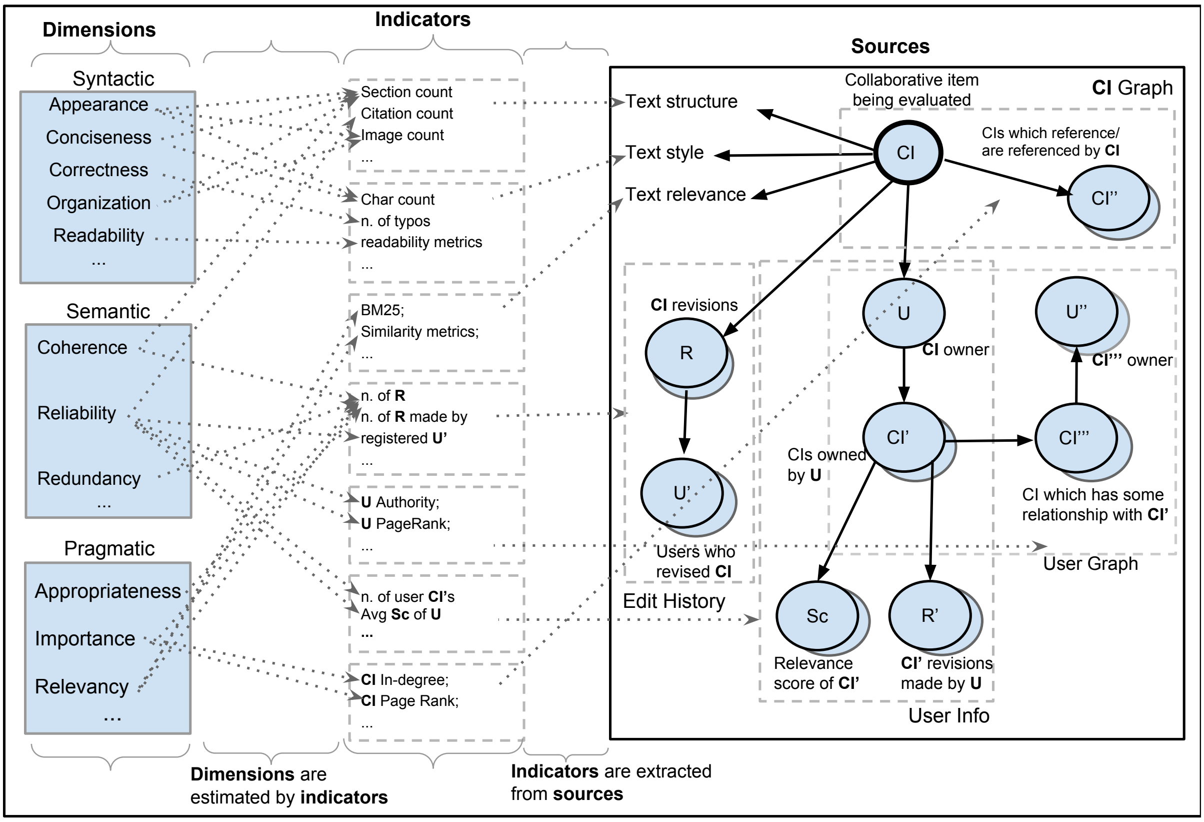


Figure 3.1. Sources, indicators and dimensions. *Dimensions*, grouped into three levels (syntactic, semantic and pragmatic) are estimated by *indicators*, which are extracted from *sources*.

service. For instance, Wikipedia demands a *neutral point of view* while in Q&A forums *relevance* is more important, since the goal is to satisfy a clear information need, described by means of a question. Also, since articles in Wikipedia are subject to many editions, other important quality dimensions are *maturity*, *stability*, and *completeness*, i.e., the content should not change and topics should be covered in depth, breadth and scope. In Q&A forums, *sufficiency* is preferable, i.e., the question should be answered. Q&A forums also stress the need for pointers to additional material and sources so that the interested user can obtain more information.

As for the indicators, it is clear that most of the effort to assess quality in collaborative repositories has focused on the syntactic and pragmatic levels. Indicators for these levels are normally easy to calculate in contrast with semantic indicators that often require the use of expensive natural language processing techniques. Indicators obtained from content structure can be used to assess *appearance*, *clarity*, and *organization* by means of the distribution of sections, images, links, and citations. They can be used indirectly to determine *reliability*, *reputation* and *validity*, which are related to

Quality Dimensions	Sources of Indicators
	Syntactic
Appearance	Text Content, Content Structure, Edit History
Clarity	Text Content, Content Structure, Edit History, CI Graph
Conciseness	Text Content, Content Structure
Consistency	Text Content, Edit History
Correctness	Text Content, Edit History
Level-of-detail	Text Content, Content Structure, Edit History, CI Graph
Organization	Text Content, Content Structure, Edit History, CI Graph
Readability	Text Content, Edit History
	Semantic
Ambiguity	-
Coherence	Edit History
Factual Accuracy	Edit History
Informativeness	Text Content, CI Graph
Meaningfulness	-
Opinative or factual	-
Redundancy	Edit History
Reliability	Edit History, Content Structure, User Info, User Graph, CI Graph
Understandability	Content Structure History, Text Content, CI Graph
Validity	Content Structure, Edit History
	Pragmatic
Appropriateness	Edit History
Completeness	Text Content, Content Structure, Edit History, CI Graph
Engagement	-
Importance	Edit History, CI Graph
Maturity and Stability	Text Content, Edit History, CI Graph
Neutral Point of View	Edit History
Relevancy	Edit History, Content Relevance, CI Graph
Reputation	Content Structure, Edit History, User Graph, UsI, CI Graph
Sufficiency	Text Content, Edit History
Timeliness	Edit History, Text Content, Edit History
Usefulness	CI Graph

Table 3.1. Quality dimensions and sources of the indicators used to assess them.

citations, as well as *conciseness*, *level-of-detail*, and *completeness*, which are related to how many structural elements can be found in the CI.

Indicators extracted from the textual content try to capture length, author writing style (by means of word usage), *readability* (using classical lexical metrics designed to estimate the age/US grade level necessary to comprehend a text), and *relevance* (using the similarity between questions and queries, in the case of Q&A forums). Length, along with other indicators, can be used to assess *appearance*, *conciseness*, *level of detail*, *completeness*, *maturity*, and *sufficiency*. Writing style can be used to assess *clarity*, *conciseness* and *correctness*. Lexical *readability* can be used to assess semantic *understandability.*

Indicators extracted from the edit history are correlated with many quality dimensions. An article that was much reviewed is probably clear, organized, up to date, and complete. In general, these indicators are very useful to assess the *maturity* and *stability* of the content.

Finally, indicators extracted from the user/editor data and user/editor graph correlate to *reliability* and *reputation.* These indicators capture, in general, experience and expertise of users/editors. For example, expertise can be inferred by the ExpertiseRank [Zhang et al., 2007] metric. Similarly, indicators extracted from the CI citation graph are also used to infer *importance* and *validity.*

These indicators need to be divided into views. First, we define view as a partition of a set of indicators where each partition is composed by indicators that are naturally seen as a group. In other words, views are built such that they do not share indicators. This enforces the creation of independent views as much as possible, which is desirable since we intend to use them to design independent experts for an ensemble classifier. Furthermore, as defined by Blum and Mitchell [1998], we can apply multiview approach when the groups can be naturally divided into views and the group of indicators forming the view needs to be enough to the prediction task, in other words, they can independently provide a good result.

Note that neither the dimensions nor the indicators are independent, which leads to mutual reinforcement (e.g., semantic validity versus pragmatic reputation) and adoption tradeoffs (e.g., a neutral point of view can preclude a balanced content coverage according to importance). As a consequence, as we can see in Figure 3.1, a same indicator can be shared by many dimensions. Because of that, we do not adopt dimensions as views[2]. Thus, in this thesis, the main criterion used to group indicators in views was the source of the indicator. Within each source, we grouped the indicators into dimensions only when they were clearly independent of the others. We also separate text length as a view due to its high correlation with quality [Blumenstock, 2008]. The set of views we use in this thesis is summarized in Table 3.2.

3.2 Quality Indicators

In this thesis, we use a large set of indicators to infer quality in Wikis and Q&A forums. These indicators are summarized in Table 3.3 and 3.4, where they were separated in

[2] We have tried different strategies to group the data, including grouping them according to its quality dimensions. We observed it is hard to get independent sets using quality dimensions because many indicators correlate to multiple dimensions.

Table 3.2. Views used in this work. Columns D stands for "domain" which can be Q&A (Q), Encyclopedia (E), or both (B).

View	Source	D	Dimensions
Length	Text Content	B	Conciseness, Level-of-detail, Completeness, and Sufficiency
Readability	Text Content	B	Readability
Relevance	Text Content	Q	Relevance
Style	Text Content	B	Many syntactic and some semantic and pragmatic
Structure	Content Structure	B	Many syntactic and some semantic and pragmatic
Edit History	Edit History	B	Many syntactic, semantic and pragmatic
User	User	Q	Reliability and Reputation
User graph	User Graph	Q	Reliability and Reputation
Article graph	CI Graph	E	Many pragmatic and some syntactic and semantic

textual and non-textual indicators. In the following sections, we provide more detailed descriptions of the indicators from each domain.

3.2.1 Indicators Extracted from Wikis

As previously mentioned, the views used in this domain are (1) Structure, (2) Readability, (3) Length, (4) Style, (5) Edit History, and (6) Article Graph. Note that in this domain we did not use indicators from editor information and editor graph as Wiki articles do not have an owner.

Structure features indicate how well the article is organized. According to Wiki quality standards[3,4] a good article must be organized such that it is clear, visually adequate, and provides the necessary references and pointers to additional material. Thus, we use features derived from the article structure in an attempt to describe its section organization, and its use and distribution of images, links, and citations. To accomplish this we have features such as the citation, image and section count, average number of citations per section, etc.

As we can see in Table 3.3, Length features are indicators of the article size. The general intuition behind them is that a mature and good quality text is probably neither too short, which could indicate an incomplete topic coverage, nor excessively long, which could indicate verbose content. Further, in Wikis, *stub articles* (draft quality) are expected to be short, which reinforces the correlation between length and quality.

Style features are intended to capture the way the authors write the articles through their word usage. The intuition behind them is that good articles should

[3] http://starwars.wikia.com/wiki/Wookieepedia:Featured_article_nominations

[4] http://starwars.wikia.com/wiki/Wookieepedia:Good_article_nominations

Table 3.3. Indicators extracted from text content and structure. Column "D" stands for domain, that is, Q&A (Q), encyclopedia (E) or both (B). "#p" stands for *number of phrases*. Features marked with "*" were first used in Q&A Forum domain.

Length					
Indicator	Description	D.	Indicator	Description	D.
tl-charcnt	Character count	B	*tl-wordcnt*	Word count	B
tl-phrcnt	Phrase Count	B			
Readability					
Indicator	Description	D.	Indicator	Description	D.
tr-ari	Automated Readability Index* [Smith and Senter, 1967]	B	*tr-liau*	Coleman-Liau* [Coleman and Liau, 1975]	B
tr-flesh	Flesch reading ease* [Flesch, 1948]	B	*tr-lix*	Läsbarhets index* [Björnsson, 1968]	B
tr-fog	Gunning Fog Index [Gunning, 1952]	B	*tr-smog*	Smog-Grading [McLaughlin, 1969]	B
tr-kincaid	Flesch-Kincaid [Ressler, 1993]	B			
Relevance					
Indicator	Description	D.	Indicator	Description	D.
tm-aspan$_S$	Largest distance between two words bigrams that appear in answer and questions	Q	*tm-nwver*$_S$	Number of new verbs in the answer which did not appear on the question	Q
tm-bm25$_{S,T}$	BM25 ranking function [Robertson and Walker, 1994] for each representation	Q	*tm-phmch*$_{S,T}$	Number of sentences shared by question and answer	Q
tm-nwadj$_S$	Number of new adjectives in the answer which did not appear on the question	Q	*tm-wmtch*$_{S,T}$	Number of words shared by question and answered	Q
tm-nwnou$_S$	Number of new nouns in the answer which did not appear on the question	Q	*tm-worder*$_S$	Number of words shared by question and answer	Q
Structure					
Indicator	Description	D.	Indicator	Description	D.
ts-abslen	Length of abstract	E	*ts-maxcod*	Maximum code length*	Q
ts-avsecl	Average section length*	B	*ts-mincod*	Minimum code length*	Q
ts-avgcod	Average code length*	Q	*ts-minsecl*	Length of shortest section*	B
ts-avparl	Average paragraph length	E	*ts-mnquot*	Minimum quoted text length*	Q
ts-avquot	Average quoted text length*	Q	*ts-mxquot*	Maximum quoted text length*	Q
ts-avsubps	Average subsections per sections	E	*ts-maxsecl*	Length of largest section*	B
ts-boldit	Italic plus Bold tag count*	Q	*ts-parcnt*	Paragraph Count*	Q
ts-cite	n. of citations (references)	E	*ts-quotes*	n. of quoted blocks*	Q
ts-citplen	n. of citations divided by text length	E	*ts-secs*	Section Count*	B
ts-citpsec	ratio between n. of citations and sections	E	*ts-subsec*	Sub-section Count*	B
ts-codes	n. of code snippets*	Q	*ts-stdsecl*	Section length standard deviation*	B
ts-avsubs	Image count*	Q	*ts-ssssec*	Sub-sub-section (HTML H3 tag) Count*	Q
ts-imgps	n. of images per section	E	*ts-stdcod*	Code length standard deviation*	Q
ts-inlink	n. of links to other qst./ans. in the forum*	Q	*ts-stdquot*	Quoted text length standard deviation*	Q
ts-list	n. of lists*	Q	*ts-usrref*	n. of interactions with other forum users*	Q
ts-listit	n. of list items in the text*	Q	*ts-xlnks*	n. of links to external sources*	B
ts-lnkpl	n. of links per text length	E	*ts-xlnkps*	n. of external links per section	E
Style					
Indicator	Description	D.	Indicator	Description	D.
ty-auxverb	n. of auxiliary verbs*	B	*ty-plgphr*	%p where (length - avg. length) $\geq$ 10 words*	B
ty-caperr	n. of capitalization errors	Q	*ty-prepo*	n. of prepositions*	B
ty-capwrd	n. of words capitalized	Q	*ty-prono*	n. of pronouns*	B
ty-conj	n. of words that are conjunctions*	B	*ty-psmphr*	%p where (avg. length - length) $\geq$ 5 words*	B
ty-dotcnt	Punctuation count	Q	*ty-questn*	n. of questions*	B
ty-dotden	Punctuation density	Q	*ty-sartic*	#p starting with an article*	B
ty-infnois	Information to noise	Q	*ty-sconj*	#p starting with a conjunction*	B
ty-klddis	KLD(Wikipedia discussion pages)	Q	*ty-sintp*	#p starting with an interrogative pronoun*	B
ty-kldqa	KLD(good answers)	Q	*ty-spaden*	Space density (n. of spaces / answer length)	Q
ty-kldtag	KLD(good answers of same category)*	Q	*ty-sprepo*	#p starting with a preposition*	B
ty-kldwiki	KLD(Wikipedia pages classified as "Good")	Q	*ty-sprono*	#p starting with a pronoun*	B
ty-lgphra	Size of the largest phrase*	B	*ty-ssubcnj*	#p starting w/ a subordinating conjunction*	B
ty-nomina	n. of nominalizations*	B	*ty-tobe*	n. of uses of verb "to be"*	B
ty-notwn	n. of words not in WordNet	Q	*ty-typo*	n. of typos	Q
ty-passive	n. of passive voice sentences*	B	*ty-wrenpy*	Entropy of the text word sizes	Q

Table 3.4. Indicators extracted from sources other than text. Column "D" stands for domain, that is, Q&A (Q), encyclopedia (E) or both (B). "#sug" stands for *number of suggested edits*; "#ans" stands for *number of answers.*

Edit History

Indicator	Description	D.	Indicator	Description	D.
r-3month	proportion of *r-rcount* in last 3 months	E	*r-probrev*	ProbReview	E
r-aaped	#sug approved by the answer author*	Q	*r-qaped*	#sug approved by the asker*	Q
r-activeu	reviews by top-5% most active reviewers	E	*r-qrejed*	#sug rejected by the asker*	Q
r-age	Article age	E	*r-qsuged*	#sug to the answer*	Q
r-ageprev	ratio between article age and n. of reviews	E	*r-queage*	Question age*	Q
r-anonym	n. of reviews made by anonymous users	E	*r-queans*	#ans posted to the question	Q
r-ansage	Answer age	Q	*r-rcount*	Review count*	B
r-ansbef	#ans posted before this answer*	Q	*r-reguser*	n. of reviews made by registered users	E
r-arejed	#sug rejected by the answer author	Q	*r-revpday*	percentage of reviews per day	E
r-asuged	#sug to the question*	Q	*r-rperusr*	ratio between *r-rcount* and n. of reviewers	E
r-avedusr	Average n. of edits per user*	Q	*r-stdpusr*	Standard deviation of edits per user*	Q
r-comans	n. of comments posted to the answer	Q	*r-stdrevu*	std. dev of *r-rperusr*	E
r-comque	n. of comments posted to the question	Q	*r-uniqusr*	n. of users who suggested edits to ans./qst.*	Q
r-discuss	n. of posts on the article's discussion page	E	*r-usrcom*	n. of users who commented the answer*	Q
r-modline	% of lines of curr. version $\neq$ from reference	E	*r-usredt*	n. of users who edit the answer	Q
r-occasion	reviews by reviewers with less than 4 edits	E			

Article Graph

Indicator	Description	D.	Indicator	Description	D.
n-assortii	Assortativity In-In	E	*n-linkcnt*	n. of links to articles (even if not written)	E
n-assortio	Assortativity In-Out	E	*n-odegree*	n. of links to other articles	E
n-assortoi	Assortativity Out-In	E	*n-pgrank*	Pagerank value of an article	E
n-assortoo	Assortativity Out-Out	E	*n-reciproc*	Reciprocity	E
n-cluster	Clustering coefficient	E	*n-translat*	n. of article versions in other languages	E
n-idegree	n. of citations of an article from other ones	E			

User

Indicator	Description	D.	Indicator	Description	D.
u-anpytag	Answers entropy*	Q	*u-mrkta*	min rank position in R_{racat}*	Q
u-answrs	n. of posted answers	Q	*u-mrktq*	min rank position in R_{rqcat}*	Q
u-apsuged	n. of suggested edits approved*	Q	*u-mxatag*	min n. of answers posted in the $\mathcal{T}$ categories*	Q
u-arateat	avg rating received in the $\mathcal{T}$ categories*	Q	*u-mxatag*	max n. of ans. posted in the $\mathcal{T}$ categories*	Q
u-arateqt	avg rating received in the $\mathcal{T}$ categories*	Q	*u-mxcoma*	max n. of comments per answer*	Q
u-arkta	avg rank position in R_{racat}*	Q	*u-mxcomq*	max n. of comments per question*	Q
u-arktq	avg rank position in R_{rqcat}*	Q	*u-mxqtag*	max n. of qsts. posted in the $\mathcal{T}$ categories*	Q
u-avansq	avg answers posted per question*	Q	*u-mxratag*	min rank position in R_{acat}*	Q
u-avatag	avg n. of ans. posted in the $\mathcal{T}$ categories	Q	*u-mxrqtag*	max rank position in R_{qcat}*	Q
u-avcoma	avg n. of comments per answer*	Q	*u-prtp3an*	*u-top3an* / n. of categories user asked	Q
u-avcomq	avg n. of comments per question*	Q	*u-prtp3qu*	*u-top3qu* / n. of categories user answered	Q
u-avqtag	avg n. of qsts. posted in the $\mathcal{T}$ categories*	Q	*u-quests*	n. of posted questions	Q
u-avratag	avg rank position in R_{acat}*	Q	*u-rateans*	Total rating received by answering qsts.	Q
u-avrqtag	avg rank position in R_{qcat}*	Q	*u-rateque*	Total rating received by asking questions	Q
u-badges	n. of merit badges*	Q	*u-rjsuged*	n. of suggested edits rejected*	Q
u-commnt	n. of comments posted to ans. and qsts.*	Q	*u-rkans*	Rank position in $R_{answers}$*	Q
u-daycrt	n. of days since register*	Q	*u-rkqust*	Rank position in $R_{questions}$*	Q
u-edits	n. of edits made on answers*	Q	*u-rratea*	Rank position in R_{rans}*	Q
u-enpytag	Questions entropy*	Q	*u-rrateq*	Rank position in R_{rask}*	Q
u-enpytag	Questions and answers entropy	Q	*u-solvqu*	n. of posted questions already solved	Q
u-mrateqt	min rating received in the $\mathcal{T}$ categories	Q	*u-srateat*	Tot. rating got by answering qsts. in $\mathcal{T}$ cats.	Q
u-lastac	n. of days since last access*	Q	*u-srateqt*	Tot. rating got by asking qsts. in $\mathcal{T}$ cats.	Q
u-maxansq	max answers posted per question*	Q	*u-sugedt*	n. of suggested edits*	Q
u-minansq	min answers posted per question*	Q	*u-top3an*	n. of categories user is a top-3 asker	Q
u-mncoma	min n. of comments per answer*	Q	*u-top3qu*	n. of categories user is a top-3 answerer	Q
u-mncomq	min n. of comments per question*	Q	*u-xrateat*	max rating received in the $\mathcal{T}$ categories*	Q
u-mnqtag	min n. of qsts. posted in the $\mathcal{T}$ categories*	Q	*u-xrateqt*	max rating received in the $\mathcal{T}$ categories*	Q
u-mnratag	max rank position in R_{acat}*	Q	*u-xrkta*	max rank position in R_{racat}*	Q
u-mnrqtag	min rank position in R_{qcat}*	Q	*u-xrktq*	max rank position in R_{rqcat}*	Q
u-mrateat	min rating received in the $\mathcal{T}$ categories*	Q			

User Graph

Indicator	Description	D.	Indicator	Description	D.
ug-auth	User Authority	Q	*ug-hub*	User Hits	Q
ug-exprank	User Expertise Rank	Q	*ug-prank*	User Page Rank	Q

Table 3.5. Terms to compute style features.

Feature	Terms
ty-auxverb	will, shall, cannot, may, need to, would, should, could, might, must, ought, ought to, can't, can
ty-prono	I, me, we, us, you, he, him, she, her, it, they, them, thou, thee, ye, myself, yourself, himself, herself, itself, ourselves, yourselves, themselves, oneself, my, mine, his, hers, yours, ours, theirs, its, our, that, their, these, this, those, your
ty-conj, *ty-sconj*	and, but, or, yet, nor
ty-nomina	suffixes tion, ment, ence, ance
ty-prepo, *ty-sprepo*	aboard, about, above, according to, across from, after, against, alongside, alongside of, along with, amid, among, apart from, around, aside from, at, away from, back of, because of, before, behind, below, beneath, beside, besides, between, beyond, but, by means of, concerning, considering, despite, down, down from, during, except, except for, excepting for, from among, from between, from under, in addition to, in behalf of, in front of, in place of, in regard to, inside of, inside, in spite of, instead of, into, like, near to, off, on account of, on behalf of, onto, on top of, on, opposite, out of, out, outside, outside of, over to, over, owing to, past, prior to, regarding, round about, round, since, subsequent to, together, with, throughout, through, till, toward, under, underneath, until, unto, up, up to, upon, with, within, without, across, along, by, of, in, to, near, of, from
ty-tobe	be, being, was, were, been, are, is
ty-sartic	the, a, an
ty-ssubcnj	after, because, lest, till, 'til, although, before, now that, unless, as, even if, provided that, provided, until, as if, even though, since, as long as, so that, whenever, as much as, if, than, as soon as, inasmuch, in order that, though, while
ty-sintp	why, who, what, whom, when, where, how

present some distinguishable characteristics related to word usage, such as short sentences. To compute them and the Readability indicators described in next Section, we use the Style and Diction software[5]. Terms used to compute some Style features are shown in Table 3.5.

