



Open Library of Humanities



Part of the Ubiquity
Partner Network

Digital Studies /
Le champ numérique

Research

How to Cite: Jimenez-Mavillard, Antonio and Juan-Luis Suárez. 2018. Community Engagement and Quality Knowledge with Stackoverflow's Reputation System: Applications to Bullipedia. *Digital Studies/Le champ numérique* 8(1): 2, pp. 1–30, DOI: <https://doi.org/10.16995/dscn.293>

Published: 23 January 2018

Peer Review:

This is a peer-reviewed article in *Digital Studies/Le champ numérique*, a journal published by the Open Library of Humanities.

Copyright:

© 2018 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Open Access:

Digital Studies/Le champ numérique is a peer-reviewed open access journal.

Digital Preservation:

The Open Library of Humanities and all its journals are digitally preserved in the CLOCKSS scholarly archive service.

RESEARCH

Community Engagement and Quality Knowledge with Stackoverflow's Reputation System: Applications to Bullipedia

Antonio Jimenez-Mavillard and Juan-Luis Suárez

The University of Western Ontario, CA

Corresponding author: Antonio Jimenez-Mavillard (ajimene6@uwo.ca)

Bullipedia, the online gastronomic encyclopedia, is an idea yet to be developed. In this work, we analyze Stack Overflow (SO) and extract some good practices from this popular question-and-answer (Q&A) site to incorporate them into the future Bullipedia. SO is an online forum in which users ask and answer questions related to programming, web development, operating systems, and other technical topics. Expertise is rewarded through a detailed reputation system: questions and answers can receive up and downvotes from other members of the community so that their authors (askers and answerers) gain reputation for posing good questions and providing helpful solutions. Besides this, the asker may mark (accept) one of the answers as the best one at any point. In this paper, we present a study on how this reputation system can be used to predict the likely accepted answer (from a set of candidate answers) for a yet unresolved question. In our approach, we selected a subset of questions with their respective answers, and for each answer we created a question-answer pair (quan). Then we extracted a set of key features from every quan, and applied supervised machine learning techniques to train a classifier that learnt, based on those features, whether or not a quan contained the accepted answer for that question. Finally, we made use of the trained classifier to predict if, given a quan (related to a question with no marked answer), its answer might potentially be the accepted one for the question. Our findings show that the model previously obtained predicted the possible answer correctly for every question with high accuracy (88 percent of the time). A question and its accepted answer constitutes a source of quality knowledge, as it provides the solution to a specific problem. We propose to adopt a similar Q&A forum and reputation system for Bullipedia, and then apply a similar classification model to identify the best answer for unsolved questions.

Keywords: Q&A site; StackOverflow; machine learning; classification; Linear SVC; knowledge; online community; reputation; trust; Bullipedia

L'encyclopédie gastronomique en ligne Bullipedia est une idée qui reste encore à développer. Dans ce travail, nous analysons le forum *Stack Overflow* (SO) et soutirons de bonnes pratiques à partir de ce site populaire de questions-réponses afin de les incorporer dans le futur Bullipedia.

SO est un forum en ligne dans lequel les utilisateurs posent et répondent à des questions reliées à la programmation, au développement web, aux systèmes d'exploitation, et autres sujets techniques. L'expertise est récompensée au moyen d'un système de réputation détaillé : les questions et les réponses peuvent recevoir des votes pour ou contre des autres membres de la communauté de sorte que leurs auteurs (demandeurs et répondeurs) acquièrent une réputation pour avoir posé de bonnes questions et avoir fourni des solutions utiles. À part cela, le demandeur peut marquer (*accepter*) l'une des réponses comme étant la meilleure à n'importe quel moment. Dans cet article, nous présentons une étude de la façon dont ce système de réputation peut être utilisé pour **prédire la réponse susceptible d'être acceptée (à partir d'une série de réponses) en ce qui concerne une question encore non résolue**.

Notre approche consiste à choisir un sous-ensemble de questions avec leurs réponses respectives, et pour chaque réponse, nous avons créé une paire question-réponse (*quan*). Nous avons ensuite extrait un ensemble de caractéristiques clés à partir de chaque quan, et avons appliqué des techniques supervisées d'apprentissage automatique pour entraîner un classificateur qui a appris, en fonction de ces caractéristiques, peu importe si la réponse acceptée pour cette question était contenue dans un quan. Enfin, nous avons utilisé le classificateur entraîné pour prédire si, selon un quan (relié à une question sans réponse marquée), sa réponse pouvait potentiellement être la réponse acceptée pour la question. Selon nos constatations, le modèle obtenu précédemment a prédit correctement la réponse possible pour chaque question avec une grande exactitude (88 pour cent du temps).