Readability features, first used by Rassbach et al. [2007], are intended to estimate the age or US grade level necessary to comprehend a text. The intuition behind these features is that good articles should be well written, understandable, and free of unnecessary complexity. To accomplish this, these features use texts properties such as number of words (or characters) in order to compute the readability. The equations to compute Readability features are the following:

[5] `http://www.gnu.org/software/diction/`

$$\textit{tr-ari} = 4.71\frac{\textit{tl-charcnt}}{\textit{tl-wordcnt}} + 0.5\frac{\textit{tl-wordcnt}}{\textit{tl-phrcnt}} - 21.43 \tag{3.1}$$

$$\textit{tr-liau} = 5.89\frac{\textit{tl-charcnt}}{\textit{tl-wordcnt}} - 0.3wf - 15.48 \tag{3.2}$$

$$\textit{tr-flesh} = 206.835 - 1.015\frac{\textit{tl-wordcnt}}{\textit{tl-phrcnt}} - 84.6\frac{\textit{syllables}}{\textit{tl-wordcnt}} \tag{3.3}$$

$$\textit{tr-kincaid} = 0.39\frac{\textit{tl-wordcnt}}{\textit{tl-phrcnt}} + 11.8\frac{\textit{syllables}}{\textit{tl-wordcnt}} - 15.59 \tag{3.4}$$

$$\textit{tr-fog} = 0.4(\frac{\textit{tl-wordcnt}}{\textit{tl-phrcnt}} + 100\frac{complexwords}{\textit{tl-wordcnt}}) \tag{3.5}$$

$$\textit{tr-lix} = \frac{\textit{tl-wordcnt}}{\textit{tl-phrcnt}} + 100\frac{complexwords}{\textit{tl-wordcnt}} \tag{3.6}$$

$$\textit{tr-smog} = 3 + \sqrt{polysyllables} \tag{3.7}$$

where wf stands for the number of sentences in a fragment of 100 words, *syllables* is the average number of syllables per word, *complexwords* is the number of words with three or more syllables.

The other sets of indicators used in this domain comprise Edit History and Article Graph, shown in Table 3.4. Edit History indicators have been mainly used to estimate the maturity level of the content. In general, a content that received many edits has likely improved over time. The Graph indicators are those extracted from the links between articles (citations). The main motivation for using them is that citations between articles can provide evidence about their importance. To calculate Article Graph indicators, the collection is seen as a graph, where nodes are articles and edges are the citations between them.

Most of the Article Graph features attempt to capture the importance of the pages. For instance, Pagerank (*n-pgrank*) states that the importance of an article p is proportional to the importance and quantity of articles that point to p. Metrics *n-idegree* and *n-odegree* correspond to in-degree and out-degree of the page. Reciprocity (*n-reciproc*) is the ratio between the number of articles that cite article p and the number of articles that p cites among the ones that cite p. Note that high reciprocity indicates related topics. Given that the k nearest neighbors of an article p are all the nodes whose distance to p is at most k edges, the Clustering Coefficient (*n-cluster*) is the ratio between the number of notes in the set of the k nearest neighbors and the maximum number of edges between p and all their k nearest neighbors. Finally, given the neighborhood of an article p, assortativity metrics estimate the linkage similarity between p and its neighbors (regarding in-degree×in-degree, in-degree×out-degree, out-degree×in-degree, and out-degree×out-degree).

Regarding the cost of creating each feature in Wiki domain, textual features are those with the lowest cost as they need just some simple text parsing. Most of the Edit History features are not demanding to obtain. The only exception is the ProbReview (*r-probrev*) which we need to do a text comparison between all article revisions. Article Graph features have a higher cost when compared to textual and Edit History features (except *r-probrev*) as we need to create the graph for the whole collection in order to compute these features. Except for out-degree which we can get just using the article text.

Overall, we exploit a set of 68 quality indicators in this domain.

3.2.2 Indicators Extracted from Q&A Forums

The views in Q&A Forums domain are (1) Structure, (2) Readability, (3) Length, (4) Style, and (5) Relevance, (6) Edit History, (7) User, and (8) User Graph. Note that, although an answer can cite another, this is so uncommon that we do not consider the citation graph as a source of indicators in this domain. In Q&A Forums, Edit History includes indicators related to the answer revisions and comments. Unlike Wikis, a user cannot directly edit answers of other users (at least in forums from Stack Exchange). They can suggest a modification that is committed only if the answer owner approves it. The user graph consists of a graph where nodes represent users (answer owners) and edges represent user interactions (answering). From this graph, first proposed by Zhang et al. [2007], it is possible to derive metrics related to the expertise and reputation of the users. Also note that, in Q&A Forums, Relevance is an important view since the answer has to be relevant to the question.

As we can see in Table 3.3, the indicators used in the Readability and Length views are exactly the same of the Wiki domain. For the Q&A Forums domain, Length features and the Readability features *tr-flesh*, *tr-fog* and *tr-smog* were first used by Agichtein et al. [2008]. We now describe the differences for the remaining views. Relevance indicators, first proposed in this domain by Surdeanu et al. [2008], try to capture the similarities between the answer and the question. These are useful to identify answers not related to the question, normally using metrics developed in Information Retrieval. As each question has two sections, title and body, two indicators are associated with each metric: (1) matching between question title and answer body and (2) matching between question and answer body. This is denoted by subscription S in the name, which assumes values b or t. For example, *tm-worder*$_b$ considers the words present in question and answer bodies, while *tm-worder*$_t$ takes into account just question title words and answer body.

To compute most of these indicators, two preprocessing tasks were performed: stop-word removal and stemming. The content was represented using bags of terms, where terms could be words, part-of-speech (POS) tags, bi-grams, syntactic dependencies[6] and generalizations. A generalization corresponds to the transformation of each term into its corresponding WordNet supersense, i.e., a category that can be assigned to nouns and verbs (e.g., "dog" is generalized to "animal", a person name is generalized to "person", a verb such as "swash" is generalized to "verb-motion"). A tagger[7] was used to extract POS tags and generalize words. More specifically, features *tm-bm25*$_{S,T}$, *tm-wmtch*$_{S,T}$, and *tm-wmtch*$_{S,T}$ used stemming and the following text representations: words ($T = w$), bi-grams ($T = b$), dependencies ($T = d$), generalized bi-grams ($T = bg$), and generalized dependencies ($T = dg$). For instance, *tm-phmch*$_{b,b}$ is the number of sentence bigrams shared by question and answer. Features *tm-aspan*$_S$ and *tm-worder*$_S$ used stemming and a content representation based on words. Finally, features *tm-nwadj*$_S$, *tm-nwnou*$_S$, and *tm-nwver*$_S$ used a content representation based on POS tags.

Answers and articles have a different structure. As a result, many indicators used in the Q&A Forums domain do not appear in the Wiki domain. For instance, Stack Overflow uses specific tags to allow the placement of program code, which is captured by indicators such as *ts-avgcod*, *ts-maxcod*, *ts-mincod*, *ts-stdcod*. As observed in the Wiki domain, many indicators are related to image and external link count (*ts-xlnks*, *ts-avsubs*), as well as section count and its distribution (*ts-secs*,*ts-subsec*, *ts-avsecl*, *ts-avsecl*, *ts-maxsecl*,*ts-minsecl*, and *ts-stdsecl*). Whereas external link count was first suggested by Shah and Pomerantz [2010], the other Structure features were first proposed by us in this domain.

Unlike the Wiki domain, Style indicators are much more used in Q&A Forums domains, since two good answers can be very different from each other. Because of this, we use more Style indicators in Q&A Forums than in Wikis. Features regarding capitalization (*ty-capwrd*, *ty-caperr*), punctuation and space (*ty-dotcnt*, *ty-dotden*, *ty-spaden*), size of words (*ty-wrenpy*), typos (*ty-notwn*, *ty-typo*), and vocabulary (*ty-kldqa*, *ty-kldtag*, *ty-klddis*, *ty-kldwiki*) were first proposed in this domain by Agichtein et al. [2008]. Feature *ty-caperr* counts what are usually capitalization errors: the first letter

[6] Dependencies were detected by the tool described in Attardi et al. [2007] and available at `http://sourceforge.net/projects/desr`.

[7] We used a tagger based on Wordnet, described in Ciaramita and Altun [2006] and available at `http://sourceforge.net/projects/supersensetag`.

of the sentence not being capitalized and the capitalization of letters that are not the first of a word. These features assume that an irregular use of capitalization may indicate a bad quality text. Features *ty-dotcnt* and *ty-dotden* try to capture the text quality through the use of punctuation, since an irregular punctuation may also be related to a bad quality text. Feature *ty-infnois*, proposed by Stvilia et al. [2005], measures the proportion of (stemmed) non-stopwords in the text.

We also use some vocabulary features in order to identify typos, similarly to Agichtein et al. [2008]. Feature *ty-notwn* computes the number of words that are not in the English lexical database WordNet[8]. Feature *ty-typo* counts the number of words present in a list of common misspellings, available from Wikipedia[9].

Another group of Style features tries to infer the difference between the language model used in the answer and other language models that can be seen as good references. First used in Agichtein et al. [2008], the idea behind them is that an answer is more likely to be written in an inadequate manner if its generating language model is much different from language models which generate good answers. Thus, the feature *ty-kldqa* compares the language model of the answer to the language model of a group of answers considered good (i.e., the top 100 answers according to their rating, obtained from a sample of Stack Overflow different from the one we use for evaluation in Section 4.1.1). Feature *ty-kldtag* is similar but using only answers in the same categories of the answer being assessed. Categories of one answer is defined by the tags which one answer can have. With the same goal, we created a sample of 100 articles, classified as Feature Articles according to the Wikipedia quality taxonomy[10], and its discussion pages. We used this sample to compare the answer language model to the language models of the Wikipedia articles and the discussion pages, which resulted in features *ty-kldwiki* and *ty-klddis*.

Edit History indicators are also different in the Q&A Forums domain since answers and queries have owners. In practice, we noted that Wiki articles are more likely to be edited over time than Q&A Forums answers. On the other hand, as in Wikis, users in Q&A Forums are encouraged to fix mistakes, include examples and further reading sources, etc. Thus, such indicators are useful to estimate how much effort was invested in an answer. Besides the features already used in Wiki domain (*r-rcount*, *r-ansage*, *r-queage*, *r-stdpusr*, and *r-avedusr*), some additional indicators were extracted in Q&A Forums. For instance, in Stack Exchange forums, a user can comment questions and answers and suggest edits to the author of an answer, who can accept them

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

[9] `http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings`

[10] `http://en.wikipedia.org/wiki/Wikipedia:ASSESS`

or not. From such comments we extracted the features *r-qsuged*, *r-asuged*, *r-uniqusr*, *r-qaped*, *r-qrejed*, *r-aaped*, and *r-arejed*. Additionally, general information about the comments, such as *r-comans*, *r-comque*, and *r-usrcom* are good indicators of community engagement. We also derived features that capture the question history by means of its answers. These are *r-ansbef* and *r-queans*. These features are important since they can indicate controversial topics and questions that are hard to answer. As mentioned in [Chua and Banerjee, 2013], hard questions tend to have more answers and the quality of answers tend to improve over time. Feature *r-ansage* was proposed by Burel et al. [2012], *r-comque* and *r-comans* by Shah and Pomerantz [2010] and *r-queans* by Agichtein et al. [2008]. The others were proposed by us in this domain.

The intuition behind User features is to infer the quality of the answer by examining the expertise of its owner. To accomplish this, we extract indicators related to the user profile or its behavior, captured from events such as (1) post of questions and answers; (2) suggestion of edits in questions and answers; (3) post of comments to questions and answers; and (4) gain of merit ratings and badges for questions and answers.

Although most of the user indicators are self-explanatory, some require a more detailed description. Note, in Table 3.4, that we refer to a question for which the best answer was already selected as *solved question*. As the user criteria to infer a good answer can differ according to the topic of the question [Kim and Oh, 2009], we propose features to capture *topic expertise*. Thus, for each question, we also store its categories by exploiting the *tags* (e.g., "html", "C++", "database") the users assign to the questions. We refer to the set of categories (tags) of the Q&A pair being predicted as $\mathcal{T}$. Let $Q_{\mathcal{T}}(u)$ be a vector with the number of questions posted by user u to each category in $\mathcal{T}$, $A_{\mathcal{T}}(u)$ be a vector with the number of answers posted by user u to each category in $\mathcal{T}$, and $QA_{\mathcal{T}}(u)$ be a vector with the number of questions and answers posted by user u to each category in $\mathcal{T}$. Indicators Questions Entropy (*u-enpytag*), Answers Entropy (*u-anpytag*), and Questions and Answers (*u-enpytag*) correspond to the entropy calculated over vectors $Q_{\mathcal{T}}(u)$, $A_{\mathcal{T}}(u)$ and $QA_{\mathcal{T}}(u)$, respectively.

There are also User indicators based on user rankings. For instance, given a user u and a list of the users sorted in decreasing order according to the number of answers they posted ($R_{answers}$), indicator *u-rkqust* is simply the rank of u in $R_{answers}$. Users are also ranked according to (a) the number of questions they posted ($R_{questions}$), (b) the number of answers they posted whose categories are in $\mathcal{T}$ (R_{acat}), (c) the number of questions they posted whose categories are in $\mathcal{T}$ (R_{qcat}), (d) the total rating received by asking questions (R_{rask}), (e) the total rating received by answering questions (R_{rans}), (f) the total rating received by asking questions whose categories are in $\mathcal{T}$ (R_{racat}), and

(g) the total rating received by answering questions whose categories are in $\mathcal{T}$ (R_{rqcat}). Features *u-avansq*, *u-answrs*, *u-quests*and *u-solvqu* were proposed by Agichtein et al. [2008]. Burel et al. [2012] proposed *u-enpytag* and Suryanto et al. [2009] proposed *u-top3qu* and *u-top3qu*. The others user features were proposed by us.

User Graph Indicators capture the expertise level of the users who answer questions by examining their relationships. While these indicators could be classified as *user indicators*, we decided to study them separately since they are particularly demanding to obtain. More specifically, we created a graph G where each node represents a user and an edge from user u to user v indicates that u answered a question posted by v. This graph was initially proposed by Zhang et al. [2007], and later used in Agichtein et al. [2008], to estimate the expertise of a user, a method named as *ExpertiseRank*. ExpertiseRank is the PageRank value computed over G' (the transposed of G) [Page et al., 1998]. Like Agichtein et al. [2008], in addition to the actual PageRank value over G' (*ug-exprank*), we also use as feature the PageRank over G (*ug-prank*) and compute the HITS algorithm to create the authority and hub features (*ug-hub*, *ug-auth*) [Kleinberg, 1999].

Regarding the cost of creating each feature in Q&A Forum domain, similar to the Wiki domain, textual features are those with the lowest cost as they need just some simple text parsing. However, for the Relevance Features, it is necessary some extra preprocessing such as part-of-speech recognition ($tm\text{-}nwvers_S$,$tm\text{-}nwnous_S$,$tm\text{-}nwadjs_S$, $tm\text{-}aspan_S$) as well as bigrams and generalizations using the WordNet supersense. Most of the Edit History and User features are not demanding to obtain, however, all the user features using categories (tags) have a higher cost as they need to filter the categories from the answer being evaluated in order to compute these features. Similar to the Article Graph, User Graph features are more expensive to obtain as we need to create the graph using all users from our collection.

Overall, we exploit a set of 186 quality indicators in this domain. Of these, 89 have never been previously used in the Q&A Forums domain. These new features are marked with an "*" in Tables 3.3 and 3.4.

3.3 Multi-View Meta-learning to Assess Quality

In the previous sections, we grouped our indicators to represent different *views* of quality. This process resembles the procedure of obtaining a single assessment of quality by combining the opinion of several experts, each with is own view. Thus, each expert/view would assess quality according to a different but complementary perspective.

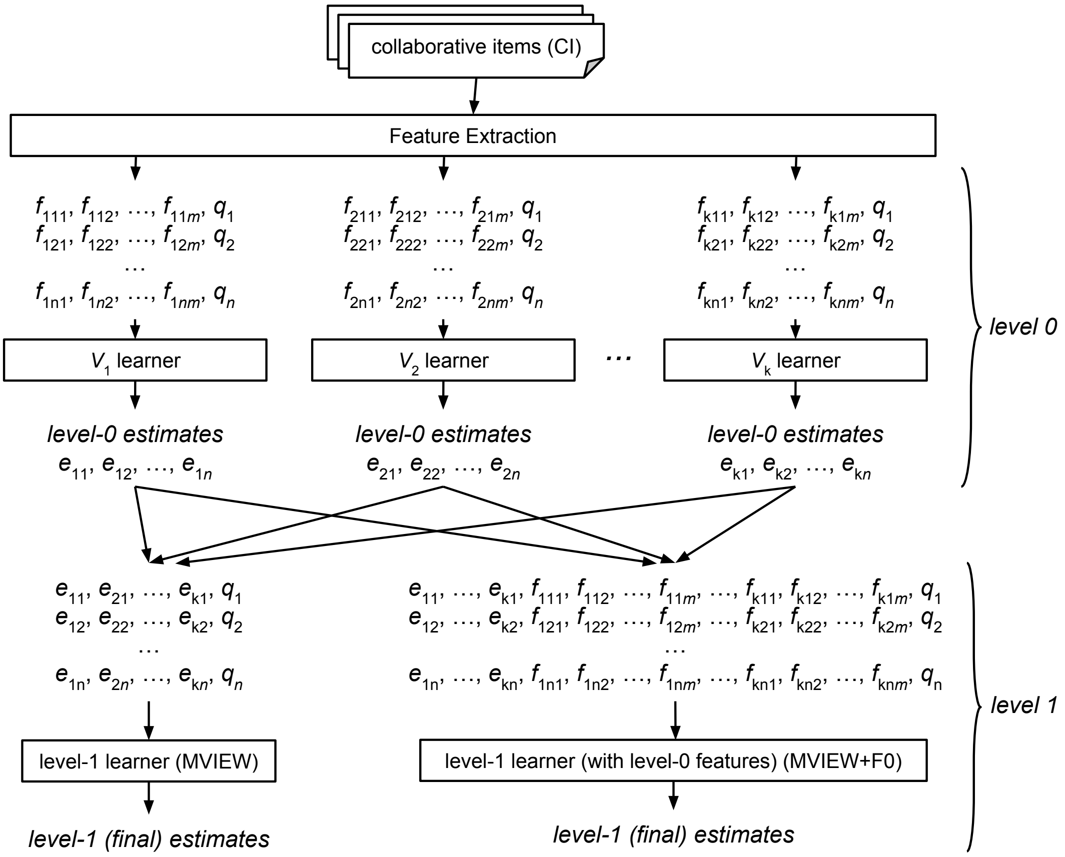

collaborative items (CI)
Feature Extraction
$f_{111}, f_{112}, \ldots, f_{11m}, q_1$
$f_{121}, f_{122}, \ldots, f_{12m}, q_2$
$\ldots$
$f_{1n1}, f_{1n2}, \ldots, f_{1nm}, q_n$
$f_{211}, f_{212}, \ldots, f_{21m}, q_1$
$f_{221}, f_{222}, \ldots, f_{22m}, q_2$
$\ldots$
$f_{2n1}, f_{2n2}, \ldots, f_{2nm}, q_n$
$f_{k11}, f_{k12}, \ldots, f_{k1m}, q_1$
$f_{k21}, f_{k22}, \ldots, f_{k2m}, q_2$
$\ldots$
$f_{kn1}, f_{kn2}, \ldots, f_{knm}, q_n$
V_1 learner
V_2 learner
...
V_k learner
level-0 estimates
$e_{11}, e_{12}, \ldots, e_{1n}$
level-0 estimates
$e_{21}, e_{22}, \ldots, e_{2n}$
level-0 estimates
$e_{k1}, e_{k2}, \ldots, e_{kn}$
level 0
$e_{11}, e_{21}, \ldots, e_{k1}, q_1$
$e_{12}, e_{22}, \ldots, e_{k2}, q_2$
$\ldots$
$e_{1n}, e_{2n}, \ldots, e_{kn}, q_n$
$e_{11}, \ldots, e_{k1}, f_{111}, f_{112}, \ldots, f_{11m}, \ldots, f_{k11}, f_{k12}, \ldots, f_{k1m}, q_1$
$e_{12}, \ldots, e_{k2}, f_{121}, f_{122}, \ldots, f_{12m}, \ldots, f_{k21}, f_{k22}, \ldots, f_{k2m}, q_2$
$\ldots$
$e_{1n}, \ldots, e_{kn}, f_{1n1}, f_{1n2}, \ldots, f_{1nm}, \ldots, f_{kn1}, f_{kn2}, \ldots, f_{knm}, q_n$
level-1 learner (MVIEW)
level-1 learner (with level-0 features) (MVIEW+F0)
level-1 (final) estimates
level-1 (final) estimates
level 1

Figure 3.2. Example of quality assessment of many collaborative items (CI) using the multi-view framework. In this figure, v is a view; q_i is the target quality value and represents the real quality score of each collaborative item (CI) i; f_{vij} is feature j for CI i in view v. Finally, e_{vi} represents the estimated quality in each view v for CI i, at learning level-0. The value e_{vi} can then be used as a feature in learning level-1.

This idea naturally suggests a meta-learning approach with two phases. In the first phase, experts are trained to provide their opinions regarding each view, using their respective indicators. In the second phase, their "opinions" are combined.

The second combination phase of expert opinions can be performed using simple methods, such as weighted majority voting or weighted average. However, the problem of how to weight the different views remains. Which views are more important: Relevance or Readability, Style or Edit History? Moreover, the importance (i.e., weight) of each opinion in the final estimate is context-dependent. This led us to adopt the machine learning strategy of *stacking* [Witten and Frank, 1999].

From now on, we mostly refer to indicators as *features*, as this is the commonly used terminology in machine learning. In the first phase, each item is represented by k

sets of features (i.e., k *views*). Given a training set, we split it into k partitions, one for each view. Using the k resulting training sets, an algorithm (the *level-0 learner*) can learn a model for each view. In the second phase, given the assessments of each level-0 model, an algorithm (the *level-1 learner*) can learn a global assessment. Thus, after training, the level-1 model can be used to provide a single quality assessment. This approach is interesting because it still provides individual view assessments (level-0 models), while at the same time learns the best way to combine different quality views. This idea is illustrated in Figure 3.2.

More formally, let v be a view and $c_{v1}, c_{v2}, ..., c_{vn}$ be n collaborative items representations for CI i, each one using the features $F_{v1}, F_{v2}, ..., F_{vm}$ in view v. As presented in Figure 3.2, the collaborative item c_{vi} is represented by the vector $(f_{vi1}, f_{vi2}, ..., f_{vim})$, where each f_{vij} is the value of feature F_{vj} in c_{vi}. As a result, each collaborative item i can be represented by the set of estimates $\{e_{1i} \ldots e_{ki}\}$, one estimate for each one of the k views. Note we can now use this new representation to build a new training set $\{(c_1, q_1), ..., (c_n, q_n)\}$, where c_i corresponds to CI i, now represented by estimates $\{e_{1i} \ldots e_{ki}\}$. Using this new training set we can learn a global model (level-1 model) to assess the quality of item i (represented by c_i). By doing so, we are learning how to combine the estimates obtained from each different view of quality. Alternatively, we can combine the level-0 features with level-1 features, as also shown in Figure 3.2, to learn a model that captures both individual feature patterns and grouped feature patterns. From now on, we refer to the level 1, represented without level-0 features, as MVIEW. When the level-1 is represented also with level-0 features, we refer to it as MVIEW+F0.

In sum, our approach can be described as follows. Given a collaborative free-editing repository, (1) a set of features is extracted and partitioned according to a set of views; (2) level-0 learners build models corresponding to each view; and (3) a level-1 learner builds a global model based on the outcomes of level-0 learners. In this section we show how to apply the multi-view approach to the Wiki and Q&A Forums domains. To accomplish this, we first present the learners adopted in each domain.

3.3.1 Level-0 and Level-1 Learners for Wikis

In Wiki repositories, the quality of an article is usually assigned to a value on a discrete scale. In Wikipedia, for instance, articles are classified, by the users, using the following classes[11] [Wikipedia, 2015a]:

[11] Note that, currently, there is also an intermediate class between ST and BC, the *C-Class*. We do not use this class because it did not exist at the time we performed our crawling. However, this does

- Featured Article (*FA*): Articles assigned to this class are, according to the evaluators, the best Wikipedia articles.

- A-Class (*AC*): A-Class articles are considered complete, but with a few pending issues that need to be solved in order to be promoted to Featured Articles.

- *Good Article* (*GA*): Good Articles are those without problems of gaps or excessive content. These are good sources of information, although other encyclopedias could provide better content.

- B-Class (*BC*): Articles assigned to this class are considered useful for most users, but lacking more precise information.

- Start-Class (*ST*): Start-Class articles are still incomplete, although containing references and pointers for more complete information.

- Stub-Class (*SB*): Stub-Class articles are draft articles, with very few paragraphs. They also have few or no citations.

Although quality levels are here described by a discrete taxonomy, in general quality can be seen as a value on a continuous scale, 0 (SB) to 5 (FA), in the case of Wikipedia. In fact, this is the most natural interpretation for the problem, if we consider that there are better or worse articles, even inside the same discrete category. For instance, in Wikipedia, we have class A articles that: (a) have recently been promoted and await expert evaluation; (b) have been evaluated by experts and await corrections; and (c) have been corrected and await promotion to featured article. In the case of other Wikis, a continuous scale is commonly used, where users score each article with a value from 1 to 5 and the final quality value is the average of all scores.

Thus, we consider quality in a continuous scale and, consequently, model the problem of quality learning as a numerical regression task. By doing so, our assessments consist of numeric values still related to the original categorical quality scale. Thus, for instance, we expect that the regressor will assign a value close to 4 to a Wikipedia article just promoted to A, while to an A-class article already awaiting promotion to FA, the regressor will assign a value close to 5.

More specifically, in order to predict the quality of Wikis articles, we use Support Vector Regression (SVR) [Drucker et al., 1996] as level-0 and level-1 learners. To apply SVR to the quality estimation task, we represent the articles as follows. Given a view v, let $A_v = \{\mathbf{a}_{v1}, \mathbf{a}_{v2}, ..., \mathbf{a}_{vn}\}$ be a set of article representations. Each article

not affect our analyses.

$\mathbf{a}_{vi}$ is represented by a set of m features $F_v = \{F_{v1}, F_{v2}, ..., F_{vm}\}$, such that $\mathbf{a}_{vi} = (f_{vi1}, f_{vi2}, ..., f_{vim})$ is a vector representing $\mathbf{a}_{vi}$, where each f_{vij} is the value of feature F_{vj} in $\mathbf{a}_{vi}$. We here assume that we have access to some *training data* of the form $A_v \times \mathbb{R} = \{(\mathbf{a}_{v1}, q_1), (\mathbf{a}_{v2}, q_2), ..., (\mathbf{a}_{vn}, q_n)\}$, where each pair $(\mathbf{a}_{vi}, q_i)$ represents an article $\mathbf{a}_{vi}$ and its corresponding quality assessment value q_i, such that if $q_1 > q_2$, then the quality of instance $\mathbf{a}_{v1}$, as perceived by the user, is higher than the quality of instance $\mathbf{a}_{v2}$. Using the training data, we apply regression with SVR, as explained in Section 2.2.1, to find the best combination of the features, for each view v, and predict the quality value q_i for any given article $\mathbf{a}_{vi}$.

3.3.2 Level-0 and Level-1 Learners for Q&A Forums

In Q&A Forums, a user (asker) can post a question about a certain topic for which she receives answers from other users. Normally, any user can label a particular answer as useful or not, while the asker can indicate the one she considers the best. Figure 3.3 illustrates the main elements of a Q&A Forum, here using the particular case of Stack Overflow[12]. As we can see, in Stack Overflow, any user can annotate whether an answer is useful or not, and vote for it favorably (upvote) or negatively (downvote). The asker can place a mark (a green "tick") on the answer he/she considers the best.

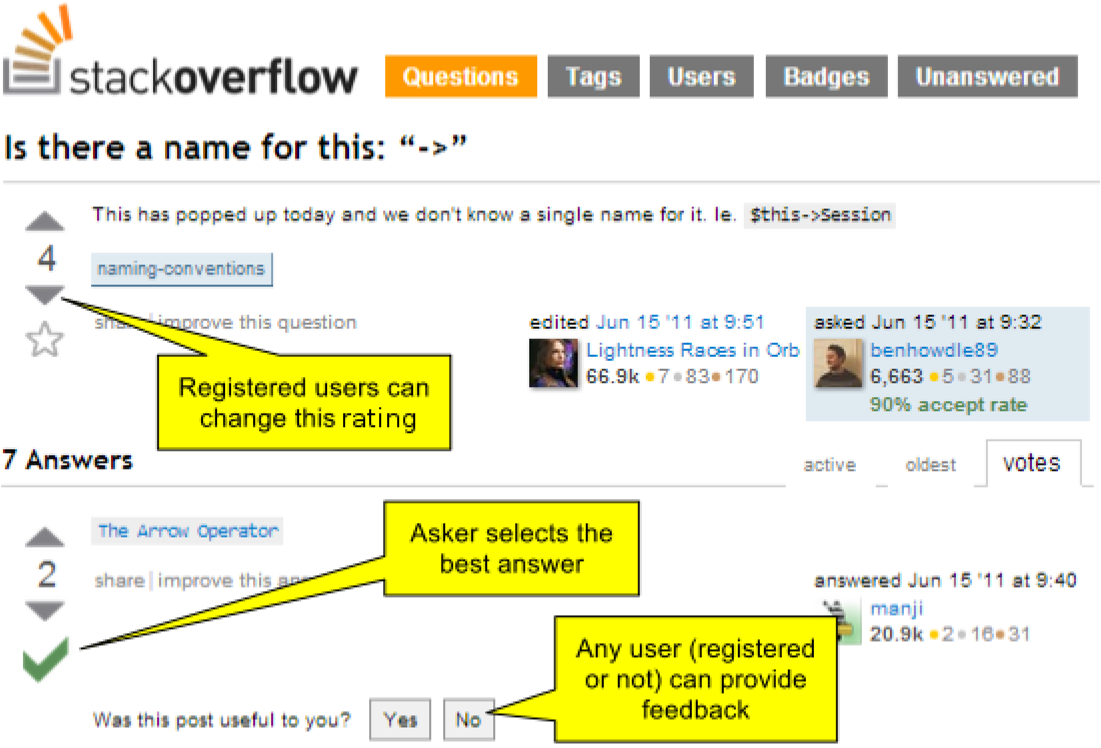
In forums such as Stack Overflow, the answers are expected to be correct and should be ranked according to their quality. The Stack Overflow guide[13] states that a good answer, besides being correct, should be clear, provide examples, quote relevant material, be updated, and link to more information and further reading.

In this work, we assume that the difference between upvotes and downvotes is an indicative of the answer quality. Note we do not treat such difference as an objective numeric quality assessment. Instead, we use it to sort answers according to their estimated quality. Thus, as we did for Wikis in the Section 3.3.1, we consider quality in a continuous scale in Q&A forums. However, differently from the Wiki domain, instead of predicting an objective numeric quality score, we are interested on ranking the answers according to their quality. Thus, in this domain we are going to use a learning to rank strategy. In particular, we adopt a well-known Learning to Rank algorithm, SVMRank detailed in Section 2.2.2.

The idea of learning to rank can be straightforwardly used in the Q&A Forums domain. Here, questions can take the role of queries and answers can take the role of documents. Then, to apply SVMRank to the quality estimation task, we represent the an-

[12] `http://www.stackoverflow.com`

[13] `http://meta.stackoverflow.com/questions/7656/how-do-i-write-a-good-answer-to-a-question`



Is there a name for this: “->”

This has popped up today and we don't know a single name for it. Ie. `$this->Session`

Was this post useful to you? Yes No

Registered users can change this rating

Asker selects the best answer

Any user (registered or not) can provide feedback

Figure 3.3. Example of a question in the Stack Overflow Q&A Forum (*Is there a name for this:* ‘$- >$’), for which one, out of seven answers, is shown. The figure also illustrates the tools users can use to indicate how good are the answers.

.

swers as follows. Without loss of generality, let v be a view and $A_v = \{\mathbf{a}_{v1}, \mathbf{a}_{v2}, ..., \mathbf{a}_{vn}\}$ be a set of answers for a question. Each answer $\mathbf{a}_{vi}$ is represented by a set of m features $F_v = \{F_{v1}, F_{v2}, ..., F_{vm}\}$, such that $\mathbf{a}_{vi} = (f_{vi1}, f_{vi2}, ..., f_{vim})$ is a vector representing $\mathbf{a}_{vi}$, where each f_{vij} is the value of feature F_{vj} in $\mathbf{a}_{vi}$.

Thus, to learn a ranking, we assume that we have access to some *training data* of the form $A_v \times \mathbb{R} = \{(\mathbf{a}_{v1}, r_1), (\mathbf{a}_{v2}, r_2), ..., (\mathbf{a}_{vn}, r_n)\}$, where each pair $(\mathbf{a}_{vi}, r_i)$ represents an answer $\mathbf{a}_{vi}$ and its corresponding quality ranking score r_i for a certain question, such that if $r_i > r_j$ then $\mathbf{a}_{vi}$ should be ordered before $\mathbf{a}_{vj}$ ($\mathbf{a}_{vj} \succ \mathbf{a}_{vi}$). This training data will be the input for the SVMRank method as explained in Section 2.2.2.

Chapter 4

Dataset and Evaluation Methodology

In this chapter, we introduce all the six datasets used in this thesis (three from each domain). After that, we present our evaluation methodology as well as all the evaluation metrics used in our analysis.

4.1 Datasets

4.1.1 Wiki Datasets

In this domain, we used three datasets in our experiments: a sample extracted from the English Wikipedia and two others Wikis, provided by Wikia service. Wikipedia was used due to its prominence and its large amount of articles with quality manually assessed by users [Wikipedia, 2015a][1]. From now on, we refer to this Wiki dataset as WIKIPEDIA.

From the Wikia service, we selected the Wikis *Wookieepedia*[2], about the Star Wars universe, and *Muppet*[3], about the TV series "The Muppet Show". These are the wikis in Wikia with the largest number of articles with quality manually assessed[4]. Their repositories are freely available for download [5,6].

The Wookieepedia collection uses a simplified version of the Wikipedia quality taxonomy, comprising only classes *FA*, *GA*, and *SB*. From now on, we refer to it as

[1] Any user can evaluate a Wikipedia article, according to the quality taxonomy detailed in the Section 3.3.1

[2] `http://starwars.wikia.com/`

[3] `http://muppet.wikia.com/`

[4] To obtain the article evaluations we used the APIs provided at `http://starwars.wikia.com/api.php` and `http://muppet.wikia.com/api.php`

[5] `http://starwars.wikia.com/wiki/Special:Statistics`

[6] `http://muppet.wikia.com/wiki/Special:Statistics`

STARWAR. The Muppet collection, on the other hand, provides a star-based taxonomy, commonly used by Wikia collections. In this taxonomy, the worst articles receive one star while the best articles receive five stars. The final rating is calculated by averaging all the user ratings. As a consequence, Muppets articles can have a fractional rating value, such as 2.7 stars. We will refer to the Muppet collection as MUPPETS.

Each sample size is presented in Table 4.1. To avoid an imbalanced category distribution when creating the samples, we extracted the same number of articles from each quality category. We chose to use balanced samples since SVR can be biased towards the majority class, which could harm our analysis regarding the relative difficulty of classification in each class. Such procedure was also adopted in previous work (cf. Weiss and Provost [2003]). In the case of MUPPETS we rounded the star ratings to the closest integers.

To obtain the features associated with the article graph, we collected the links between articles from each Wiki including the ones not in the samples. The total number of articles and revisions as well as information about the article graphs are shown in Table 4.1, for all datasets. In that table, edges correspond to links between pages and nodes correspond to the article pages. We used the *Web Graph* library [Boldi and Vigna, 2004] to create the graph and extract all network attributes.

Table 4.1. Sample size, for each Wiki dataset used in our experiments.

Dataset	# Articles	# Reviews	# Edges	# Nodes	Version date
WIKIPEDIA	3,294	1,992,463	86,077,675	3,185,457	Jan/2008
MUPPETS	1,550	38,291	282,568	29,868	Sep/2009
STARWAR	1,446	127,551	1,017,241	106,434	Oct/2009

4.1.2 Q&A Forums Datasets

Our datasets for Q&A Forums consist of three Stack Exchange[7] forum samples, namely (1) *Stack Overflow*, a Q&A Forum for programmers, (2) *Seasoned Advice*, a cooking Q&A Forum[8]; and (3) *English language and Usage*, an English language Q&A Forum[9]. From now on, we call these datasets as STACK, COOK, and ENGLISH respectively.

Stack Overflow was chosen because it is the largest forum hosted by Stack Exchange. The remaining forums were chosen because they cover completely different topics, allowing us to broaden our analyses. We also highlight that each of these Stack Exchange forums focus on a specific topic. Questions not related to these topics are

[7] http://stackexchange.com/

[8] http://cooking.stackexchange.com/

[9] http://english.stackexchange.com/

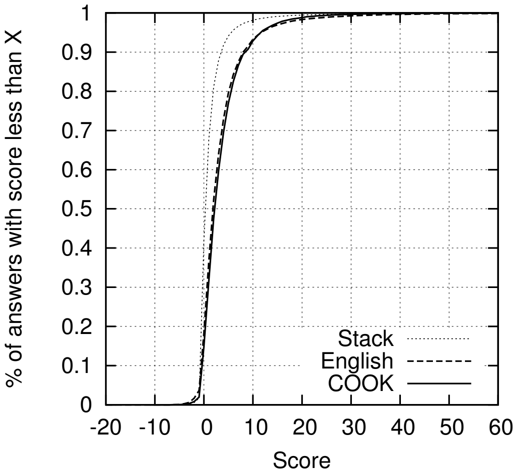


Figure 4.1. Percentage of answers with score smaller than X

removed or marked as closed by their administrators. Thus, we expect they contain less spam and noise than more general Q&A Forums such as YahooAnswers[10].

Table 4.2. Sample size, for each Q&A Forum dataset used in our experiments.

Dataset	# Questions	# Answers	Version date
STACK	9,721	53,263	Mar/2012
COOK	1,751	10,086	Feb/2013
ENGLISH	5,751	31,084	Mar/2013

All the randomly selected samples are described in Table 4.2. Note we consider only questions with at least four answers, since questions with less answers can be easily assessed by the users such that an automatic ranking system is of little utility. To create the user graph (cf. Section 3.2), we considered all the Q&A Forums users (even the ones which are not in the sample) and their questions and answers.

The ground truth for our approach is the difference between upvotes and downvotes, which we refer to as the *answer rating.* More specifically, the rating r_a for an answer a is given by Equation 4.1 below

$$r_a = r'_a + r'_{min} \tag{4.1}$$

where $r'_a = u_a - d_a$ is the difference between the number of upvotes u_a and downvotes d_a received by answer a, and r'_{min} is the minimum difference between upvotes and downvotes observed in each collection, used to avoid negative values. Note that, the oldest answer can tend to a higher ratting than a newer one. However, in this thesis,

[10] `http://answers.yahoo.com/`

we decided to do the same as our baselines and we did not take into account when the answer received the score.

As shown in Figure 4.1, the answer rating distribution in all datasets follows a power law. For example, in STACK ratings vary from -166 to 505, with values from -20 to 15 corresponding to 99% of the instances in our sample. Such a skewed distribution is due to the popularity of the answers, with only a few of them attracting large audiences. This same behavior is also observed in the other Q&A Forum collections.

4.2 Evaluation Methodology

In this section, we describe the evaluation methodology we use to perform our experiments. Our goals are threefold: (1) to compare our proposals with state-of-the-art baseline methods in the task of automatic quality assessment; (2) to understand the impact of each group of features in these tasks; and (3) to analyse the impact of views and their interactions in the context of the proposed meta-learning approach.

4.2.1 Evaluation Setup

In our experiments, we utilized an n-fold cross validation procedure [Mitchell, 1997]. Each dataset was randomly split into n parts, such that, in each run, one part was used as a test set, one part was used as the validation set for parameter tuning and the remaining parts were used as the training set. The split on training and test sets was the same in all experiments. For the Wikis domain we used a 10-fold cross validation. For the Q&A Forums domain, we used a 5-fold cross validation. We opted for fewer folds in the latter due to the large size of the Q&A Forums datasets. Nevertheless, in both domains, we observed little variation in the folds.

To analyse level-0 features and effectiveness without meta-learning, we performed a n-fold cross validation procedure in the dataset without any modification. However, when analysing the meta-learning approach, in order to make sure that the same training and test instances are used by the methods in each cross validation turn and that test information of level-0 is not used as training information of level-1, we carry out the experiments according to Algorithm 2. For the baseline of the meta-learning method, we computed the results using PTs_{vp}, defined in Algorithm 1[11], considering that there

[11] Note that, in order to simplify our explanation in Section 3.3, we represented the predictions in level-0 using the variable e_{vi}, but here, to make it clearer, we used Pts_{vp} for the predictions in the test set and Ptr_{vp} for the predictions in training set.

Algorithm 2 Training and testing procedures

```
Require: $V$ is a set of views
Require: For each view $v \in V$, $X_v$ is a dataset divided into $n$ partitions, $X_v = \{x_{v1} ... x_{vn}\}$
  for $p = 1$ to $n$ do
      for all $v \in V$ do
          $train \leftarrow X_v - x_{vp}$
          $test \leftarrow x_{vp}$
          $t \leftarrow n - 1$
          Do a cross validation of $t$ partitions in $train$ to obtain the predictions $Ptr_{vp}$
  and the training models $\{M_{vp1} ... M_{vpt}\}$
          Apply the model $M_{vp1}$ in $test$ to obtain the predictions $PTs_{vp}$
  for $p = 1$ to $n$ do
      Create the training set $T_p$ using the predictions $Ptr_{vp}$ for each $v \in V$
      Create the test set $S_p$ using the perditions $Pts_{vp}$ for each $v \in V$
      $t \leftarrow n - 1$
      Do a cross validation of $t$ partitions in $T_p$ to obtain the training models
  $\{M_{p1} ... M_{pt}\}$
      Apply the model $M_{p1}$ in the test to obtain the predictions $P_p$
```

is just one view V, with all the features, and for the meta-learning method we computed the results using P_p.

For all comparisons reported in this work, we used the signed-rank test of Wilcoxon [Wilcoxon, 1945] to determine if the differences in effectiveness were statistically significant. This is a nonparametric paired test that does not assume any particular distribution on the tested values. In all cases, we only draw conclusions from results that were considered statistically significant with at least 95% of confidence level.

4.2.2 Evaluation Metrics

In the Wiki domain, since we are dealing with regression methods, we evaluate their effectiveness by using the *mean squared error* measure (MSE). MSE is defined as:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} e^2 \tag{4.2}$$

where e is the error value and n is the number of articles. We compute error e as the absolute difference between the quality value predicted and the true quality value, extracted from the database. In our experiments, we used quality values from 0 (Stub article) through 5 (Featured Article) for WIKIPEDIA, 0 (Stub article) through 2 (Featured Article) for STARWAR, and 1 (one star) through 5 (five stars) for SW5 and MUPPETS.

In the Wiki domain, we also analyze the performance per class. To accomplish this, we round the predictions to the closest integers representing the class. For example, if rating 2.3 is assigned to article a in WIKIPEDIA we consider 2 as the quality class (BC) of a. We can then compute the F1 measure for class i as:

$$\mathrm{F1(i)} = 2 \times \frac{precision(i) \times recall(i)}{precision(i) + recall(i)} \tag{4.3}$$

where $precision(i)$ is the proportion of instances correctly predicted as class i and $recall(i)$ is the proportion of instances of the class i correctly classified.

To evaluate our assessment strategies in Q&A Forums domain, we adopt ranking comparison metrics since we treat the problem in this domain as a ranking task. In particular, we used the *Normalized Discounted Cumulative Gain at top k* (NDCG@k). This metric, first proposed in Järvelin and Kekäläinen [2000], measures how close the predicted quality ranking of answers is to their true quality ranking. More formally, NDCG@k is defined as:

$$NDCG@k = \frac{1}{N}\sum_{i=1}^{k}\left(\frac{r_i}{\log_2(i+1)}\right) \tag{4.4}$$

where r_i is the true quality assessment for the answer at position i in the ranking, and N is a normalization factor. The factor N is equal to the *discounted cumulative gain* (the sum part in equation (4.4)) of an *ideal ranking*. The ideal ranking in the Q&A Forums domain is the ranking where, given a pair of answers (a_i, a_j), a_i is better ranked than a_j if r'_i is greater than r'_j (cf. Equation 4.1). Thus, the higher the high quality documents are placed in the ranking, the higher the value for NDCG@k. In addition, note that, in the Q&A Forums domain, we compute the NDCG@k for all questions and then compute their average.

In order to evaluate the concordance among views, we adopt the percentage of agreement used in Fleiss' kappa [Fleiss and Cohen, 1973]. This computes the agreement among multiple raters, in our case, the views. More specifically, for a given instance i, we compute the percentage of agreement P_i, defined as:

$$P_i = \frac{1}{n(n-1)}\sum_{j=1}^{k} n_{ij}(n_{ij}-1) \tag{4.5}$$

where n is the number of quality classes and n_{ij} is the number of views that were assigned the same class j for instance i. Note that, we discretized the quality score predicted for each view by rounding it to obtain the integer value j.

We also study the views behavior using correlation metrics. Regarding Wikis, we use the Pearson correlation coefficient [Anthony J. Onwuegbuzie, 2007]. In the case of Q&A Forums we use the Kendall Tau ranking correlation coefficient [Kendall, 1938], since we treat that as a ranking problem[12].

Given two prediction vectors X and Y, the Pearson correlation is defined by Equation 4.6.

$$Pearson(X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(\frac{X_i - \overline{X}}{\sigma(X)})(\frac{Y_i - \overline{Y}}{\sigma(Y)}) \tag{4.6}$$

where n is the number of instances, $\overline{X}$ and $\overline{Y}$ are the average predictions of the arrays X and Y, respectively, and $\sigma(X)$ and $\sigma(Y)$ are their standard deviations.

As for Q&A Forums, we take the Kendall Tau ranking correlation per query, given by Equation 4.7.

$$Kendall(Q) = \frac{countA(Q) - countD(Q)}{0.5n(n-1)} \tag{4.7}$$

In this equation, n is the number of answers given to a question q, Q represents a set of pairs (x_i, y_i), where x_i and y_i are the rank positions of answer i for q, obtained by two different ranking models, x and y. Functions $countA$ and $countD$ count the agreement $(x_i = y_i)$ and disagreement $(x_i \neq y_i)$ among these pairs. The final value we use corresponds to the average of the Kendall(Q) computed for all questions.

To analyze the impact of the features within views we use the SPEA2 feature selection algorithm, described in Section 2.3. To accomplish this, in our experiments we use the following parameters: number of generations G = 50; population size N = 75; archive size A = 225; mutation probability p_m = 0.3; and crossover probability p_c = 0.8. These parameters were chosen according to Laumanns et al. [2001].

[12] Note that, in the Wiki domain, we also tested correlations using the Kendal-tau coefficient. As the conclusions we observed are the same, we report only Pearson results.

Chapter 5

Results on Wikis

In this chapter we present our results and analysis regarding the Wiki domain. First, we compare our approach with baselines, after that we present an analysis regarding performance and correlation of the views. Finally, the results regarding the feature selection and analysis are presented.