Une question et sa réponse acceptée constituent une source de connaissances de qualité, car elle fournit la solution à un problème précis. Nous proposons d'adopter un forum de questions-réponses semblable pour Bullipedia, et d'appliquer ensuite un modèle de classification semblable pour cerner la meilleure réponse à des questions non résolues.

Mots clés: Site de questions-réponses; Stack Overflow; apprentissage automatique; classification; Linéaire SVC; connaissance; communaut

Introduction

Bullipedia has its origin in elBulli (1962–2011), one of the most lauded restaurants of all time. Michelin awarded it with three stars (1976, 1990, and 1997), and more recently it was voted best restaurant in the world in 2002 and from 2006 to 2009 by industry authority *Restaurant* magazine (Williams 2012). The successful restaurant incorporated disciplines such as technology, science, philosophy, and the arts in its research, and Ferran Adrià, its owner and the most influential chef in the world, published his results in international conferences, books or journal articles, in a similar way to the academic process of peer review.

elBulli has now become elBulliFoundation, a center that seeks to be a hub for creativity and innovation in high cuisine, and continue the creation activity of the former restaurant. The key project of the foundation is an attempt to *externalize* all its wisdom onto the Bullipedia – Adrià's vision for *"an online database that will contain every piece of gastronomic knowledge ever gathered"* (Williams 2012). He justified the need for such a culinary encyclopedia by claiming that *"there is no a clear codification on cuisine"* (Pantaleoni 2013). However, the Bullipedia is an idea yet to be developed. Thus, the question to answer at this point is: *What should the Bullipedia be like?* By analyzing different sources (specific literature, elBulli's publications, interviews with Adrià, news, and emails with elBulliFoundation's staff), we have identified several requirements that the Bullipedia must meet to achieve its mandate. We focus on *encouraging user contribution and holding quality knowledge* in this work.

For a project such as this (creating an online encyclopedia on cuisine) we believe that the collaboration of the community is indispensable to building quality contents. Bullipedia is an idea inherently 2.0 that can take advantage of crowdsourcing (Quinn and Bederson 2011; Doan, Dieter and Harmelen 2011) and harness the collective intelligence (O'Reilly 2005) to generate value. There exist many successful projects that could teach us valuable lessons for Bullipedia. The best example is Wikipedia, the online encyclopedia par excellence, whose success is due largely to its reliance on the crowd to create, edit, and curate its content. Other relevant cases are Allrecipes, Epicurious, and Cookpad, three of the most popular websites for recipe exchange.

On these platforms, the members upload their own recipes and review and rate other members' recipes.

A major concern in these kinds of projects that rely on their own users to succeed is precisely how to engage them. In our previous work (Jiménez-Mavillard and Suárez 2015), we recapitulated the main motivations for the crowd to create content. From them, in the context of this work, we highlight recognition and reputation as intrinsically rewarding factors that motivate the community to collaborate (Herzberg 2008). For this reason, we look into SO and we argue that its reputation system: 1) encourages participation, 2) is an excellent quality control mechanism that guarantees true value because it allows users to collectively decide whether or not contents are reliable, and 3) can be used to measure and predict the quality of these contents.

This paper is organized as follows. In section 2, we describe Q&A sites and in particular SO. In section 3, we outline the problem and significant related works. The experiment and the methodology is detailed in section 4, and the results are shown in section 5. We discuss the relevance of SO and other projects to Bullipedia in section 6, and finally, end with some conclusions and future work in section 7.

Stack Overflow and Q&A sites

Since the origin of the Internet, the volume of information has been increasing on the web, digital libraries and other media. Traditional search engines are helpful tools to tackle the abundance of information, but they just give ranked lists of documents that users must browse manually. In many cases, they simply want the exact answer to their questions, asked in natural language.

Q&A sites have emerged in the past few years as an enormous market to fulfill these information needs. They are online platforms where users share their knowledge and expertise on a variety of subjects. In essence, users of the Q&A community ask questions and other users answer them. These sites go beyond traditional keyword-based querying and retrieve information in more precise form than given by a list of documents. This fact has changed the way people search for information on the web. For instance, the abundance of information to which developers are exposed via social media is changing the way they collaborate, communicate, and

learn (Vasilescu and Serebrenik 2013). Other authors have also investigated the interaction of developers with SO, and reported how this exchange of questions and answers is providing a valuable knowledge base that can be leveraged during software development (Treude, Barzilay and Storey 2011). These changes have produced a marked shift initially from mere websites born to provide useful answers to the question, towards large collaborative production and social computing web platforms aimed at crowdsourcing knowledge by allowing users to ask and answer questions. The end product of such community-driven knowledge creation process is of enduring value to a broad audience; it is a large repository of valuable knowledge that helps users to solve their problems effectively (Anderson et al. 2012).