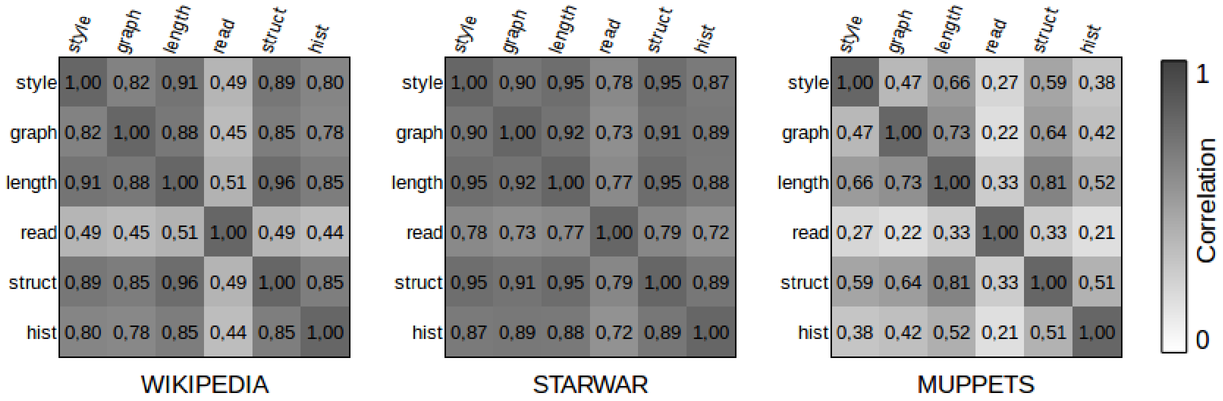
5.1 Comparison with Baseline

We start by describing the impact of using meta-learning on quality evaluation. Table 5.1 presents the MSE (cf. Section 4.2.2) results for each collection. MVIEW is our meta-learning method that uses only view predictions in level-1 (cf. Section 3.3). MVIEW+F0 is the variant of MVIEW that represents articles in level-1 learning using view predictions and level-0 attributes. The results obtained by these methods are compared to the baseline method, SVR, described in Section 2.2.1. Results statistically better than SVR are marked with "*".

Table 5.1 shows that the best performance of MVIEW and MVIEW+F0 was obtained in STARWAR with gains of up to 30%. The worst results occur in MUPPETS. In fact, in MUPPETS, the multi-view approaches were not able to reach statistically significant gains over the baseline. We can also observe that the best method in WIKIPEDIA was MVIEW+F0.

5.2 Analysis of the Views

To better understand the multi-view performance, we first analyze the agreement between the views. To this end, Figure 5.1 shows the Pearson correlation between the view predictions. In the figure, the higher the correlation between the views, the darker the cell color. We can observe that, in general, most views are correlated with each other. This is specially evident in WIKIPEDIA and STARWAR, when compared to MUPPETS. The least correlated view is Readability in all collections.



WIKIPEDIA

	style	graph	length	read	struct	hist
style	1,00	0,82	0,91	0,49	0,89	0,80
graph	0,82	1,00	0,88	0,45	0,85	0,78
length	0,91	0,88	1,00	0,51	0,96	0,85
read	0,49	0,45	0,51	1,00	0,49	0,44
struct	0,89	0,85	0,96	0,49	1,00	0,85
hist	0,80	0,78	0,85	0,44	0,85	1,00

STARWAR

	style	graph	length	read	struct	hist
style	1,00	0,90	0,95	0,78	0,95	0,87
graph	0,90	1,00	0,92	0,73	0,91	0,89
length	0,95	0,92	1,00	0,77	0,95	0,88
read	0,78	0,73	0,77	1,00	0,79	0,72
struct	0,95	0,91	0,95	0,79	1,00	0,89
hist	0,87	0,89	0,88	0,72	0,89	1,00

MUPPETS

	style	graph	length	read	struct	hist
style	1,00	0,47	0,66	0,27	0,59	0,38
graph	0,47	1,00	0,73	0,22	0,64	0,42
length	0,66	0,73	1,00	0,33	0,81	0,52
read	0,27	0,22	0,33	1,00	0,33	0,21
struct	0,59	0,64	0,81	0,33	1,00	0,51
hist	0,38	0,42	0,52	0,21	0,51	1,00

Figure 5.1. Correlations between views in datasets WIKIPEDIA, STARWAR, and MUPPETS: the darker the color, the higher the correlation (only positive correlations were observed). Labels (left to right, top to bottom) correspond to views Style, Article Graph, Length, Readability, Structure, and Edit History, respectively.

In general, the best performances were obtained in datasets where the views were more correlated (WIKIPEDIA and STARWAR). This strengthens the notion that view agreements are useful as they reinforce correct predictions. However, independent views could contribute to more diverse estimates and, by extension, to correct predictions for a larger number of articles, specially if the views are good in different parts of the dataset. As observed by dos Santos et al. [2006], this happens when the multi-view method is able to learn the parts of the dataset in which particular views perform

Table 5.1. MSE obtained by approaches MVIEW, MVIEW+F0, and SVR in datasets WIKIPEDIA, STARWAR, and MUPPETS.

Sample	Method	MSE	% Gain
WIKIPEDIA	SVR	0.887	-
	MVIEW	0.873*	+1.6%
	MVIEW+F0	0.834*	+5.9%
STARWAR	SVR	0.084	-
	MVIEW	0.058*	+30.9%
	MVIEW+F0	0.068*	+19.0%
MUPPETS	SVR	1.690	-
	MVIEW	1.693	-0.2%
	MVIEW+F0	1.703	-0.8%

better. To investigate how well the views perform in different parts of the datasets, in Table 5.2 we present the proportion of instances in which each specific view provided the best estimate (i.e, the estimate with lowest regression error).

Table 5.2. Proportion of instances where each specific view provided the best estimate among all the views. Error is calculated assuming a 95% confidence interval and a Normal distribution.

Sample	**Style**	**Art. Graph**	**Length**	**Readability**	**Structure**	**Ed. History**	**Avg ± Error**
WIKIPEDIA	17%	17%	15%	15%	18%	17%	16.5 ± 0.98
STARWAR	15%	17%	19%	14%	18%	18%	16.8 ± 1.55
MUPPETS	15%	12%	7%	18%	22%	26%	16.7 ± 5.49

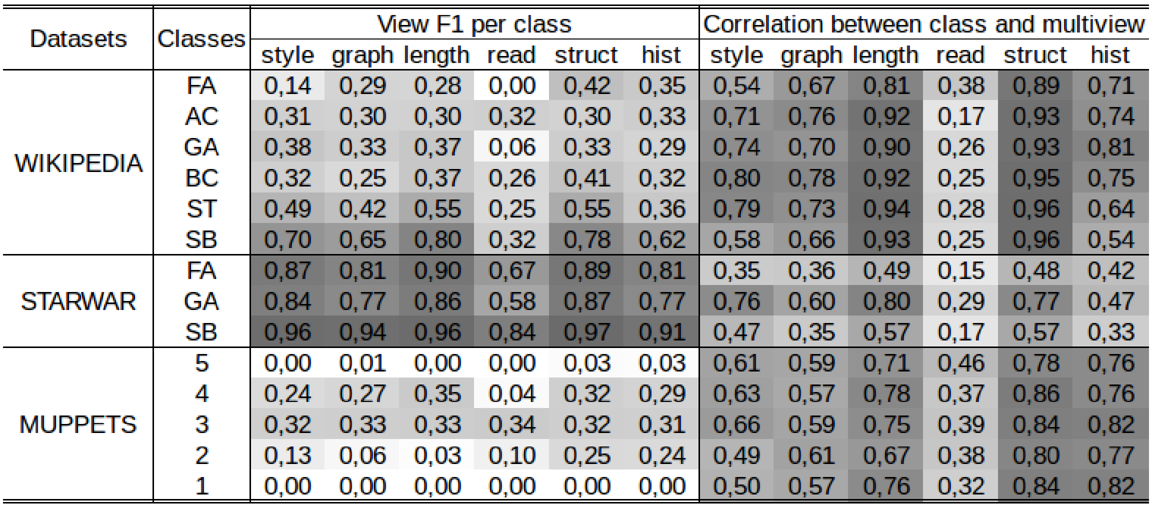
As we can see, in general, the performance of the views is balanced in all the collections with no largely dominant views. This is promising since a largely dominant view v would lead the multi-view approach to the trivial strategy of mirroring the decisions of v. We also note that the largest deviation from average was observed in MUPPETS, where Length and Edit History presented a performance very different from the other datasets.

In order to understand what the multi-view approach (MVIEW) learned from these views, in Figure 5.2, for the set of instances of a given quality class, we present: (a) the performance of each view and (b) the correlation between the view prediction and the MVIEW prediction. To evaluate the performance of the views we converted the view estimates to quality classes and computed the F1 measure as defined in Equation 4.3. Thus, higher values (indicated by dark colors) represent a balanced combination between finding most instances of a quality class (recall) and correctly classifying the instances (precision). Note also that, the correlation between view and MVIEW prediction is important since a view is certainty good if it is correlated to the final quality prediction.

From Figure 5.2, we can conclude that, in general, the views with best performance (Length and Structure in WIKIPEDIA and STARWAR, and Structure and Edit History in MUPPETS) are the most correlated with the multi-view results. Likewise, MVIEW is the least correlated with Readability, which presented the worst performance in all datasets. Such results suggest that MVIEW considered the best solution to predict the article quality.

The low performance of Readability is not surprising since its features do not take into consideration how well the article covers its topic. Thus, even a draft can be considered a good article, if it is readable according to the metrics of Section 3.2.

In WIKIPEDIA, it is clear that MVIEW relied strongly on Length and Structure, which are largely correlated with each other. As consequence, it was more correlated



Datasets	Classes	View F1 per class						Correlation between class and multiview					
		style	graph	length	read	struct	hist	style	graph	length	read	struct	hist
WIKIPEDIA	FA	0,14	0,29	0,28	0,00	0,42	0,35	0,54	0,67	0,81	0,38	0,89	0,71
	AC	0,31	0,30	0,30	0,32	0,30	0,33	0,71	0,76	0,92	0,17	0,93	0,74
	GA	0,38	0,33	0,37	0,06	0,33	0,29	0,74	0,70	0,90	0,26	0,93	0,81
	BC	0,32	0,25	0,37	0,26	0,41	0,32	0,80	0,78	0,92	0,25	0,95	0,75
	ST	0,49	0,42	0,55	0,25	0,55	0,36	0,79	0,73	0,94	0,28	0,96	0,64
	SB	0,70	0,65	0,80	0,32	0,78	0,62	0,58	0,66	0,93	0,25	0,96	0,54
STARWAR	FA	0,87	0,81	0,90	0,67	0,89	0,81	0,35	0,36	0,49	0,15	0,48	0,42
	GA	0,84	0,77	0,86	0,58	0,87	0,77	0,76	0,60	0,80	0,29	0,77	0,47
	SB	0,96	0,94	0,96	0,84	0,97	0,91	0,47	0,35	0,57	0,17	0,57	0,33
MUPPETS	5	0,00	0,01	0,00	0,00	0,03	0,03	0,61	0,59	0,71	0,46	0,78	0,76
	4	0,24	0,27	0,35	0,04	0,32	0,29	0,63	0,57	0,78	0,37	0,86	0,76
	3	0,32	0,33	0,33	0,34	0,32	0,31	0,66	0,59	0,75	0,39	0,84	0,82
	2	0,13	0,06	0,03	0,10	0,25	0,24	0,49	0,61	0,67	0,38	0,80	0,77
	1	0,00	0,00	0,00	0,00	0,00	0,00	0,50	0,57	0,76	0,32	0,84	0,82

Figure 5.2. F1 values obtained per class and correlations between view and multi-view obtained per class. The larger is F1/correlation, the darker is the color (no negative correlations were observed).

with Length than with Edit History even for classes AC and FA, best predicted by Edit History. In fact, Edit History was more useful to MVIEW exactly in the high-quality classes FA, GA and AC. This is probably due to the fact that Edit History information is sparser for low-quality classes (specially SB and ST) and, for theses classes, the other views provided more reliable estimates. We also observe that the higher are the quality class levels, the more difficult is to distinguish them from each other. This is due to the quality criteria adopted by Wikipedia, where syntactic and structural features are able to distinguish low quality classes. For example, SB (Stub) articles are easily distinguished from Start and BC articles due to its short length. However, it is hard to distinguish FA and GA articles because it is hard to assess how compelling is the writing style or how complete is the content.

Regarding STARWAR, except for Readability, all other views performed well with a slight advantage for Structure, Length and Style. Like in WIKIPEDIA, these dominant views are also very correlated with each other (cf. Figure 5.1). When comparing all the datasets, we note that the STARWAR sample presented the highest F1 values for all views and classes. Such effectiveness can be partially attributed to the characteristics of the classes in this dataset, since the STARWAR taxonomy includes the *SB* class, which can be easily identified due to its usually very small article length. Thus, errors will occur mostly between classes *FA* and *GA*.

For the MUPPETS collection, Structure, Edit History, and Length were the best predictor views and the most correlated to MVIEW. In this dataset, the views were more uncorrelated. However, no view was able to provide reliable estimates, specially for classes 1 and 5. In this dataset, the views were not able to distinguish most of the articles, resulting in low values of precision and recall. As a consequence, MVIEW was not able to learn an useful combination of the view estimates.

The low performance in MUPPETS is probably due to the lack of a precise criteria used in the star-based taxonomy. Differently from the Wikipedia-based taxonomy, the number of stars is not associated with any standard mandatory criteria. As a consequence, criteria used in star-based taxonomies are personal and much more subjective. For instance, whereas in Wikipedia citations have to be present for an article to be classified as *FA*, no similar criterion would be required to classify a Muppets article as five stars. Thus, a user can give a high rate to a Muppets article about a certain character only because she likes that character.

Finally, in Figure 5.3, we observe the relation between view agreement and error – each plotted point corresponds to an instance used as input to the level-1 learner in MVIEW. Each instance is represented by the agreement and the error amongst its views. We calculated the agreement of instance i as the percentage of agreement P_i – Kappa (cf. Section 4.2). We applied jitter to the Kappa values to avoid overlap. The error was obtained using MSE, considering the view estimates as predicted values. The MSE values were standardized to have zero mean and unit variance. To facilitate the visualization of the error distribution, (a) we remove outliers using the Chauvenet's criterion [Barnett and Lewis, 1994] (b) we plot the regression line that best fits the points and (c) split the graph at the center of mass of the points, dividing it into four quadrants.

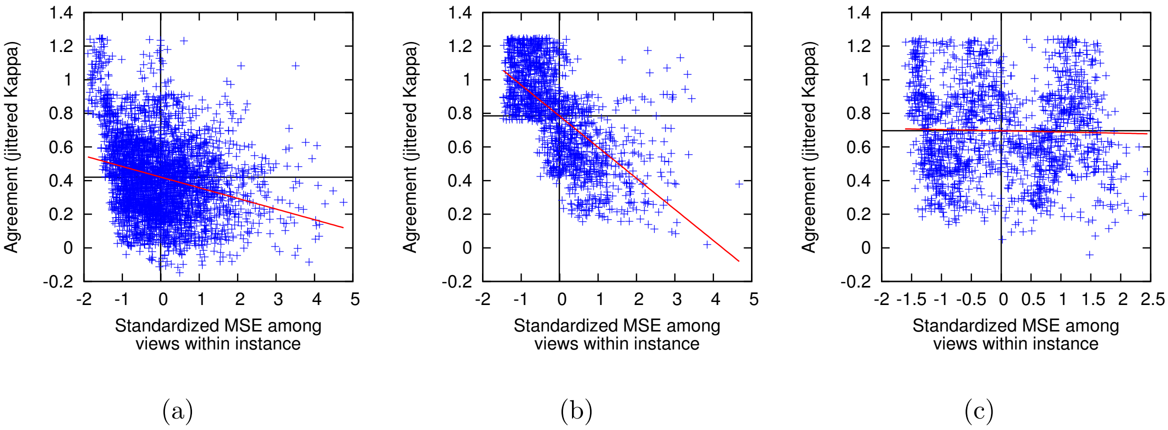


Figure 5.3. Agreement and MSE error per instance in WIKIPEDIA (a), STARWAR (b), and MUPPETS (c).

In the graphs of Figure 5.3, in an ideal situation, points should concentrate in the top-left quadrant (the views agree with each other and correctly classify the instance) and bottom-right quadrant (when the views disagree, the instances are hard to classify, leading to larger errors). As we observe, this is the case for STARWAR(cf. Figure 5.3 (b)), where MVIEW reached the best results. On the other hand, in MUPPETS(cf. Figure 5.3 (c)), the errors are almost evenly distributed. As a result, the views agree in cases where they misclassify the instance. A combinator like MVIEW will hardly improve these cases, since the views are in fact reinforcing a wrong decision. Although a similar situation is observed for WIKIPEDIA, we note that the right-top quadrant is less dense than the left-bottom. Points in left-bottom indicate that although the views disagree with each other, they provide decisions close to the correct one (and, probably, some of them provide correct decisions). This is the most promising situation for a combinator since it has the opportunity to provide a better global decision.

5.3 Analysis of Features within Views

To analyze the impact of the features, we performed feature selection using SPEA2, as described in Section 2.3. Table 5.3 presents the MSE values obtained by tested approaches in datasets WIKIPEDIA, STARWAR, and MUPPETS. SVR corresponds to the use of Support Vector Regression. MVIEW corresponds to the multi-view approach, using SVR as level-0 and level-1 regressors, as previously described. GA corresponds to the multi-view approach, with features selected on level-0 using SPEA2. Similarly, GA-F corresponds to MVIEW+F0 with using just features selected by SPEA2. Results marked with "*" are significantly better than MVIEW while those marked with "†" are significantly worse than it. We note that, after feature selection, the number of features was reduced to 68%, 71%, and 87% of the original number of features in WIKIPEDIA, STARWAR, and MUPPETS, respectively, while the MSE values are, in general, very similar.

Table 5.3. MSE values for collections WIKIPEDIA, STARWAR, and MUPPETS. Column '% of feat.' represents the percentage of features used (an average for GA and GA-F, since MSE values were obtained using cross-validation).

View	Approaches per collection					
	WIKIPEDIA		STARWAR		MUPPETS	
	MSE	% of feat.	MSE	#	MSE	% of feat.
SVR	0.887†	100%	0.084†	100%	1.690*	100%
MVIEW	0.873	100%	0.058	100%	1.693	100%
GA	0.879	31.8%	0.064	28.5%	1.678	13.5%
GA-F	0.871	31.8%	0.067	28.5%	1.672	13.5%

Table 5.4. Features selected by GA method for an arbitrary fold.

	Features per collection		
View	**WIKIPEDIA**	**STARWAR**	**MUPPETS**
Structure	*ts-citpsec*; *ts-imgps*; *ts-secs*; *ts-avsecl*; *ts-avparl*	*ts-citpsec*; *ts-citplen*; *ts-abslen*; *ts-stdsecl*; *ts-maxsecl*; *ts-subsec*; *ts-lnkpl*	*ts-imgps*; *ts-avparl*
Length	*tl-phrcnt*	*tl-wordcnt*	*tl-wordcnt*
Style	*ty-questn*; *ty-conj*; *ty-sprono*; *ty-ssubcnj*	*ty-passive*; *ty-plgphr*; *ty-auxverb*; *ty-sartic*; *ty-ssubcnj*; *ty-sintp*; *ty-prono*	*ty-auxverb*; *ty-sintp*; *ty-prono*
Readability	–	*tr-ari*; *tr-lix*	–
Ed. History	*r-rperusr*; *r-anonym*; *r-rcount*; *r-reguser*; *r-activeu*; *r-stdrevu*; *r-probrev*	*r-age*; *r-anonym*; *r-rcount*; *r-reguser*; *r-modline*; *r-3month*; *r-stdrevu*	*r-age*
Art. Graph	*n-assortoi*; *n-assortoo*; *n-cluster*; *n-pgrank*; *n-reciproc*; *n-translat*	*n-assortoo*; *n-reciproc*	*n-reciproc*

We present in Table 5.4 the set of features selected by SPEA2 that reached the best performances shown in Table 5.3. From Structure view, SPEA2 selected features related to citations and sections in both WIKIPEDIA and STARWAR. In WIKIPEDIA, features *ts-imgps* and *ts-avparl* were also selected. Different subsets of Style features were used for each dataset. For all the datasets, no redundant Length features were selected. Regarding Readability features, the majority was removed from the STARWAR dataset, along with all Readability features from WIKIPEDIA and MUPPETS.

In WIKIPEDIA, Edit History features such as *number of discussions*, and *reviews made by non-registered users* (which is somewhat redundant with *number of edits made by registered users* — *r-rperusr*) were removed. The preserved features were associated with *frequency of editions* (*r-rcount*) and *reputation of the reviewer* (*r-rperusr*, *r-anonym*, *r-reguser*, *r-activeu*, *r-stdrevu*, *r-probrev*). Different feature sets were used in the other collections. For instance, in STARWAR, although it has some features related to the user reputation, features related to the stability of the article were also preserved (*r-modline*, *r-3month*). For the MUPPETS, the only preserved feature was *r-age*.

The following analysis is based on *all* non-dominated solutions selected by SPEA2. As discussed in Section 4.2.2, SPEA2 aims at approximating the Pareto Front. Thus, we can extract and analyze the features present in solutions closer to the pareto front, i.e., features used by the nondominated solutions. We can see an example of Pareto front in Figure 5.4 which shows the WIKIPEDIA SPEA2 solutions, highlighting the nondominated solutions. In Table 5.5 and 5.6 we rank the features on the Pareto front by frequency of appearance in the best solutions, considering our three test collections. This ranking was built using all nondominated solutions from

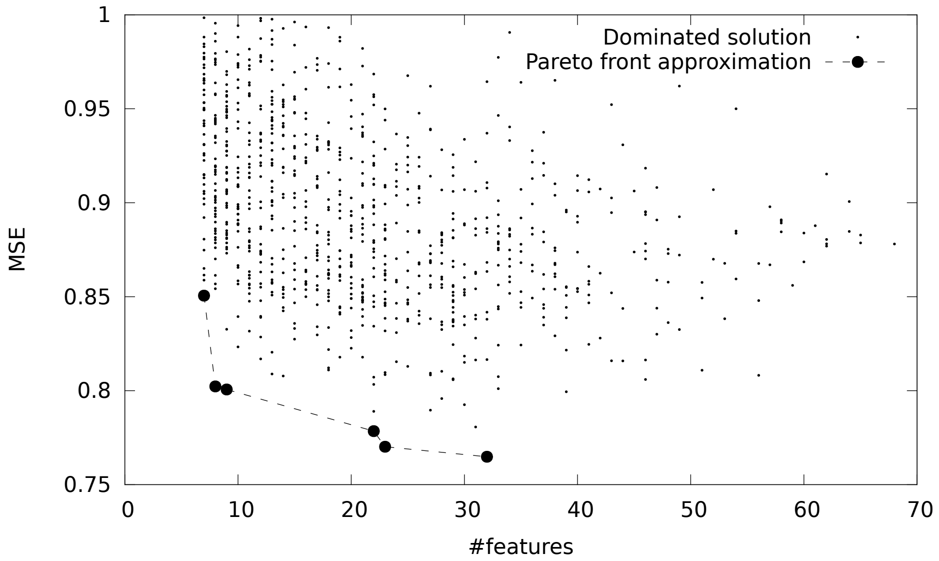


Figure 5.4. Approximation of Pareto front for a WIKIPEDIA sample. Larger dots represent non-dominated solutions.

all folds, totalizing 21 WIKIPEDIA solutions, 28 STARWAR solutions, and 14 MUPPETS solutions, respectively. This ranking is important not only for this analysis but also to know which feature is more important to take into account when predicting the quality of collaborative content.

In general, since at least one feature from each view appeared in top 20 position, we can conclude that the combination of feature views are important to predict the article quality. Furthermore, we can also see in the top position of each rank that textual features have a similar importance when compared to Review features, specially Structure and Length features.

The Edit History is particularly useful for WIKIPEDIA. Besides some features, which we can observe in Table 5.4 among the top features, we observe *occasional users review rate* (*r-occasion*) which is important to identify which types of contributions an article is receiving. The number of discussions of an article (*r-discuss*), useful to infer the article importance and the engagement of the users in the article, was also among the top features. In addition we have the number of modified lines in a period of 3 months (*r-modline*), which is important to measure the stability of an article. In these collections, we can also observe some features regarding the structure of the article, in particular citations and sections (*ts-imgps*, *ts-citpsec*, *ts-avparl*, *ts-avsecl*, *ts-secs*, *ts-avsecl*). In smaller amounts, we also find features from all other views.

Structure, Style and Length features were the topmost features in STARWAR. An example of such features is the *passive voice count*, *word count* and *subsection* *count* (*ty-passive*, *tl-wordcnt*, *ts-subsec*). We also have some Review features namely the frequency of edits and whether the editor was a registered user.

In MUPPETS we can observe that fewer features were selected, when compared to the other collections. We can see that the age of the article (*r-age*) was selected in all solutions in the pool and features such as image per section, average paragraph length and number of registered users (*ts-imgps*, *ts-avparl*, *r-reguser*) were considered in many solutions in the collection. This highlights the importance of Edit History and Structure also in MUPPETS collection.

The differences observed in the selected features among the datasets can be attributed to different evaluation criteria; how much and fast such criteria are adopted; and the nature of topics and reviewers. Clearly, WIKIPEDIA is a well organized community with established evaluation procedures and well defined evaluation criteria. These criteria are strongly reinforced by the community and widely adopted by the reviewers. Also, the adopted procedures and engagement of the users imply that articles have to be strictly evaluated to be top rated, many articles are constantly reviewed, and many users become very proficient reviewers. As a result, many articles have richer review history and their quality is more related to how well they follow the established rules.

STARWAR and MUPPETS datasets, on the contrary, were created by a smaller community (with less reviewers than Wikipedia), adopt much more subjective evaluation criteria and do not reinforce its rules through strict procedures. For instance, STARWAR quality taxonomy has only three classes: Stub Class, Good Articles and Featured Articles. Since Stub articles are usually short draft articles, they are easily distinguishable from good and featured articles by its length. There is no significant difference among good and featured articles regarding how much they are reviewed or follow citations rules. As results, length, structure and history review features play different role in the two STARWAR and WIKIPEDIA. MUPPETS adopts a still more subjective evaluation procedure, based on a star-count criteria. In this evaluation, it is hard to distinguish if the votes are for quality or popularity. In general, popular characters receive more stars (and reviews) which is reflected by the importance of history features in Muppets. Apart from this, it is hard to point out the criteria used by the reviewers to distinguish articles from different quality classes.

In general, although we observe that the best selection of features is different depending on the dataset, we have similarities regarding the number of features of each view. For example, Edit History view is important in all collections, while Structure features are better in WIKIPEDIA and STARWAR than in MUPPETS, where we have more Style features in the top 20 features.

5.4 Discussion

In this chapter, we carried out a thorough analysis of the application of our multi-view approach to automatically assess quality of Wiki articles. With that, we were able to improve the results in two out of three tested collections. In addition, we also noted that collections using the star-based taxonomy were harder to classify than those using the Wikipedia-based taxonomy, which is probably due to the lack of criteria associated with their quality rating. By analyzing the performance per class and view, we found that the view performance can differ according to the quality class being assessed. For example, Length features are good predictors of low quality Wikipedia articles while Edit History and Structure view are better than Length in the highest quality. We further noted that our multi-view approach is better in instances when views agree with each other in the correct class (i.e. when they are correct) and disagree with each other in the instances hardest to predict.

We also performed a feature selection approach where we could reduce the feature set without losing performance. Through the analysis of the selected features, we observed, in general, the importance of combining features from different views and, in particular, that Structure and Edit History are good predictors of quality in Wiki datasets. We also presented the rank of importance of each feature.

Table 5.5. Ranking (from 1 to 10) of most common features in non-dominated solutions. Column '%' represents the relative frequency of the feature in the pool.

	Datasets					
Rank	WIKIPEDIA		STARWAR		MUPPETS	
	Feature	%	Feature	%	Feature	%
1	*r-stdrevu* (hist)	86	*ty-passive* (style)	64	*r-age* (hist)	100
2	*tl-charcnt* (length)	81	*ts-avparl* (struct)	61	*ts-imgps* (struct)	86
3	*ts-imgps* (struct)	67	*ty-prono* (style)	57	*ts-avparl* (struct)	43
4	*ts-citpsec* (struct)	62	*tl-wordcnt* (length)	57	*r-reguser* (hist)	36
5	*r-rperusr* (hist)	48	*tl-phrcnt* (length)	46	*ty-prepo* (style)	29
6	*r-occasion* (hist)	48	*ts-subsec* (struct)	46	*tl-phrcnt* (length)	29
7	*tl-phrcnt* (length)	48	*r-probrev* (hist)	43	*tl-wordcnt* (length)	29
8	*ts-avparl* (struct)	48	*ts-stdsecl* (struct)	43	*tr-ari* (read)	21
9	*ty-psmphr* (style)	43	*ts-abslen* (struct)	39	*ty-auxverb* (style)	21
10	*r-modline* (hist)	38	*ts-secs* (struct)	39	*ty-psmphr* (style)	21

Table 5.6. Ranking (from 11 to 50) of most common features in non-dominated solutions. Column '%' represents the relative frequency of the feature in the pool.

	Datasets					
Rank	WIKIPEDIA		STARWAR		MUPPETS	
	Feature	%	Feature	%	Feature	%
11	*ts-secs* (struct)	38	*r-stdrevu* (hist)	36	*n-pgrank* (graph)	21
12	*r-discuss* (hist)	33	*r-age* (hist)	32	*n-idegree* (graph)	21
13	*ty-nomina* (style)	33	*r-rperusr* (hist)	32	*r-ageprev* (hist)	14
14	*n-assortoi* (graph)	33	*r-rcount* (hist)	32	*r-rcount* (hist)	14
15	*ts-avsecl* (struct)	33	*r-reguser* (hist)	32	*r-activeu* (hist)	14
16	*r-reguser* (hist)	29	*tl-charcnt* (length)	32	*r-stdrevu* (hist)	14
17	*r-probrev* (hist)	29	*n-idegree* (graph)	32	*r-probrev* (hist)	14
18	*tr-flesh* (read)	29	*tr-ari* (read)	29	*tr-fog* (read)	14
19	*tr-smog* (read)	29	*tr-lix* (read)	29	*ty-passive* (style)	14
20	*n-reciproc* (graph)	29	*ts-avsubps* (struct)	29	*tr-smog* (read)	14
21	*r-activeu* (hist)	24	*r-ageprev* (hist)	25	*ty-ssubcnj* (style)	14
22	*ty-sprono* (style)	24	*r-activeu* (hist)	25	*n-assortoi* (graph)	14
23	*ty-conj* (style)	24	*ty-conj* (style)	25	*n-assortii* (graph)	14
24	*ty-prono* (style)	24	*ty-ssubcnj* (style)	25	*ts-citpsec* (struct)	14
25	*n-assortoo* (graph)	24	*n-linkcnt* (graph)	25	*n-linkcnt* (graph)	14
26	*n-odegree* (graph)	24	*n-translat* (graph)	25	*n-translat* (graph)	14
27	*n-translat* (graph)	24	*ts-avsecl* (struct)	25	*ts-abslen* (struct)	14
28	*r-anonym* (hist)	19	*r-modline* (hist)	21	*ts-maxsecl* (struct)	14
29	*r-rcount* (hist)	19	*ty-sartic* (style)	21	*ts-avsecl* (struct)	14
30	*tr-lix* (read)	19	*tr-fog* (read)	18	*ts-lnkpl* (struct)	14
31	*ty-questn* (style)	19	*tr-flesh* (read)	18	*r-discuss* (hist)	7
32	*ty-auxverb* (style)	19	*tr-kincaid* (read)	18	*r-3month* (hist)	7
33	*ty-sprepo* (style)	19	*ts-maxsecl* (struct)	18	*tr-liau* (read)	7
34	*n-idegree* (graph)	19	*ts-lnkpl* (struct)	18	*tr-flesh* (read)	7
35	*ts-cite* (struct)	19	*r-anonym* (hist)	14	*tr-kincaid* (read)	7
36	*r-revpday* (hist)	14	*tr-smog* (read)	14	*ty-sprono* (style)	7
37	*tr-liau* (read)	14	*ty-plgphr* (style)	14	*ty-conj* (style)	7
38	*ty-plgphr* (style)	14	*ty-questn* (style)	14	*ty-sartic* (style)	7
39	*ty-sconj* (style)	14	*ty-sintp* (style)	14	*ty-sintp* (style)	7
40	*ty-tobe* (style)	14	*ty-tobe* (style)	14	*ty-tobe* (style)	7
41	*n-pgrank* (graph)	14	*ts-citpsec* (struct)	14	*ty-sprepo* (style)	7
42	*n-cluster* (graph)	14	*ts-minsecl* (struct)	14	*ty-nomina* (style)	7
43	*n-linkcnt* (graph)	14	*ts-xlnkps* (struct)	14	*tl-charcnt* (length)	7
44	*ts-abslen* (struct)	14	*r-3month* (hist)	11	*ty-prono* (style)	7
45	*ts-subsec* (struct)	14	*tr-liau* (read)	11	*n-assortoo* (graph)	7
46	*ty-lgphra* (struct)	14	*ty-auxverb* (style)	11	*n-assortio* (graph)	7
47	*tr-ari* (read)	10	*ty-sprono* (style)	11	*n-odegree* (graph)	7
48	*tr-kincaid* (read)	10	*ty-prepo* (style)	11	*n-reciproc* (graph)	7
49	*ty-sartic* (style)	10	*ty-sprepo* (style)	11	*ts-cite* (struct)	7

Table 5.7. Ranking (from 50 to 68) of most common features in non-dominated solutions. Column '%' represents the relative frequency of the feature in the pool.

Rank	**Datasets**					
	WIKIPEDIA		STARWAR		MUPPETS	
	Feature	%	Feature	%	Feature	%
50	*ty-ssubcnj* (style)	10	*n-assortoo* (graph)	11	*ts-stdsecl* (struct)	0
51	*n-assortio* (graph)	10	*n-assortii* (graph)	11	*r-anonym* (hist)	0
52	*ts-citplen* (struct)	10	*n-cluster* (graph)	11	*ts-xlnks* (struct)	0
53	*ts-xlnks* (struct)	10	*ts-imgps* (struct)	11	*ty-lgphra* (struct)	0
54	*ts-minsecl* (struct)	10	*r-discuss* (hist)	7	*ts-subsec* (struct)	0
55	*ts-stdsecl* (struct)	10	*r-revpday* (hist)	7	*ts-minsecl* (struct)	0
56	*r-age* (hist)	5	*ty-sconj* (style)	7	*ty-plgphr* (style)	0
57	*r-ageprev* (hist)	5	*ty-nomina* (style)	7	*n-cluster* (graph)	0
58	*r-3month* (hist)	5	*n-assortoi* (graph)	7	*ts-citplen* (struct)	0
59	*tr-fog* (read)	5	*n-assortio* (graph)	7	*r-revpday* (hist)	0
60	*ty-passive* (style)	5	*n-pgrank* (graph)	7	*ts-secs* (struct)	0
61	*n-assortii* (graph)	5	*ty-lgphra* (struct)	7	*ts-xlnkps* (struct)	0
62	*ts-maxsecl* (struct)	5	*r-occasion* (hist)	4	*ty-sconj* (style)	0
63	*ts-xlnkps* (struct)	5	*ty-psmphr* (style)	4	*r-rperusr* (hist)	0
64	*ts-lnkpl* (struct)	5	*n-reciproc* (graph)	4	*tr-lix* (read)	0
65	*tl-wordcnt* (length)	0	*ts-citplen* (struct)	4	*r-modline* (hist)	0
66	*ty-sintp* (style)	0	*ts-cite* (struct)	4	*r-occasion* (hist)	0
67	*ty-prepo* (style)	0	*ts-xlnks* (struct)	4	*ty-questn* (style)	0
68	*ts-avsubps* (struct)	0	*n-odegree* (graph)	0	*ts-avsubps* (struct)	0

Chapter 6

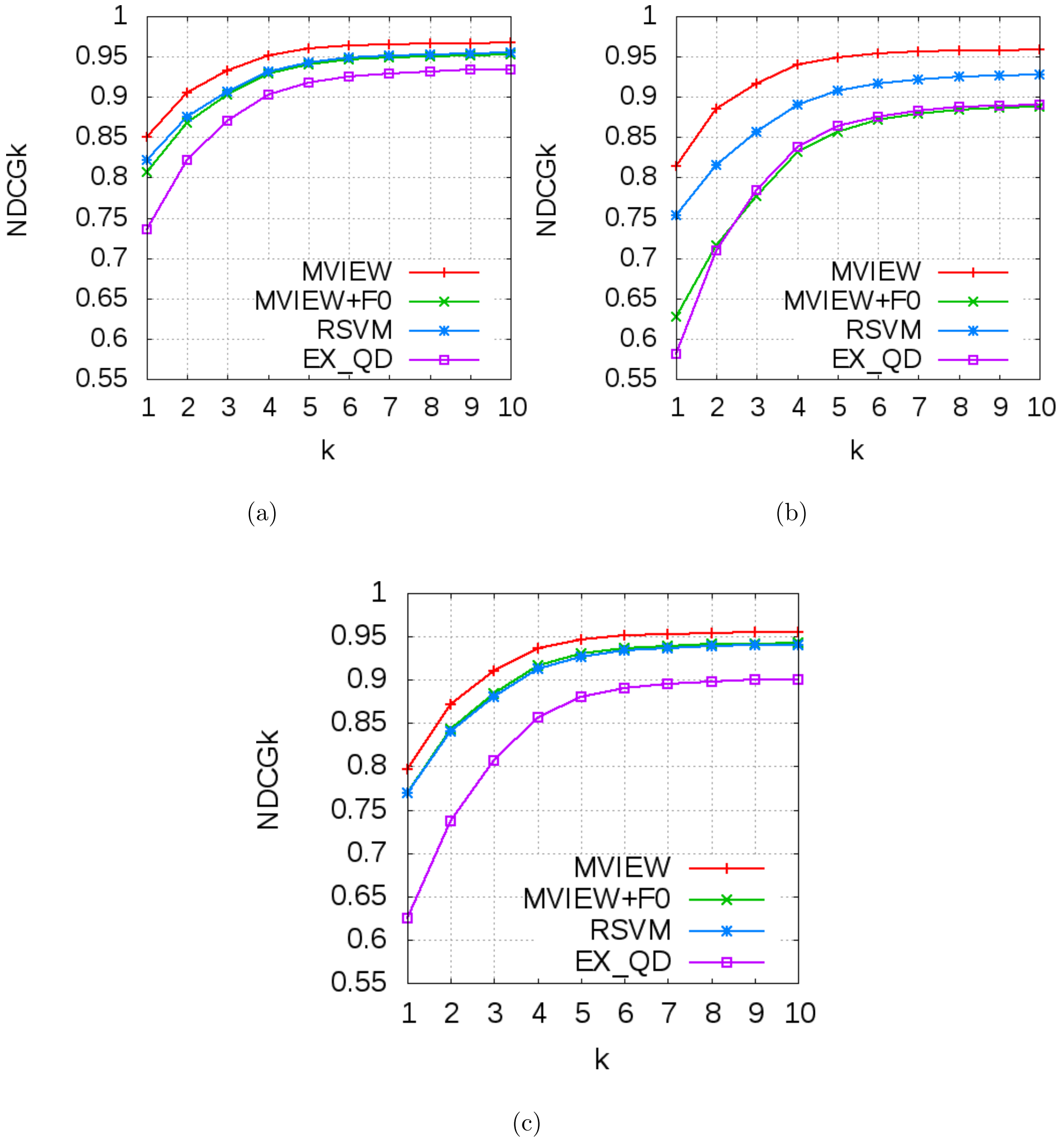
Results on Q&A Forum

Similar to the previous chapter, in this chapter we present our results and analysis regarding the Q&A Forum domain. To accomplish this, we first compared our approach with baselines. After that, we present our analysis regarding performance and correlation of the views. Finally, the results regarding the feature selection and analysis are presented.

6.1 Comparison with Baselines

We now evaluate the multi-view approach in the domain of Q&A Forums. As previously mentioned, in this domain, our aim is to rank answers according to quality views. We start by comparing the performance of our method with two baselines . The first is SVMRANK, a very well known learning-to-rank method based on SVM. Our implementation of SVMRANK uses the level-0 features described in Section 3.2 to learn the answers ranking. Our second baseline is the method proposed in Suryanto et al. [2009], explained in Section 2.4.5, which we refer to as EX_QD. As before, we use two variations of our multi-view approach. The first, which we call MVIEW, uses view predictions in level-1 learning. The second, which we call MVIEW+F0, also includes level-0 features in level-1 learning.

Figure 6.1 shows NDCG@k results obtained for our proposed methods and baselines in Q&A Forums test datasets. As we can see, MVIEW outperformed all the baselines in all datasets. Unlike in the Wiki domain, the MVIEW+F0 variant was never able to outperform MVIEW. In fact, it was even outperformed by SVMRANK in the COOK dataset. MVIEW reached statistically significant gains ($p < 0.05$) over EX_QD and SVMRANK in all datasets, considering all values used for k. The highest gains were obtained over SVMRANK in COOK with an increase of 8.1% in performance. In ENGLISH and STACK, the MVIEW approach had an improvement of 3.5% over SVMRANK. It should be noticed that some of the baselines are already strong in some of the tested datasets, which makes it very hard to obtain improvements. In any case, we were successful in our goal, obtaining significant gains over them, mainly in the initial portions of the rankings, an important property for this type of application.



(a)

(b)

(c)

Figure 6.1. Methods comparison using $NDCG@K$ in STACK (a), COOK (b), and ENGLISH (c).

6.2 Analysis of Views

We now analyze the proportion of instances in which each specific view was the best predictor. As in this domain instances correspond to questions, to estimate the performance of a view we use the average NDCG@k associated with all questions. In particular, for question q and view v, the NDCG@k for q is obtained considering the ranking of the answers given to q according to v. Table 6.1 shows the results of the percentage of q which has the best NDCG@k for a given view v.

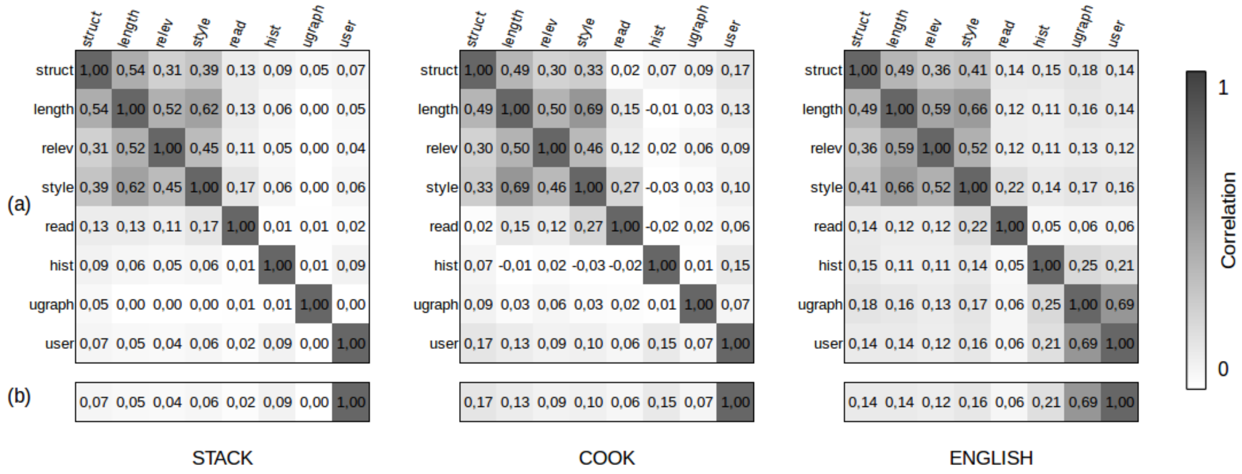
Table 6.1. Proportion of instances where each specific view provided the best estimate. Error is calculated assuming a 95% confidence interval and a Normal distribution. *Struct*, *Rel*, *Read* and *UGraph* are short versions for Structure, Relevance, Readability and User Graph.

Sample	Struct	Length	Rel	Style	Read	Ed. History	UGraph	User	Avg ± Error
STACK	8%	3%	24%	13%	7%	26%	13%	6%	12.5 ± 5.84
COOK	6%	4%	22%	12%	6%	30%	17%	3%	12.5 ± 6.75
ENGLISH	9%	3%	19%	11%	7%	29%	11%	10%	12.4 ± 5.60

As we can see, although some views may be very useful in different parts of the dataset, Length and Readability rarely provided good rankings in all datasets. Thus, unlike Wiki datasets, text Length is not a good indicator of quality in Q&A Forums domain. In this domain, Edit History is the best quality predictor probably because the more an answer is edited or commented, the better and more useful it is to the users. Another important view in this domain is Relevance.

Similarly to the correlation analysis that we have performed in the Wiki domain, Figure 6.2 presents the Kendall Tau rank correlation (cf. Section 4.2.2, Eq. 4.7) in the datasets STACK, COOK, and ENGLISH. In this figure there are two kinds of correlations: (a) between views and (b) between view and multi-view rankings. As before, the darker the color, the higher the correlation. Differently from the Wiki domain, however, the most important views in Q&A Forums domain (Edit History and Relevance) are not strongly correlated with each other, as we can see in Figure 6.2(a). In general, in the Q&A Forums datasets, the views are less correlated than in the Wiki datasets.

We can also note that the same general correlation patterns are observed in all datasets. As expected, text views are more correlated with each other than with other views. User Graph and User are correlated with each other only in ENGLISH. By inspecting the correlations between view and multi-view rankings in Figure 6.2(b), it is clear the importance of the Edit History view to MVIEW. Text views are also useful, specially for ENGLISH. The User view is more important in COOK and ENGLISH, while User Graph is useful only in ENGLISH. User and User Graph are also very



STACK

	struct	length	relev	style	read	hist	ugraph	user
struct	1,00	0,54	0,31	0,39	0,13	0,09	0,05	0,07
length	0,54	1,00	0,52	0,62	0,13	0,06	0,00	0,05
relev	0,31	0,52	1,00	0,45	0,11	0,05	0,00	0,04
style	0,39	0,62	0,45	1,00	0,17	0,06	0,00	0,06
read	0,13	0,13	0,11	0,17	1,00	0,01	0,01	0,02
hist	0,09	0,06	0,05	0,06	0,01	1,00	0,01	0,09
ugraph	0,05	0,00	0,00	0,00	0,01	0,01	1,00	0,00
user	0,07	0,05	0,04	0,06	0,02	0,09	0,00	1,00
(b)	0,07	0,05	0,04	0,06	0,02	0,09	0,00	1,00

COOK

	struct	length	relev	style	read	hist	ugraph	user
struct	1,00	0,49	0,30	0,33	0,02	0,07	0,09	0,17
length	0,49	1,00	0,50	0,69	0,15	-0,01	0,03	0,13
relev	0,30	0,50	1,00	0,46	0,12	0,02	0,06	0,09
style	0,33	0,69	0,46	1,00	0,27	-0,03	0,03	0,10
read	0,02	0,15	0,12	0,27	1,00	-0,02	0,02	0,06
hist	0,07	-0,01	0,02	-0,03	-0,02	1,00	0,01	0,15
ugraph	0,09	0,03	0,06	0,03	0,02	0,01	1,00	0,07
user	0,17	0,13	0,09	0,10	0,06	0,15	0,07	1,00
(b)	0,17	0,13	0,09	0,10	0,06	0,15	0,07	1,00

ENGLISH

	struct	length	relev	style	read	hist	ugraph	user
struct	1,00	0,49	0,36	0,41	0,14	0,15	0,18	0,14
length	0,49	1,00	0,59	0,66	0,12	0,11	0,16	0,14
relev	0,36	0,59	1,00	0,52	0,12	0,11	0,13	0,12
style	0,41	0,66	0,52	1,00	0,22	0,14	0,17	0,16
read	0,14	0,12	0,12	0,22	1,00	0,05	0,06	0,06
hist	0,15	0,11	0,11	0,14	0,05	1,00	0,25	0,21
ugraph	0,18	0,16	0,13	0,17	0,06	0,25	1,00	0,69
user	0,14	0,14	0,12	0,16	0,06	0,21	0,69	1,00
(b)	0,14	0,14	0,12	0,16	0,06	0,21	0,69	1,00

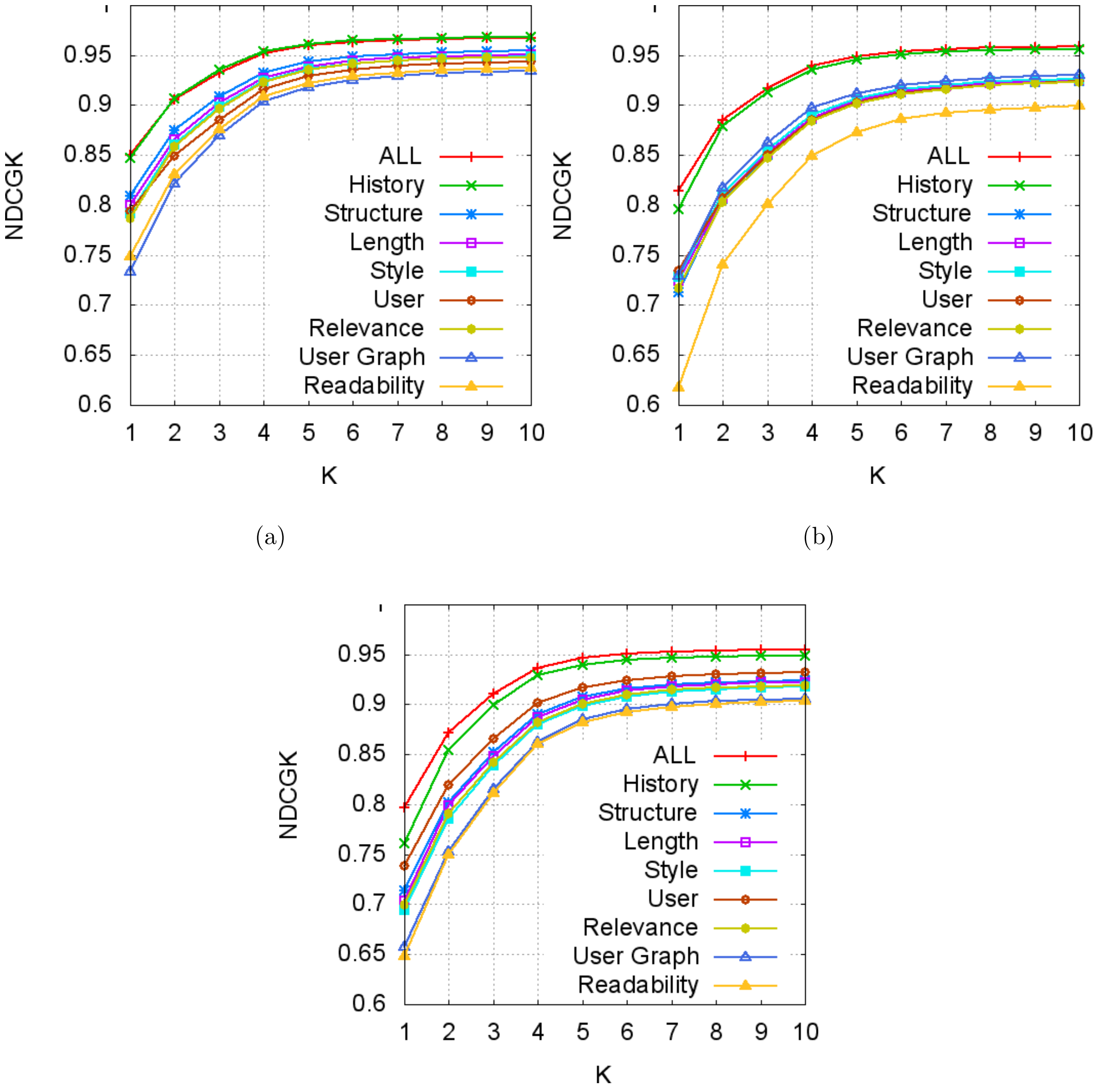
Figure 6.2. View correlation in datasets STACK, COOK, and ENGLISH: (a) correlations between views and (b) correlations between view and multi-view. In this figure, the darker the color, the higher the correlation (all negative correlations observed were near zero). Labels correspond to views (left to right and top to bottom) structure, Length, Relevance, Style, Readability, Edit History, User Graph, and User, respectively.

correlated in ENGLISH, making them as important as User Features. In the others collections the User Graph features were not so important. Finally, in STACK, MVIEW does not seem to take User and User Graph into account in its estimates, as suggested by their extremely low correlation.

Figure 6.3 compares the NDCG@k performance of MVIEW using features of all views (ALL) and individual views. As expected, Edit History is the most important view. Considering only statistically significant results, Edit History is outperformed by ALL for all k values in ENGLISH. In COOK, it is outperformed in the first position of the rank, i.e., the "best" answer. In STACK, Edit History reaches the same performance of all views combined. Such results show that the multi-view approach is able to preserve the good performance of a single dominant view, if such view does exist in a given dataset, while improving its performance in other cases (e.g., ENGLISH and COOK.). That was not the case for the baseline methods SVMRANK and EX_QD which performed worse than Edit History (taken in isolation) in all datasets.

Among the textual features, Structure was the best predictor in ENGLISH and STACK collections. In COOK, except by for the best feature (Edit History) and the worst view (Readability), all the remaining views reached similar performance. This includes User Graph, which performed poorly in ENGLISH and STACK.

By observing Figure 6.2, we note that, as in the Wiki domain, our multi-view approach is more correlated with the best predictors. For instance, In ENGLISH,



(a)

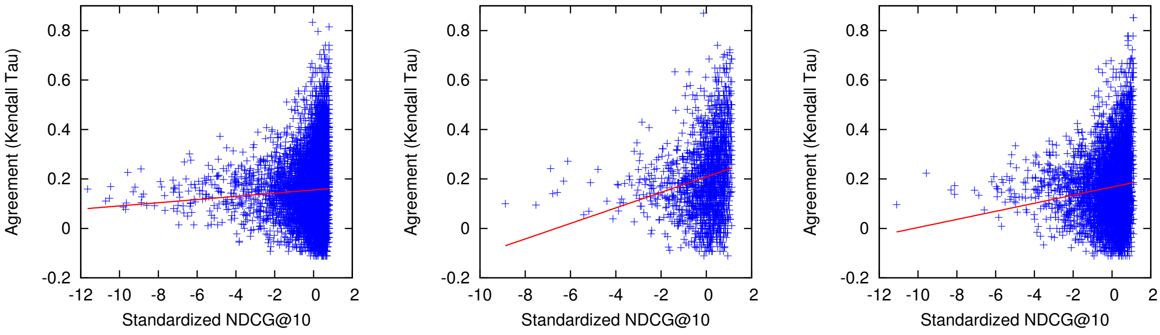
(b)

(c)

Figure 6.3. Performance of MVIEW compared to individual views using NDCG@k in datasets STACK (a), COOK (b), and ENGLISH (c).

the high correlations of multi-view with Edit History, User and User Graph paid off with significant gains over all baselines. In COOK, it was able to obtain statistically significant improvements in NDCG@1, corresponding to the "best answer", by also exploiting some correlations, mainly with Edit History and User Graph. Only in STACK, MVIEW was not able to obtain complementary information from other views and simply achieved the same performance as Edit History in isolation. To summarize, MVIEW was either capable of selecting the best single view or to combine it with other views when these contained complementary and useful information.

Finally, in Figure 6.4, we observe the relation between view ranking agreement and ranking performance measured by NDCG@10. In this figure, each point corresponds to the set of answers given to a query. Each instance is represented by the ranking agreement and the ranking performance amongst its views. We calculated the agreement using the Kendal Tau correlation metric. The ranking quality was evaluated as the average NDCG@10 of the view rankings. The NDCG@10 values were standardized to have zero mean and unit variance. To facilitate the visualization of the point distribution, we plot the regression line that best fits the data points. In this case, we did not split the graph at the center of mass of the points, as results were too concentrated in only one side of the Figure, making it hard to make visual conclusions.



(a) (b) (c)

Figure 6.4. Kendall tau correlation between view questions and its $NDCG@K$ in the collections STACK (a), COOK (b), and ENGLISH (c).

As we can see, for most queries in the three collections, the views present weak to moderate correlations. Clearly, the higher the correlation (even if reverse), the better the ranking. The three collections present very similar distributions with COOK showing a slightly higher trend towards the combination between better rankings and higher correlations.

6.3 Analysis of Features within Views

In this Section we discuss the impact of the features in Q&A Forums. As before, we used SPEA2 as feature selector. Figure 6.5 shows non-dominated solutions for a sample of the COOK dataset. The analysis in this section consists in observing feature patterns in such non-dominated solutions.

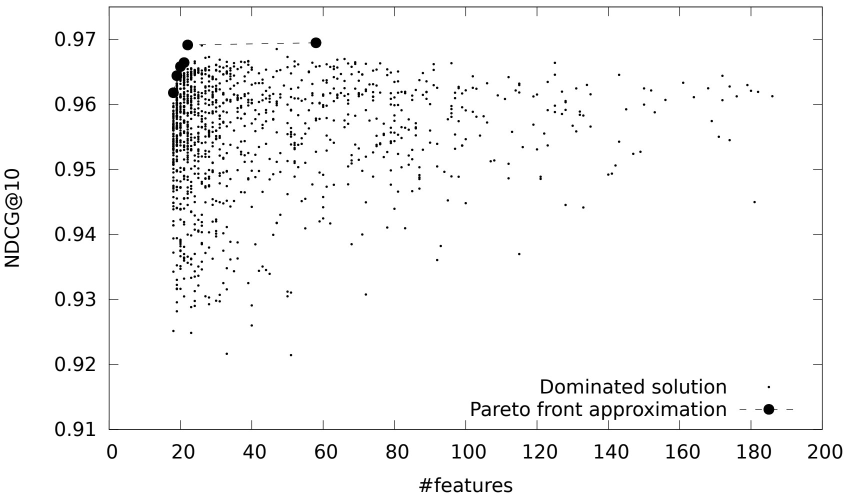


Figure 6.5. Approximation of Pareto front for a COOK sample. Larger dots represent non-dominated solutions.

We start by comparing the baseline methods using all features with the best non-dominated solutions found by SPEA2 in each dataset. Table 6.2 shows the NDCG@10 scores obtained by RSVM, MVIEW, GA, and GA+F0. GA is the version of SPEA2 which uses only the level-1 representation (views) while GA+F0 also includes level-0 features. Results marked with "*" are significantly better than MVIEW and those marked with "†" are significantly worse than it. As we can see, using around 20% of features, we were able to reach the same performance of our multi-view approach. Moreover, for STACK, we were able to significantly improve the effectiveness.

Table 6.2. NDCG@10 values for collections STACK, COOK, and ENGLISH. Column '% of feat.' represents the percentage of features used (an average for GA and GA-F because values were obtained using cross-validation).

View	**Method per collection**					
	STACK		COOK		ENGLISH	
	NDCG@10	% of feat.	NDCG@10	% of feat.	NDCG@10	% of feat.
RSVM	0.955†	100%	0.940†	100%	0.927†	100%
MVIEW	0.967	100%	0.955	100%	0.959	100%
GA	0.976*	19.2%	0.957	19.56%	0.958	23.5%
GA+F0	0.972	19.2%	0.953	19.56%	0.942	23.5%

Table 6.3 shows the best representation for an arbitrary fold of each collection. As we can see in Table 6.3, no view was discarded, as each one was represented by at least one feature (except by Readability view in ENGLISH).

In the example of Table 6.3, we can see that some redundant features were removed. For example, the Length view in STACK was represented only by *tl-phrcnt*,

Table 6.3. Features selected using SPEA2 for an arbitrary fold.

	Features per collection		
View	**STACK**	**COOK**	**ENGLISH**
Structure	*ts-xlnks*; *ts-inlink*; *ts-codes*; *ts-listit*; *ts-mincod*; *ts-stdcod*; *ts-mxquot*; *ts-mnquot*; *ts-maxsecl*; *ts-minsecl*; *ts-avsecl*; *ts-usrref*; *ts-boldit*; *ts-parcnt*	*ts-avsubs*; *ts-maxcod*; *ts-mnquot*; *ts-stdquot*; *ts-avsecl*; *ts-parcnt*	*ts-maxcod*; *ts-stdcod*; *ts-stdquot*; *ts-boldit*
Length	*tl-phrcnt*	*tl-wordcnt*; *tl-phrcnt*	–
Relevance	*tm-aspan*$_b$; *tm-nwnout*$_t$; *tm-phmch*$_{t,b}$; *tm-phmch*$_{b,bg}$; *tm-phmch*$_{b,dg}$; *tm-bm25*$_{t,b}$; *tm-bm25*$_{b,w}$; *tm-bm25*$_{b,d}$; *tm-wmch*$_{t,d}$; *tm-wmtch*$_{b,dg}$	*tm-aspan*$_t$; *tm-phmch*$_{b,w}$; *tm-worder*$_b$; *tm-nwaad*$_t$; *tm-nwver*$_b$; *tm-nwadj*$_b$; *tm-phmch*$_{b,dg}$; *tm-bm25*$_{t,w}$; *tm-bm25*$_{t,bg}$; *tm-bm25*$_{t,d}$; *tm-bm25*$_{t,dg}$; *tm-bm25*$_{b,b}$; *tm-bm25*$_{b,dg}$; *tm-wmtch*$_{b,d}$; *tm-wmtch*$_{b,dg}$	*tm-aspan*$_t$; *tm-phmch*$_{b,w}$; *tm-nwnout*$_t$; *tm-phmch*$_{b,bg}$; *tm-bm25*$_{t,d}$; *tm-bm25*$_{b,w}$; *tm-wmtch*$_{b,d}$
Style	*ty-psmphr*; *ty-passive*; *ty-lgphra*; *ty-auxverb*; *ty-prono*; *ty-nomina*; *ty-sprono*; *ty-dotden*; *ty-capwrd*; *ty-dotcnt*; *ty-infnois*; *ty-kldqa*; *ty-kldtag*; *ty-notwn*; *ty-typo*	*ty-passive*; *ty-prono*; *ty-sprono*; *ty-spaden*; *ty-infnois*; *ty-kldqa*; *ty-klddis*; *ty-kldwiki*	*ty-auxverb*; *ty-conj*; *ty-prepo*; *ty-sprono*; *ty-spaden*; *ty-caperr*; *ty-dotcnt*; *ty-wrenpy*; *ty-infnois*; *ty-notwn*
Readability	*tr-ari*; *tr-lix*	*tr-ari*; *tr-smog*	–
Ed. History	*r-ansage*; *r-rcount*; *r-comans*; *r-qsuged*; *r-qrejed*; *r-usredt*; *r-comque*; *r-usrcom*	*r-queage*; *r-avedusr*; *r-aaped*; *r-arejed*; *r-usrcom*; *r-ansbef*	*r-ansage*; *r-queage*; *r-aaped*; *r-comans*; *r-ansbef*
User Graph	*ug-exprank*	*ug-exprank*; *ug-hub*	*ug-prank*; *ug-exprank*
User	*u-avansq*; *u-avatag*; *u-avqtag*; *u-avratag*; *u-answrs*; *u-commnt*; *u-prtp3an*; *u-mxratag*; *u-mxatag*; *u-mncomq*; *u-mnqtag*; *u-prtp3qu*; *u-enpytag*; *u-anpytag*; *u-xrktq*; *u-mrateat*; *u-rateans*; *u-xrateat*; *u-mrkta*; *u-rrateq*	*u-avansq*; *u-avcoma*; *u-avrqtag*; *u-quests*; *u-prtp3qu*; *u-maxansq*; *u-mxcomq*; *u-mxratag*; *u-mnqtag*; *u-mnrqtag*; *u-prtp3an*; *u-rkans*; *u-rkqust*; *u-xrktq*; *u-arateat*; *u-mrateat*; *u-xrateqt*	*u-avatag*; *u-answrs*; *u-apsuged*; *u-daycrt*; *u-maxansq*; *u-mxqtag*; *u-mxratag*; *u-mnrqtag*; *u-prtp3qu*; *u-rkans*; *u-enpytag*; *u-xrktq*; *u-rateque*; *u-rratea*; *u-xrateat*; *u-srateqt*; *u-mrkta*

while in the ENGLISH dataset that and the Readability view were completely absent. In fact, most features from the Readability view were discarded in almost all datasets, which would be expected since they are very similar to each other and performed poorly (cf. Section 6.2). SPEA2 retained more features from User view than from the remaining, which only highlights the importance of user reputation for the quality of an answer.

Regarding the Structure view, the STACK dataset retained the most diversified features. They include information about links in the text (*ts-xlnks*, *ts-inlink*, *ts-usrref*); structure style such as paragraphs, font formating, lists and sections (*ts-parcnt*, *ts-listit*, *ts-boldit*, *ts-maxsecl*, *ts-minsecl*, *ts-avsecl*).We can see also that quotations were used both in the COOK and ENGLISH collection.

Style features were largely used in all datasets. Again, the STACK dataset was very diversified, with features from word usage (*ty-passive*, *ty-prono*, *ty-auxverb*, *ty-*

nomina), use of capital letters (*ty-capwrd*), vocabulary (*ty-kldqa*, *ty-kldtag*), and typos (*ty-typo*). This diversity may have contributed to the GA performance improvement in STACK. Although the features were also diverse in COOK and ENGLISH datasets, they were less frequent. COOK was the only dataset that did not have any typo features.

Regarding the Edit History view, we observed features related to user engagement in all datasets. That is the case of *r-rcount*, *r-aaped*, *r-arejed* and *r-comans.* The same was observed for features used to infer the answer/question evolution and maturity, namely *r-ansage*, *r-queage*, and *r-ansbef.*

At most 50% of User Graph features were used in all collections for the analyzed fold. Since they are very similar to each other, in STACK only *ug-exprank* was used, while in ENGLISH and COOK only two features that capture similar aspects of quality (the reputation of the answer author) were used namely, user Expertise Rank and user Hub (*ug-exprank*, *ug-hub*).

In Table 6.4, 6.5 and 6.6 we present the features used in the non-dominated solutions, sorted by the frequency of their use. As in the Wiki domain, this ranking was built using all nondominated solutions from all folds, totalizing 17 STACK solutions, 28 COOK solutions, and 33 ENGLISH solutions, respectively.

In general, we can see that the most used features were from the Edit History and User views. While the first view is good to infer the usefulness and engagement of the users, the latter is a good estimator of reputation. There are some common features among the datasets. For example, the length feature *tl-phrcnt* was important to STACK and COOK. The feature *r-usrcom*, which measures the engagement of users in one answer, appeared as the most common feature in STACK and COOK. As expected, the Structure feature number of codes (*ts-codes*) was important just in the STACK dataset while a quotation related feature (*ts-mnquot*) was important in the ENGLISH dataset. In general, quotations are certainly more important in answers regarding language usage than in forums about cooking or programming. Relevance features were used specially in the COOK dataset. Regarding Relevance features, we can also see that the title representation of the question was more used in the best solutions. Only one style feature appears in the top 20 in STACK (sentences that start with preposition, *ty-sprepo*) while several appear in COOK (*ty-passive*, *ty-sintp*, *ty-sprepo*) and ENGLISH (*tm-nwvebt*$_t$, *ty-prepo*).

6.4 Discussion

In this chapter we presented the results regarding our multi-view approach for ranking answers in Q&A Forums. In particular, we adopted an approach based on the L2R method SVMRank and represented the Q&A pairs using eight views: Review, User, User Graph, Structure, Style, Length, Readability, and Relevance. In total, we evaluated 186 features and, to the best of our knowledge, 89 of them were not used in the Q&A Forums domain before. Our approach was trained to learn the answer rating, based on the feedback users give to answers in three different Q&A Forums.

We show that our multi-view approach was able to outperform our baseline with statistically significant gains of up to 8.1% in NDCG@k. Through correlation and performance analysis, we observed that our approach was either capable of selecting the best single view or to combine them when they contained complementary information.

By carrying out a feature selection, we were able to reduce the number of features without denigrating the performance and also eliminating redundant features. We also show that, in STACK, we could significantly improve the result. By analyzing the non-dominated individuals, we found that, similarly what was previously observed for Wikis, Review features are the most important in the Q&A Forums domain and Readability, the worst. User features are also important for that task.

Table 6.4. Most common features in non-dominated solutions. Column '%' represents the relative frequency of the feature in the pool.

	Datasets					
Rank	STACK		COOK		ENGLISH	
	Feature	%	Feature	%	Feature	%
1	*r-usrcom* (hist)	100	*r-usrcom* (hist)	93	*r-ansbef* (hist)	100
2	*u-xrktq* (user)	100	*r-ansbef* (hist)	71	*r-ansage* (hist)	73
3	*u-srateqt* (user)	71	*r-ansage* (hist)	68	*r-avedusr* (hist)	64
4	*u-enpytag* (user)	65	*tl-phrcnt* (length)	50	*u-rratea* (user)	61
5	*r-usredt* (hist)	47	*ug-auth* (ugraph)	43	*u-rkans* (user)	58
6	*u-rratea* (user)	47	*r-uniqusr* (hist)	43	*r-usrcom* (hist)	55
7	*tl-phrcnt* (length)	41	*u-xrateat* (user)	43	*ty-tobe* (style)	48
8	*u-rrateq* (user)	41	*u-avcoma* (user)	39	*ts-parcnt* (structure)	45
9	*ty-sprepo* (style)	35	*ug-prank* (ugraph)	32	*u-badges* (user)	42
10	*ts-ssssec* (structure)	35	*u-mncoma* (user)	32	*r-aaped* (hist)	39
11	*ts-parcnt* (structure)	35	*u-rratea* (user)	32	*ts-mnquot* (structure)	33
12	*u-anpytag* (user)	35	*tm-aspan*$_t$ (relev)	29	*u-avatag* (user)	33
13	*tm-aspan*$_t$ (relev)	29	*u-rrateq* (user)	29	*u-answrs* (user)	33
14	*r-rcount* (hist)	29	*tm-phmch*$_{b,dg}$ (relev)	25	*tm-nwvebt*$_t$ (relev)	30
15	*r-queans* (hist)	29	*tm-bm25*$_{t,dg}$ (relev)	25	*ty-prepo* (style)	30
16	*ts-codes* (structure)	29	*tm-wmch*$_{t,b}$ (relev)	25	*r-rcount* (hist)	30
17	*u-avatag* (user)	29	*ty-passive* (style)	25	*r-usredt* (hist)	30
18	*u-avcomq* (user)	29	*ty-sintp* (style)	25	*u-top3qu* (user)	30
19	*u-mxatag* (user)	29	*ty-sprepo* (style)	25	*u-xrateat* (user)	30
20	*u-mxratag* (user)	29	*r-arejed* (hist)	25	*tm-phmch*$_{t,d}$ (relev)	27
21	*u-xrateqt* (user)	29	*r-comans* (hist)	25	*tl-phrcnt* (length)	27
22	*tm-bm25*$_{b,d}$ (relev)	24	*tr-fog* (read)	25	*ts-boldit* (structure)	27
23	*ty-capwrd* (style)	24	*ts-maxsecl* (structure)	25	*u-avcoma* (user)	27
24	*ty-typo* (style)	24	*ts-avsecl* (structure)	25	*u-mxatag* (user)	27
25	*ug-exprank* (ugraph)	24	*u-mrateat* (user)	25	*u-rateque* (user)	27
26	*r-ansage* (hist)	24	*u-prtp3an* (user)	25	*u-mrkta* (user)	27
27	*ts-mincod* (structure)	24	*tm-phmch*$_{t,b}$ (relev)	21	*tm-wmch*$_{t,dg}$ (relev)	24
28	*ts-maxsecl* (structure)	24	*tm-bm25*$_{t,bg}$ (relev)	21	*ty-sartic* (style)	24
29	*ts-boldit* (structure)	24	*tm-bm25*$_{b,bg}$ (relev)	21	*ug-prank* (ugraph)	24
30	*ts-avsecl* (structure)	24	*ug-exprank* (ugraph)	21	*r-comans* (hist)	24
31	*u-maxansq* (user)	24	*ts-minsecl* (structure)	21	*u-mrktq* (user)	24
32	*u-srateat* (user)	24	*u-quests* (user)	21	*u-enpytag* (user)	24
33	*tm-worder*$_t$ (relev)	18	*u-answrs* (user)	21	*u-srateqt* (user)	24
34	*tm-phmch*$_{t,bg}$ (relev)	18	*u-minansq* (user)	21	*u-arktq* (user)	24
35	*tm-phmch*$_{b,d}$ (relev)	18	*u-mrktq* (user)	21	*tm-bm25*$_{t,bg}$ (relev)	21
36	*tm-wmch*$_{t,d}$ (relev)	18	*tm-worder*$_b$ (relev)	18	*ty-ssubcnj* (style)	21
37	*tm-wmch*$_{t,dg}$ (relev)	18	*tm-nwnout*$_t$ (relev)	18	*ty-sprepo* (style)	21
38	*ty-passive* (style)	18	*tm-nwadj*$_b$ (relev)	18	*ty-kldwiki* (style)	21
39	*ty-tobe* (style)	18	*ty-tobe* (style)	18	*r-comque* (hist)	21
40	*ty-sprono* (style)	18	*ty-sprono* (style)	18	*ts-maxsecl* (structure)	21
41	*ty-prepo* (style)	18	*r-qrejed* (hist)	18	*u-mxratag* (user)	21
42	*ty-prono* (style)	18	*r-aaped* (hist)	18	*tm-aspan*$_b$ (relev)	18
43	*ty-spaden* (style)	18	*tr-smog* (read)	18	*tm-phmch*$_{t,bg}$ (relev)	18
44	*ty-dotcnt* (style)	18	*ts-mincod* (structure)	18	*tm-bm25*$_{b,dg}$ (relev)	18
45	*ty-kldqa* (style)	18	*ts-parcnt* (structure)	18	*ty-wrenpy* (style)	18
46	*r-aaped* (hist)	18	*u-sugedt* (user)	18	*ty-dotcnt* (style)	18
47	*r-comans* (hist)	18	*u-top3qu* (user)	18	*ty-caperr* (style)	18
48	*tr-ari* (read)	18	*u-top3an* (user)	18	*r-stdpusr* (hist)	18
49	*ts-avsubs* (structure)	18	*u-avqtag* (user)	18	*r-queans* (hist)	18
50	*tr-smog* (read)	18	*u-srateat* (user)	18	*tr-smog* (read)	18
51	*ts-xlnks* (structure)	18	*u-xrktq* (user)	18	*ts-stdcod* (structure)	18
52	*ts-minsecl* (structure)	18	*u-rkqust* (user)	18	*ts-secs* (structure)	18
53	*ts-stdquot* (structure)	18	*u-srateqt* (user)	18	*tl-wordcnt* (length)	18
54	*u-badges* (user)	18	*tm-phmch*$_{t,w}$ (relev)	14	*u-top3an* (user)	18
55	*u-mncomq* (user)	18	*tm-nwvebt*$_t$ (relev)	14	*u-avrqtag* (user)	18
56	*u-mrateat* (user)	18	*tm-phmch*$_{b,d}$ (relev)	14	*u-srateat* (user)	18
57	*u-arateqt* (user)	18	*tm-bm25*$_{t,w}$ (relev)	14	*u-mrateat* (user)	18
58	*u-mrkta* (user)	18	*tm-bm25*$_{b,dg}$ (relev)	14	*u-arateat* (user)	18
59	*tm-aspan*$_b$ (relev)	12	*tm-wmtch*$_{b,bg}$ (relev)	14	*u-rrateq* (user)	18
60	*tm-phmch*$_{t,w}$ (relev)	12	*ty-lgphra* (style)	14	*tm-worder*$_t$ (relev)	15
61	*tm-nwnout*$_t$ (relev)	12	*ty-prono* (style)	14	*tm-nwnout*$_t$ (relev)	15
62	*tm-nwnou*$_b$ (relev)	12	*ty-sconj* (style)	14	*tm-nwadj*$_b$ (relev)	15
63	*tm-wmcht,w* (relev)	12	*ty-wrenpy* (style)	14	*tm-phmch*$_{b,bg}$ (relev)	15
64	*tm-phmch*$_{b,bg}$ (relev)	12	*ty-caperr* (style)	14	*tm-phmch*$_{b,d}$ (relev)	15
65	*tm-phmch*$_{b,dg}$ (relev)	12	*ty-kldqa* (style)	14	*ty-psmphr* (style)	15

Table 6.5. Most common features in non-dominated solutions. Column '%' represents the relative frequency of the feature in the pool.

	Datasets					
Rank	STACK		COOK		ENGLISH	
	Feature	%	Feature	%	Feature	%
66	*tm-bm25*$_{t,b}$ (relev)	12	*ty-kldwiki* (style)	14	*ty-notwn* (style)	15
67	*tm-bm25*$_{t,d}$ (relev)	12	*r-qaped* (hist)	14	*ug-exprank* (ugraph)	15
68	*tm-bm25*$_{b,w}$ (relev)	12	*tr-kincaid* (read)	14	*r-queage* (hist)	15
69	*tm-bm25*$_{t,dg}$ (relev)	12	*r-queans* (hist)	14	*r-qaped* (hist)	15
70	*tm-bm25*$_{b,bg}$ (relev)	12	*ts-stdquot* (structure)	14	*tr-flesh* (read)	15
71	*tm-bm25*$_{b,dg}$ (relev)	12	*ts-mnquot* (structure)	14	*ts-avsubs* (structure)	15
72	*tm-wmtch*$_{b,dg}$ (relev)	12	*tl-wordcnt* (length)	14	*ts-ssssec* (structure)	15
73	*ty-psmphr* (style)	12	*u-edits* (user)	14	*ts-stdquot* (structure)	15
74	*ty-auxverb* (style)	12	*u-avrqtag* (user)	14	*ts-avsecl* (structure)	15
75	*ty-lgphra* (style)	12	*u-solvqu* (user)	14	*u-sugedt* (user)	15
76	*ty-nomina* (style)	12	*u-arktq* (user)	14	*u-rjsuged* (user)	15
77	*ty-sconj* (style)	12	*tm-nwaad*$_{t}$ (relev)	11	*u-mncoma* (user)	15
78	*ty-ssubcnj* (style)	12	*tm-nwver*$_{b}$ (relev)	11	*u-anpytag* (user)	15
79	*ty-infnois* (style)	12	*tm-wmcht,w* (relev)	11	*u-arateqt* (user)	15
80	*ty-caperr* (style)	12	*tm-phmch*$_{b,b}$ (relev)	11	*tm-nwaad*$_{t}$ (relev)	12
81	*ty-notwn* (style)	12	*tm-phmch*$_{t,d}$ (relev)	11	*tm-bm25*$_{t,w}$ (relev)	12
82	*ty-kldtag* (style)	12	*tm-phmch*$_{t,dg}$ (relev)	11	*tm-bm25*$_{b,bg}$ (relev)	12
83	*r-avedusr* (hist)	12	*tm-bm25*$_{t,b}$ (relev)	11	*tm-wmtch*$_{b,d}$ (relev)	12
84	*ug-auth* (ugraph)	12	*tm-bm25*$_{b,d}$ (relev)	11	*ty-sprono* (style)	12
85	*r-arejed* (hist)	12	*tm-wmch*$_{t,dg}$ (relev)	11	*ty-prono* (style)	12
86	*r-comque* (hist)	12	*ty-psmphr* (style)	11	*ty-capwrd* (style)	12
87	*r-qsuged* (hist)	12	*ty-spaden* (style)	11	*ty-dotden* (style)	12
88	*tr-lix* (read)	12	*ty-notwn* (style)	11	*ty-typo* (style)	12
89	*ts-list* (structure)	12	*ty-kldtag* (style)	11	*ty-kldqa* (style)	12
90	*ts-mnquot* (structure)	12	*ty-klddis* (style)	11	*r-arejed* (hist)	12
91	*tl-wordcnt* (length)	12	*r-avedusr* (hist)	11	*r-qrejed* (hist)	12
92	*ts-usrref* (structure)	12	*r-rcount* (hist)	11	*ts-codes* (structure)	12
93	*u-quests* (user)	12	*tr-ari* (read)	11	*ts-mincod* (structure)	12
94	*u-edits* (user)	12	*ts-subsec* (structure)	11	*ts-stdsecl* (structure)	12
95	*u-rjsuged* (user)	12	*ts-stdsecl* (structure)	11	*ts-minsecl* (structure)	12
96	*u-avratag* (user)	12	*tl-charcnt* (length)	11	*ts-avquot* (structure)	12
97	*u-answrs* (user)	12	*u-commnt* (user)	11	*u-commnt* (user)	12
98	*u-minansq* (user)	12	*u-badges* (user)	11	*u-edits* (user)	12
99	*u-mxatag* (user)	12	*u-rjsuged* (user)	11	*u-mxrqtag* (user)	12
100	*u-prtp3qu* (user)	12	*u-avatag* (user)	11	*u-minansq* (user)	12
101	*u-xrateat* (user)	12	*u-mnqtag* (user)	11	*u-daycrt* (user)	12
102	*u-rateans* (user)	12	*u-mnrqtag* (user)	11	*u-lastac* (user)	12
103	*tm-worder*$_{b}$ (relev)	6	*u-mxratag* (user)	11	*u-xrktq* (user)	12
104	*tm-nwvebt*$_{t}$ (relev)	6	*u-arateat* (user)	11	*u-xrateqt* (user)	12
105	*tm-nwadj*$_{b}$ (relev)	6	*u-arateqt* (user)	11	*tm-phmch*$_{t,w}$ (relev)	9
106	*tm-phmch*$_{t,b}$ (relev)	6	*u-rateans* (user)	11	*tm-worder*$_{b}$ (relev)	9
107	*tm-phmch*$_{t,d}$ (relev)	6	*u-mrkta* (user)	11	*tm-phmch*$_{t,dg}$ (relev)	9
108	*tm-phmch*$_{t,dg}$ (relev)	6	*tm-aspan*$_{b}$ (relev)	7	*tm-wmtch*$_{b,bg}$ (relev)	9
109	*tm-bm25*$_{t,w}$ (relev)	6	*tm-phmch*$_{b,w}$ (relev)	7	*tm-wmch*$_{t,b}$ (relev)	9
110	*tm-bm25*$_{t,bg}$ (relev)	6	*tm-bm25*$_{t,d}$ (relev)	7	*tm-wmtch*$_{b,b}$ (relev)	9
111	*tm-bm25*$_{b,b}$ (relev)	6	*tm-bm25*$_{b,w}$ (relev)	7	*tm-wmtch*$_{b,dg}$ (relev)	9
112	*tm-wmtch*$_{b,d}$ (relev)	6	*tm-bm25*$_{b,b}$ (relev)	7	*ty-lgphra* (style)	9
113	*tm-wmch*$_{t,bg}$ (relev)	6	*tm-wmtch*$_{b,dg}$ (relev)	7	*ty-sintp* (style)	9
114	*tm-wmtch*$_{b,b}$ (relev)	6	*ty-questn* (style)	7	*ty-spaden* (style)	9
115	*ty-questn* (style)	6	*ty-auxverb* (style)	7	*r-asuged* (hist)	9
116	*ty-plgphr* (style)	6	*ty-nomina* (style)	7	*tr-liau* (read)	9
117	*ty-conj* (style)	6	*ty-ssubcnj* (style)	7	*tr-fog* (read)	9
118	*ty-dotden* (style)	6	*ty-infnois* (style)	7	*ts-avgcod* (structure)	9
119	*ty-wrenpy* (style)	6	*ty-typo* (style)	7	*ts-usrref* (structure)	9
120	*ug-prank* (ugraph)	6	*r-queage* (hist)	7	*u-avqtag* (user)	9
121	*ty-klddis* (style)	6	*r-usredt* (hist)	7	*u-avcomq* (user)	9
122	*ty-kldwiki* (style)	6	*ts-avsubs* (structure)	7	*u-mnratag* (user)	9
123	*r-stdpusr* (hist)	6	*ts-xlnks* (structure)	7	*u-mxcoma* (user)	9
124	*r-asuged* (hist)	6	*ts-usrref* (structure)	7	*u-rkqust* (user)	9
125	*r-qrejed* (hist)	6	*u-avratag* (user)	7	*u-rateans* (user)	9
126	*r-qaped* (hist)	6	*u-mxrqtag* (user)	7	*tm-aspan*$_{t}$ (relev)	6
127	*tr-liau* (read)	6	*u-mncomq* (user)	7	*tm-phmch*$_{b,w}$ (relev)	6
128	*r-ansbef* (hist)	6	*u-mnratag* (user)	7	*tm-wmcht,w* (relev)	6
129	*ts-quotes* (structure)	6	*u-maxansq* (user)	7	*tm-phmch*$_{t,b}$ (relev)	6
130	*ts-inlink* (structure)	6	*u-mxcomq* (user)	7	*tm-bm25*$_{t,d}$ (relev)	6

Table 6.6. Most common features in non-dominated solutions. Column '%' represents the relative frequency of the feature in the pool.

	Datasets					
Rank	STACK		COOK		ENGLISH	
	Feature	%	Feature	%	Feature	%
131	*tr-fog* (read)	6	*u-rateque* (user)	7	*tm-bm25*$_{b,w}$ (relev)	6
132	*ts-stdcod* (structure)	6	*tm-nwnou*$_b$ (relev)	4	*tm-bm25*$_{b,b}$ (relev)	6
133	*ts-listit* (structure)	6	*tm-wmtch*$_{b,d}$ (relev)	4	*tm-wmch*$_{t,d}$ (relev)	6
134	*ts-secs* (structure)	6	*tm-wmch*$_{t,bg}$ (relev)	4	*tm-wmch*$_{t,bg}$ (relev)	6
135	*ts-avquot* (structure)	6	*ty-prepo* (style)	4	*ty-passive* (style)	6
136	*ts-mxquot* (structure)	6	*ty-capwrd* (style)	4	*ty-sconj* (style)	6
137	*ts-avgcod* (structure)	6	*ty-dotcnt* (style)	4	*ty-klddis* (style)	6
138	*u-avansq* (user)	6	*ug-hub* (ugraph)	4	*ug-auth* (ugraph)	6
139	*tl-charcnt* (length)	6	*r-comque* (hist)	4	*r-uniqusr* (hist)	6
140	*u-commnt* (user)	6	*tr-flesh* (read)	4	*ts-maxcod* (structure)	6
141	*u-sugedt* (user)	6	*ts-quotes* (structure)	4	*ts-listit* (structure)	6
142	*u-top3an* (user)	6	*ts-codes* (structure)	4	*ts-subsec* (structure)	6
143	*u-avqtag* (user)	6	*ts-list* (structure)	4	*u-mxatag* (user)	6
144	*u-mncoma* (user)	6	*ts-maxcod* (structure)	4	*u-mnrqtag* (user)	6
145	*u-mnqtag* (user)	6	*ts-avquot* (structure)	4	*u-mrateqt* (user)	6
146	*u-lastac* (user)	6	*ts-avgcod* (structure)	4	*u-prtp3qu* (user)	6
147	*u-mxcoma* (user)	6	*u-avansq* (user)	4	*u-prtp3an* (user)	6
148	*u-rateque* (user)	6	*u-apsuged* (user)	4	*u-enpytag* (user)	6
149	*u-mrktq* (user)	6	*u-daycrt* (user)	4	*tm-nwnou*$_b$ (relev)	3
150	*u-prtp3an* (user)	6	*u-mxatag* (user)	4	*tm-wmtch*$_{b,w}$ (relev)	3
151	*u-solvqu* (user)	6	*u-mxcoma* (user)	4	*tm-phmch*$_{b,dg}$ (relev)	3
152	*tr-kincaid* (read)	0	*u-rkans* (user)	4	*tm-bm25*$_{t,b}$ (relev)	3
153	*u-rkqust* (user)	0	*u-prtp3qu* (user)	4	*tm-bm25*$_{b,d}$ (relev)	3
154	*tm-wmch*$_{t,b}$ (relev)	0	*u-enpytag* (user)	4	*ty-plgphr* (style)	3
155	*u-daycrt* (user)	0	*u-enpytag* (user)	4	*ty-auxverb* (style)	3
156	*u-mnrqtag* (user)	0	*u-xrateqt* (user)	4	*ty-conj* (style)	3
157	*tm-wmtch*$_{b,bg}$ (relev)	0	*u-lastac* (user)	0	*ty-nomina* (style)	3
158	*u-arkta* (user)	0	*tm-wmch*$_{t,d}$ (relev)	0	*ty-infnois* (style)	3
159	*u-mrateqt* (user)	0	*u-avcomq* (user)	0	*r-qsuged* (hist)	3
160	*u-mxqtag* (user)	0	*u-arkta* (user)	0	*tr-ari* (read)	3
161	*tm-phmch*$_{b,w}$ (relev)	0	*u-mrateqt* (user)	0	*tr-kincaid* (read)	3
162	*ty-sintp* (style)	0	*ts-listit* (structure)	0	*ts-inlink* (structure)	3
163	*u-avrqtag* (user)	0	*u-anpytag* (user)	0	*tr-lix* (read)	3
164	*u-apsuged* (user)	0	*u-mxqtag* (user)	0	*ts-list* (structure)	3
165	*u-rkans* (user)	0	*ts-mxquot* (structure)	0	*ts-mxquot* (structure)	3
166	*tr-flesh* (read)	0	*ts-boldit* (structure)	0	*u-avansq* (user)	3
167	*ty-sartic* (style)	0	*ty-conj* (style)	0	*tl-charcnt* (length)	3
168	*tm-nwaad*$_t$ (relev)	0	*tm-wmtch*$_{b,b}$ (relev)	0	*u-quests* (user)	3
169	*ts-stdsecl* (structure)	0	*tr-lix* (read)	0	*u-avratag* (user)	3
170	*u-mxrqtag* (user)	0	*ty-sartic* (style)	0	*u-apsuged* (user)	3
171	*u-avcoma* (user)	0	*r-qsuged* (hist)	0	*u-mnqtag* (user)	3
172	*u-enpytag* (user)	0	*tm-wmtch*$_{b,w}$ (relev)	0	*u-maxansq* (user)	3
173	*tm-wmtch*$_{b,w}$ (relev)	0	*r-stdpusr* (hist)	0	*u-mxqtag* (user)	3
174	*u-top3qu* (user)	0	*tm-phmch*$_{b,bg}$ (relev)	0	*u-solvqu* (user)	3
175	*ts-maxcod* (structure)	0	*ts-inlink* (structure)	0	*u-mncomq* (user)	0
176	*u-arktq* (user)	0	*u-mxatag* (user)	0	*u-arkta* (user)	0
177	*ts-subsec* (structure)	0	*ts-stdcod* (structure)	0	*ty-kldtag* (style)	0
178	*r-uniqusr* (hist)	0	*tm-worder*$_t$ (relev)	0	*ty-questn* (style)	0
179	*u-arateat* (user)	0	*ty-plgphr* (style)	0	*ts-xlnks* (structure)	0
180	*u-xrkta* (user)	0	*r-asuged* (hist)	0	*u-xrkta* (user)	0
181	*tm-phmch*$_{b,b}$ (relev)	0	*ty-dotden* (style)	0	*tm-phmch*$_{b,b}$ (relev)	0
182	*u-mnratag* (user)	0	*u-xrkta* (user)	0	*tm-bm25*$_{t,dg}$ (relev)	0
183	*tm-nwver*$_b$ (relev)	0	*ts-ssssec* (structure)	0	*ts-quotes* (structure)	0
184	*ug-hub* (ugraph)	0	*ts-secs* (structure)	0	*tm-nwver*$_b$ (relev)	0
185	*u-mxcomq* (user)	0	*tr-liau* (read)	0	*ug-hub* (ugraph)	0
186	*r-queage* (hist)	0	*tm-phmch*$_{t,bg}$ (relev)	0	*u-mxcomq* (user)	0

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we have introduced a general quality evaluation approach for user-generated content based on the idea that quality is a multi-faceted concept. Accordingly, the quality assessment of an CI must consider the different quality views applicable to that item. In our framework, we propose the use of views that are related to several quality dimensions and sources. These views are: (1) general enough to be applicable to many kinds of items as long as they have a predetermined and rational quality evaluation criteria, (2) capable to highlight quality dimensions where quality indicators are lacking, (3) lead to a natural combination strategy to obtain a summarized quality estimate, and (4) useful to organize and assess the importance of quality views and indicators.

We show the generality of our proposed framework by applying it to six datasets belonging to two different domains: Wikis and Q&A Forums. We were able to extract quality indicators related to all views and dimensions in both domains. We were also able to optimize different performance criteria specific for each domain, such as numeric assessments in Wikis and ranking order in Q&A Forums.

The organization of the indicators into views led to the assessment of quality by combining individual view quality estimates. We proposed to combine such views using a supervised approach which resulted in a two-step learning framework. We first learn individual view assessments and then learn how to combine them. This approach was able to improve the results in five out of six datasets. We also spent a lot of experimental effort to better understand how and when our multi-view approach is supposed to work. We found, for instance, that it is able to improve the results when views reinforce correct decisions, while errors occur in more severe disagreements, which usually occur in the hardest instances to predict.

The organization of the indicators into views has also allowed us to study their importance in different domains and datasets. We observed that views and indicators have varying importance according to the domain and dataset where they were applied. Our study was based on correlation analysis and inspection of Pareto non-dominated solutions obtained by a genetic feature selection algorithm, SPEA2. We have found that (a) some views are important in both domains (e.g., History view) where others are more important to specific domains (Structure in Wiki domain and User in Q&A Forums domain) and (b) while many indicators carry redundant information and can be safely ignored, all the views (except by Readability) were consistently useful in all datasets.

It is interesting to note that the organization of the indicators into views according to dimensions and sources clearly indicates the lacking of semantic indicators. In fact, most of the research in literature has focused on syntactic dimensions. Important semantic dimensions, such as coherence and factual accuracy, are only indirectly estimated by means of indicators extracted from the edit history source. In future, more research effort has to be directed towards the assessment of semantic quality dimensions.

7.2 Implications for other Data and Information Quality Problems on the Web

Although we have focused in quality assessment of articles in Wikis and Q&A Forums, some of our results may have implications for other data and quality problems on the Web. For example, our consistent results from these domains indicates that similar results may be obtained using our proposed techniques and set of features for other collaborative projects, such as community-oriented blogs, and or even for the problem of ranking pages in a search engine.

In this work, we observed that usually Edit History features are good predictors of the content quality. At the same time, the usually poor effectiveness of Readability features indicates they may have to be adapted to each specific collection. Moreover, for scenarios such as the assessment of quality in very short and informal messages, eg. posts in microblogs, such as Twitter[1], these features may have to be completely

[1] For example, users trying to influence the opinions of others may have a tendency to write better written messages [Bigonha et al., 2010].

rethought. For instance, instead of a static equation, we could use machine learning to combine different aspects of the text to infer the readability of a post.

Using inverse reasoning, lack of quality may also indicate other types of problems such as spam or the presence of attacks such as vandalism [Chin et al., 2010]. Thus, methods for detecting these problems could benefit from our proposal.

Further, our results can benefit designers of applications and tools that try to estimate or communicate issues regarding quality of content to users, such as the work of Pirolli et al. [2009] and Chevalier et al. [2010].

Finally, content quality is a very subjective concept and depends on aspects that can be different among communities. As such, it depends on different evaluation criteria. In Wikipedia, for instance, different project groups can evaluate the quality of a same article considering aspects as diverse as its adherence to specific structural style and content coverage[2]. Thus, for any system which aims at automatically assessing content quality, it would be very useful to learn about its target community and its specific quality criteria. This is even more important when we consider much more diversified content, such as those found in blogs, twitter, and News pages, to cite a few.

7.3 Future Work

During this research, we observed many opportunities for future work. In the following paragraphs, we describe them, beginning with those that can be more easily implemented.

Although we have studied many features during this work, other features could be also explored. For example, as we did in Q&A forums, we can estimate the amount of typos in a Wiki article by counting the number of words not present in Wordnet and in the list of common mistakes. For both domains, we can improve graph features using PageRank weighted by the topic of the CI. In this way, we can see which article/user is important within an specific topic. We can also improve some linguistic features such as Readability. Instead of using general parameters for such features, we can use parameters learned in a per-collection basis.

The process of quality classification is still manual in most sites. Also, no visualization assistance is provided to the users to facilitate the recognition of the quality

[2] In Wikipedia, articles about people, places, and insects are expected to fit specific structural organization, different among them. The coverage of an article (eg, the biography of the statistician and geneticist Ronald Fisher) can be considered very adequate for a group (Biology) while imprecise for another (Statistics).

of the content they consume. Thus, tools could be designed to assist users with the process of quality assessment and visualization. An example of an application to assist users in quality assessment is described as follows. Articles from the Portuguese Wikipedia are automatically assessed by a robot[3]. Note that, these robots attempt to measure the quality of the articles in a naive way, that is, by checking if articles have a specific length, a certain number of images, or if it was manually labeled with tags indicating flaw conditions. For that reason, we intend to create a robot to infer the quality of the Portuguese Wikipedia, using the approach proposed in this thesis. We expect that our method can automatically infer the quality of the articles better since it takes into account much more features.

We can also propose other features in order to analyze its impact on quality assessment. For example, we can analyze the impact of typos (used here in Q&A Forums) in order to predict the quality of Wiki articles. In both domains, we can analyze the sentiment polarity of the text as well as if it is relevant to predict the quality of content using this kind of feature. Furthermore, we can try to infer the empirical security dimension in those domains by using indicators like number of reverted edits and other user features – some of them were already proposed here such as number of registered users and ProbReview. Other interesting tool which we can create is an application that, given an Wiki article or a answer from a Q&A Forum, it gives all the quality indicators used in this thesis.

We also can use the features we proposed to predict quality in others domains such as product reviews, collaborative translation (for instance, the Cucumis collaborative translation community discussed in Chapter 1), and reputation of the user [Bigonha et al., 2010; Wöhner et al., 2011; Anthony et al., 2005]. In addition, some pages such as blogs and news have tools to share user preferences such as the "like" button used in Facebook. We can use our features to predict how many "likes" one pages will have. Furthermore, In this thesis, we could see the importance of textual features to infer the quality specially in Wikis. Thus, another future work is to learn how to rank web pages using textual features studied here together with well-known features of a web document such as PageRank and HITS. We can also use quality indicators in order to recommend reviews since quality can be important in that context.

Other interesting topic to investigate is the quality evolution in time of user-generated content such as Wikipedia. For each Wikipedia article, it is possible to obtain its version when it was promoted to (or demoted from) a certain quality class. Thus, given q_{it} (the quality q of an article i in a certain time t), the problem here is to

[3] http://pt.wikipedia.org/wiki/Wikip%C3%A9dia:Avalia%C3%A7%C3%A3o_autom%C3%A1tica

predict which class q_{t+1} the article will have in a certain time $t + 1$ in the future. To accomplish this, we can use as features the quality value q_{it} and the proposed features of this work (i.e., presented in Section 3.2) of the article in the time t, together with the features in the time $t + 1$, and the difference between the values of features in t and $t + 1$. The importance of this task is to predict if an article a, which has already a quality assigned to it, has improved or decreased its quality value. Our hypothesis is that, by using the previous quality value, we can improve the quality prediction of the article. In Q&A Forum this problem also can be interesting specially because, as mentioned previously, an answer can tend to a higher ratting than a newer one.

Finally, we also intend to explore multi-classification in quality assessment. As previously mentioned, articles from Wikipedia are generally evaluated by groups of users that belong to different projects. A project is a group of articles sharing a common topic (eg. Evolution, Biographies, Places). Projects are maintained by users which can help to organize, manage, evaluate and, in some cases, define a standard editing process [4]. This is the case of the article about *Charles Darwin*, which is associated with projects *Biographies* and *Evolution.* Since an article can belong to multiple projects, it can receive multiple, and distinct, quality evaluations. For instance, the article about Darwin can be considered *FA* by the Evolution project but only *GA* by the Biographies project, since they can have different requirements. In our work, we decided not to deal with multi-classification, using only articles which were classified into just one class by all their projects. However, predicting the quality of an article per project can be an interesting future work.

[4] `http://en.wikipedia.org/wiki/Wikipedia:WikiProject`

Bibliography

Adler, T. B. and de Alfaro, L. (2007). A Content-Driven Reputation System for the Wikipedia. In *Proceedings of the 16th International Conference on World Wide Web*, pages 261--270, Banff, Alberta, Canada.

Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. (2008). Finding High-Quality Content in Social Media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 183--194, Palo Alto, California, USA.

Alexander, J. E. and Tate, M. A. (1999). *Web Wisdom; How to Evaluate and Create Information Quality on the Web*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

Anderson, A., Huttenlocher, D., Kleinberg, J., and Leskovec, J. (2012). Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 850--858, Beijing, China.

Anthony, D., Smith, S., and Williamso, T. (2005). Explaining Quality in Internet Collective Goods: Zealots and Good Samaritans in the Case of Wikipedia. `http://web.mit.edu/iandeseminar/Papers/Fall2005/anthony.pdf`. Hanover: Dartmouth College.

Anthony J. Onwuegbuzie, Larry Daniel, N. L. L. (2007). Pearson Product-Moment Correlation Coefficient. In Salkind, N. J., editor, *Encyclopedia of Measurement and Statistics*, pages 751–756. SAGE Publications, Inc.

Argamon, S., Koppel, M., Fine, J., and Shimoni, A. R. (2003). Gender, Genre, and Writing Style in Formal Written Texts. *Text & Talk An Interdisciplinary Journal of Language, Discourse & Communication Studies*, 23:321--346.

Attardi, G., dell'Orletta, F., Simi, M., Chanev, A., and Ciaramita, M. (2007). Multilingual Dependency Parsing and Domain Adaptation using DeSR. In *Proceedings of the*

2007 Conference on Computational Natural Language Learning, pages 1112--1118, Prague, Czech Republic.

Barnett, V. and Lewis, T. (1994). *Outliers in Statistical Data.* Wiley Series in Probability & Statistics. Wiley.

Bendersky, M., Croft, W. B., and Diao, Y. (2011). Quality-Biased Ranking of Web Documents. In *Proceedings of the Fourth International Conference on Web Search and Data Mining*, pages 95--104, Hong Kong.

Benevenuto, F., Rodrigues, T., Almeida, V., Almeida, J., and Gonçalves, M. (2009). Detecting Spammers and Content Promoters in Online Video Social Networks. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 620--627, Boston, MA, USA.

Bethard, S., Wetzer, P., Butcher, K., Martin, J. H., and Sumner, T. (2009). Automatically Characterizing Resource Quality for Educational Digital Libraries. In *Proceedings of the 2009 Joint International Conference on Digital libraries*, pages 221--230, Austin, TX, USA.

Bigonha, C., Cardoso, T. N., Moro, M. M., Almeida, V., and Gonçalves, M. A. (2010). Detecting Evangelists and Detractors on Twitter. In *Anais do Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 107--114, Belo Horizonte, Minas Gerais, Brazil.

Björnsson, C. (1968). *Lesbarkeit durch Lix.* Stockholm: Pedagogiskt Centrum.

Blum, A. and Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92--100, Madison, WI, USA.

Blumenstock, J. E. (2008). Size Matters: Word Count as a Measure of Quality on Wikipedia. In *Proceedings of the 17th International Conference on World Wide Web*, pages 1095--1096, Beijing, China.

Boldi, P. and Vigna, S. (2004). The Webgraph Framework I: Compression Techniques. In *Proceedings of the 13th International Conference on World Wide Web*, pages 595--601, New York, NY, USA.

Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proceedings of the Sixth Workshop on Very Large Corpora*, volume 182.

Brandão, W. C., Santos, R. L., Ziviani, N., Moura, E. S., and Silva, A. S. (2014). Learning to Expand Queries using Entities. *Journal of the Association for Information Science and Technology*, 65(9):1870--1883.

Brin, S. and Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1-7):107--117.

Burel, G., He, Y., and Alani, H. (2012). Automatic Identification of Best Answers in Online Enquiry Communities. *The Semantic Web: Research and Applications, Lecture Notes in Computer Science*, 7295:514–529.

Cessie, S. L. and Houwelingen, J. V. (1992). Ridge Estimators in Logistic Regression. *Journal of the Royal Statistical Society. Series C*, 41(1):191--201.

Chang, C. C. and Lin, C. J. (2001). LIBSVM: a Library for Support Vector Machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chevalier, F., Huot, S., and Fekete, J.-D. (2010). WikipediaViz: Conveying Article Quality for Casual Wikipedia Readers. In *Proceedings of the 2010 Pacific Visualization Symposium*, pages 49 –56, Taiwan.

Chin, S.-C., Street, W. N., Srinivasan, P., and Eichmann, D. (2010). Detecting Wikipedia Vandalism with Active Learning and Statistical Language Models. In *Proceedings of the 4th Workshop on Information Credibility*, pages 3--10, Raleigh, North Carolina, USA.

Chu, W., Keerthi, S. S., and Ong, C. J. (2001). A Unified Loss Function in Bayesian Framework for Support Vector Regression. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 51--58, San Francisco, CA, USA.

Chua, A. Y. and Banerjee, S. (2013). So Fast so Good: An Analysis of Answer Quality and Answer Speed in Community Question-Answering Sites. *Journal of the American Society for Information Science and Technology*, 64(10):2058--2068.

Ciaramita, M. and Altun, Y. (2006). Broad-coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 594--602, Stroudsburg, PA, USA.

Coleman, M. and Liau, T. L. (1975). A Computer Readability Formula Designed for Machine Scoring. *Journal of Applied Psychology*, 60(2):283--284.

Collins (2003). *Collins English Dictionary – Complete and Unabridged.* HarperCollins Publishers.

Conti, R., Marzini, E., Spognardi, A., Matteucci, I., Mori, P., and Petrocchi, M. (2014). Maturity Assessment of Wikipedia Medical Articles. In *Proceedings of the 27th International Symposium on Computer-Based Medical Systems*, pages 281--286.

Cusinato, A., Della Mea, V., Di Salvatore, F., and Mizzaro, S. (2009). QuWi: Quality Control in Wikipedia. In *Proceedings of the 3rd Workshop on Information Credibility on the Web*, pages 27--34, Madrid, Spain. ACM.

Dalip, D. H., Cardoso, T., Gonçalves, M., Cristo, M. U., and Calado, P. (2012a). A Multi-view Approach for the Quality Assessment of Wiki Articles. *Journal of Information and Data Management*, 3(1).

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. (2009). Automatic Quality Assessment of Content Created Collaboratively by Web Communities: a Case Study of Wikipedia. In *Proceedings of the 2009 Joint International Conference on Digital libraries*, pages 295--304, Austin, TX, USA.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. (2011a). Automatic Assessment of Document Quality in Web Collaborative Digital Libraries. *Journal of Data and Information Quality*, 2(3):1--30.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. (2012b). On MultiView-Based Meta-Learning for Automatic Quality Assessment of Wiki Articles . In *Proceedings of the 2012 International Conference on Theory and Practice of Digital Libraries*, Paphos, Cyprus.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. (2013). Exploiting User Feedback to Learn to Rank Answers in Q&A Forums: A Case Study with Stack Overflow. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 543--552, Dublin, Ireland.

Dalip, D. H., Lima, H., Gonçalves, M. A., Cristo, M., and Calado, P. (2014). Quality Assessment of Collaborative Content with Minimal Information. In *Proceedings of the 2014 ACM/IEEE Joint International Conference on Digital libraries*, pages 295--304, London, England.

Dalip, D. H., Santos, R. L., Oliveira, D. R. R., Amaral, V. F., Gonçalves, M. A., Prates, R. O., Minardi, R. C. M., and Almeida, J. M. D. (2011b). GreenWiki - A Tool to

Support Users ' Assessment of the Quality of Wikipedia Articles. In *Proceedings of the 2011 Joint International Conference on Digital libraries*, pages 469–470, Ottawa, Canada.

De la Calzada, G. and Dekhtyar, A. (2010). On Measuring the Quality of Wikipedia Articles. In *Proceedings of the 4th workshop on Information Credibility*, pages 11--18, Raleigh, North Carolina, USA.

Dondio, P., Barrett, S., and Weber, S. (2006a). Calculating the Trustworthiness of a Wikipedia Article Using Dante Methodology. In *International Conference on e-Society*, Dublin, Ireland.

Dondio, P., Barrett, S., Weber, S., and Seigneur, J. (2006b). Extracting Trust from Domain Analysis: A Case Study on the Wikipedia Project. In *Autonomic and Trusted Computing*, pages 362--373. Springer Berlin / Heidelberg.

dos Santos, E. M., Sabourin, R., and Maupin, P. (2006). Single and Multi-Objective Genetic Algorithms for the Selection of Ensemble of Classifiers. In *International Joint Conference on Neural Networks, 2006*, pages 3070--3077, Athens, Greece.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. (1996). Support Vector Regression Machines. In Mozer, M., Jordan, M. I., and Petsche, T., editors, *Neural Information Processing Systems*, pages 155–161. MIT Press.

Fleiss, J. L. and Cohen, J. (1973). The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability. *Educational and Psychological Measurement*, 33(3):613--619.

Flekova, L., Ferschke, O., and Gurevych, I. (2014). What Makes a Good Biography?: Multidimensional Quality Analysis Based on Wikipedia Article Feedback Data. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 855--866, Seoul, Korea.

Flesch, R. (1948). A New Readability Yardstick. *Journal of Applied Psychology*, pages 221--235.

Fogg, B. J., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., Paul, J., Rangnekar, A., Shon, J., Swani, P., and Treinen, M. (2001). What makes Web sites credible?: a Report on a Large Quantitative Study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 61--68, Seattle, Washington, United States.

Fogg, B. J., Soohoo, C., Danielson, D. R., Marable, L., Stanford, J., and Tauber, E. R. (2003). How do Users Evaluate the Credibility of Web Sites?: a Study with over 2,500 Participants. In *Proceedings of the 2003 Conference on Designing for User Experiences*, pages 1--15, San Francisco, California, USA.

Freund, Y. and Mason, L. (1999). The Alternating Decision Tree Learning Algorithm. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 124--133, Bled, Slovenia.

Ge, M. and Helfert, M. (2007). A Review of Information Quality Research - Develop a Research Agenda. In *International Conference on Information Quality*, pages 76–91.

Gelbukh, A., editor (2011). *Computational Linguistics and Intelligent Text Processing*, volume 6609 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Gkotsis, G., Stepanyan, K., Pedrinaci, C., Domingue, J., and Liakata, M. (2014). It's all in the Content: State of the Art Best Answer Prediction Based on Discretisation of Shallow Linguistic Features. In *Proceedings of the 2014 ACM Conference on Web science*, pages 202--210, Bloomington, Indiana, USA.

Gunning, R. (1952). *The Technique of Clear Writing*. McGraw-Hill International Book Co.

Han, J., Chen, K., and Wang, J. (2014). Web Article Quality Ranking Based on Web Community Knowledge. *Computing*, pages 1–29.

Han, J. and Wang, C. (2011). Probabilistic Quality Assessment Based on Article's Revision History. *Database and Expert Systems Applications*, 2:574--588.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2000). A Practical Guide to Support Vector Classification. *National Taiwan University*.

Hu, M., Lim, E.-P., Sun, A., Lauw, H. W., and Vuong, B.-Q. (2007). Measuring Article Quality in Wikipedia: Models and Evaluation. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge management*, pages 243--252, Lisbon, Portugal.

Järvelin, K. and Kekäläinen, J. (2000). IR Evaluation Methods for Retrieving Highly Relevant Documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 41--48, Athens, Greece.

Jeon, J., Croft, W. B., Lee, J. H., and Park, S. (2006). A Framework to Predict the Quality of Answers with Non-textual Features. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 228--235, Seattle, Washington, USA.

Joachims, T. (2002). Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133--142, Edmonton, Alberta, Canada.

Kakade, S. M. and Foster, D. P. (2007). Multi-view Regression via Canonical Correlation Analysis. In *Proceedings of the Conference on Learning Theory*, pages 82--96, San Diego, CA, USA.

Kendall, M. G. (1938). A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81--93.

Kim, S. and Oh, S. (2009). Users' Relevance Criteria for Evaluating Answers in a Social Q&A Site. *Journal of the American Society for Information Science and Technology*, 60(4):716--727.

Kleinberg, J. M. (1999). Hubs, Authorities, and Communities. *ACM Computing Surveys*, 31(4es).

Korfiatis, N., Poulos, M., and Bokos, G. (2006). Evaluating Authoritative Sources Using Social Networks: An Insight from Wikipedia. *Online Information Review*, 30(3):252--262.

Krowne, A. (2003). Building a Digital Library the Commons-based Peer Production Way. *D-Lib magazine*, 9(1082).

Laumanns, M., Zitzler, E., and Thiele, L. (2001). On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization. In *Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization*, pages 181--196, London, UK, UK.

Li, B., Jin, T., Lyu, M. R., King, I., and Mak, B. (2012). Analyzing and Predicting Question Quality in Community Question Answering Services. In *Proceedings of the 21st International Conference on World Wide Web (companion volume)*, pages 775--782, Lyon, France.

Maged, Maramba, I., and Wheeler, S. (2006). Wikis, Blogs and Podcasts: a New Generation of Web-based Tools for Virtual Collaborative Clinical Practice and Education. *BMC Medical Education*, 6:41+.

McLaughlin, G. H. (1969). SMOG Grading: A New Readability Formula. *Journal of Reading*, 12(8):639--646.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill Higher Education.

Mohan, A., Chen, Z., and Weinberger, K. (2011). Web-Search Ranking with Initialized Gradient Boosted Regression Trees. *JMLR Workshop and Conference Proceedings: Proceedings of the Yahoo! Learning to Rank Challenge*, 14:77–89.

Moraes, F., Vasconcelos, M., Prado, P., Dalip, D., Almeida, J., and Gonçalves, M. (2013). Polarity Detection of Foursquare Tips. In Jatowt, A., Lim, E.-P., Ding, Y., Miura, A., Tezuka, T., Dias, G., Tanaka, K., Flanagin, A., and Dai, B., editors, *Social Informatics*, Lecture Notes in Computer Science, vol. 8238, pages 153–162. Springer International Publishing.

Opitz, D. W. (1999). Feature Selection for Ensembles. In *Proceedings of the 16th National Conference on Artificial Intelligence*, pages 379--384, Orlando, FL, USA.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project.

Pal, A. and Konstan, J. A. (2010). Expert Identification in Community Question Answering: Exploring Question Selection Bias. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1505--1508, Toronto, ON, Canada. ACM.

Pirolli, P., Wollny, E., and Suh, B. (2009). So You Know You're Getting the Best Possible Information: a Tool that Increases Wikipedia Credibility. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pages 1505--1508, Boston, MA, USA. ACM.

Ponzanelli, L., Mocci, A., Bacchelli, A., Lanza, M., and Fullerton, D. (2014). Improving Low Quality Stack Overflow Post Detection. In *Proceedings of the International Conference on Software Maintenance and Evolution*, pages 541--544, Victoria, Canada.

Potthast, M., Stein, B., and Gerling, R. (2008). Automatic Vandalism Detection in Wikipedia. In Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., and White, R., editors, *Advances in Information Retrieval*, Lecture Notes in Computer Science, vol. 4956, pages 663–668. Springer Berlin Heidelberg.

Rassbach, L., Pincock, T., and Mingus, B. (2007). Exploring the Feasibility of Automatically Rating Online Article Quality. `http://upload.wikimedia.org/wikipedia/wikimania2007/d/d3/RassbachPincockMingus07.pdf`.

Ressler, S. (1993). *Perspectives on Electronic Publishing: Standards, Solutions, and More*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Robertson, S. E. and Walker, S. (1994). Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 232--241, Dublin, Ireland.

Rosenzweig, R. (2006). Can History Be Open Source? Wikipedia and the Future of the Past. *The Journal of American History*, 93(1):pp. 117–146.

Rubio, R., Martín, S., and Morán, S. (2010). Collaborative Web Learning Tools: Wikis and Blogs. *Computer Applications in Engineering Education*, 18:502–511.

Sakai, T., Ishikawa, D., Kando, N., Seki, Y., Kuriyama, K., and Lin, C.-Y. (2011). Using Graded-relevance Metrics for Evaluating Community QA Answer Selection. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 187--196, Hong Kong, China.

Shah, C. and Pomerantz, J. (2010). Evaluating and Predicting Answer Quality in Community Q&A. In *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Geneva, Switzerland.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

Smith, E. A. and Senter, R. J. (1967). Automated Readability Index. *Aerospace Medical Division*.

Stvilia, B., Gasser, L., Twidale, M. B., and Smith, L. C. (2007). A Framework for Information Quality Assessment. *Journal of the American Society for Information Science and Technology*, 58(12):1720–1733.

Stvilia, B., Twidale, M. B., Smith, L. C., and Gasser, L. (2005). Assessing Information Quality of a Community-based Encyclopedia. In *Proceedings of the International Conference on Information Quality*, pages 442--454, USA.

Surdeanu, M., Ciaramita, M., and Zaragoza, H. (2008). Learning to Rank Answers on Large Online Q&A Collections. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, Columbus, OH, USA.

Suryanto, M. A., Lim, E. P., Sun, A., and Chiang, R. H. L. (2009). Quality-aware Collaborative Question Answering: Methods and Evaluation. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 142--151, Barcelona, Spain.

Suzuki, Y. and Yoshikawa, M. (2013). Assessing Quality Score of Wikipedia Article Using Mutual Evaluation of Editors and Texts. In *Proceedings of the 22Nd ACM International Conference on Information and Knowledge Management*, pages 1727--1732, San Francisco, California, USA.

Tejay, G., Dhillon, G., and Chin, A. G. (2006). Data Quality Dimensions for Information Systems Security: A Theoretical Exposition. In *Security Management, Integrity, and Internal Control in Information Systems*, pages 21--39. Springer.

Tsymbal, A., Puuronen, S., and Patterson, D. W. (2003). Ensemble Feature Selection with the Simple Bayesian Classification. *Information Fusion*, 4(2):87--100.

Vafaie, H. and Imam, I. F. (1994). Feature Selection Methods: Genetic Algorithms vs Greedy-like Search. In *Proceedings of International Conference on Fuzzy and Intelligent Control Systems*, Orlando, FL, USA.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc.

Voß, J. (2005). Measuring Wikipedia. In *Proceedings 10th International Conference of the International Society for Scientometrics and Informetrics*, Stockholm, Sweden.

Wang, R. Y. and Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5--34.

Weimer, M., Gurevych, I., and Mühlhäuser, M. (2007). Automatically Assessing the Post Quality in Online Discussions on Software. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 125--128, Morristown, NJ, USA.

Weiss, G. M. and Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19:315--354.

Wikipedia (2015a). Version 1.0 editorial team/assessment. `http://en.wikipedia.org/wiki/Wikipedia:Version_1.0_Editorial_Team/Assessment`.

Wikipedia (2015b). Version 1.0 editorial team/release version criteria. `https://en.wikipedia.org/wiki/Wikipedia:1.0/Criteria`.

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics*, pages 80–83.

Wilkinson, D. M. and Huberman, B. A. (2007). Cooperation and quality in Wikipedia. In *Proceedings of the 2007 International Symposium on Wikis*, pages 157--164, Montreal, Quebec, Canada.

Witten, I. H. and Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques*. 1ª edition.

Wöhner, T., Köhler, S., and Peters, R. (2011). Automatic Reputation Assessment in Wikipedia. *Proceedings of the 2011 International Conference on Computer and Information Science*.

Wöhner, T. and Peters, R. (2009). Assessing the Quality of Wikipedia Articles with Lifecycle Based Metrics. In *Proceedings of the 5th International Symposium on Wikis*, pages 16:1--16:10, Orlando, Florida.

Wolpert, D. (1992). Stacked Generalization. *Neural Networks*, 5(2):241--259.

Xu, Y. and Luo, T. (2011). Measuring Article Quality in Wikipedia: Lexical Clue Model. In *Proceedings of the 3rd Symposium on Web Society*, pages 141 –146, Oackland, CA, USA.

Zeng, H., Alhossaini, M., Ding, L., Fikes, R., and Mcguinness, D. L. (2006). Computing trust from revision history. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust*, Oshawa, Canada.

Zhang, J., Ackerman, M. S., and Adamic, L. (2007). Expertise networks in online communities. In *Proceedings of the 16th international conference on World Wide Web*, page 221, Banff, Alberta, Canada. ACM Press.

Zheng, R., Li, J., Chen, H., and Huang, Z. (2006). A Framework for Authorship Identification of Online Messages: Writing-style Features and Classification Techniques. *Journal of the American Society for Information Science and Technology*, 57:378--393.

Zhou, J., Xu, Q., Chen, J., and Qu, W. (2009). A Multi-view Approach for Relation Extraction. In Liu, W., Luo, X., Wang, F., and Lei, J., editors, *Web Information Systems and Mining*, Lecture Notes in Computer Science, vol. 5854, pages 53–62. Springer Berlin / Heidelberg.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical report.