The ever-increasing amount of Q&A sites has caused the number of questions answered on them to far exceed the number of questions answered by library reference services (Janes 2003), which until recently were one of the few institutional sources for such information. Library reference services have a long tradition of evaluation to establish the degree to which a service is meeting user needs. Such evaluation is no less critical for Q&A sites, and perhaps even more so, as these sites do not have a set of professional guidelines and ethics behind them, as library services do (Shah and Pomerantz 2010). Instead, most Q&A sites use collaborative voting mechanisms for users inside the community to evaluate and maintain high quality questions and answers (Tian et al. 2013). By a quality answer, we mean the one that satisfies the asker (Liu et al. 2008) and also other web users who will face similar problems in the future (Liu et al. 2011).

There are currently a variety of Q&A sites. The first Q&A site was the Korean Naver Knowledge iN, launched in 2002, while the first English-language one was Yahoo! Answers, launched in 2005. Some sites (such as Yahoo! Answers or Quora) are intended for general topics while others (such as Digital Humanities Q&A and SO) are formed by specialized communities centered on specific domains. In this paper, we are focusing on this last Q&A site.

SO is a popular Q&A site in which users ask and answer questions related to programming, web development, operating systems, and other technical topics. This forum was created by Jeff Atwood and Joel Spolsky in 2008, as an alternative

to traditional sources of information for programmers, like books, blogs, or other existing Q&A sites. The fact that both Atwood and Spolsky were popular bloggers contributed to its success in the early stages of the project as they brought their two communities of readers to their new site and generated the critical mass that made it work (Atwood 2008; Spolsky 2008). This success was also promoted by its novel features; it incorporated a voting and Wikipedia-like edition system, and it was open and free, unlike earlier Q&A sites such as Experts-Exchange. Soon, the phenomenon of SO caught the attention of investors and media. The New York Times reported that the Q&A site raised \$6 million in a first round of funding (Ha 2010), and additional rounds would raise further resources: \$12 million in 2011, \$10 million in 2013, and \$40 million in 2015 (Stack Overflow 2016).

The site employs gamification to encourage its participants to contribute (Deterding 2012). Participation is rewarded by means of an elaborate reputation system that is set in motion through a rich set of actions. The main actions are asking and answering questions, but users also can vote up or down on the quality of other members' contributions. The basic mode of viewing content is from the question page, which lists a single question along with all its answers and their respective votes. The vote score on an answer –the difference between the upvotes and downvotes it receives– determines the relative ordering in which it is displayed on the question page. When users vote, askers and answerers gain reputation for asking good questions and providing helpful answers, and they also obtain badges that give them more privileges on the website. In addition, at any point, an asker can select one of the posted answers as the accepted answer, suggesting that this is the most useful response. This also makes the asker and the answerer earn reputation. The reputation score can be seen as a measure of expertise and trust, which signifies how much the community trusts a user (Tian et al. 2013).

SO's success is largely due to the engaged and active user community that collaboratively manages the site. This community is increasing both in size and in the amount of content it generates. According to the January 2014 SO data dump provided by the Stack Exchange network (and analyzed in this work), SO stores around 9.5 million questions, almost 16 million answers, and has a community with

more than 4 million users. The number of questions added each month has been steadily growing since the inception of SO (Ponzanelli et al. 2014) and has reached peaks of more than 200,000 new questions per month (see **Figure 1**).

The content is heavily curated by the community; for example, duplicate questions are quickly flagged as such and merged with existing questions, and posts considered to be unhelpful (unrelated answers, commentary on other answers, etc.) are removed. As a result of this self-regulation, and despite its size, content on SO tends to be of very high quality (Anderson et al. 2012).

Problem and related work

The primary idea of this study is to understand the relation between a question and its accepted answer in order to predict potential accepted answers (from a set of candidate answers) for new questions. We tackled this problem by analyzing SO to verify if the metrics associated to the reputation system activities – question's score, answer's score, user's reputation, etc. – can decisively predict accepted answers for yet unresolved questions.

This problem has two main components: crowdsourcing and machine learning. The term *crowdsourcing* is the combination of the words *crowd* and *outsourcing*, and

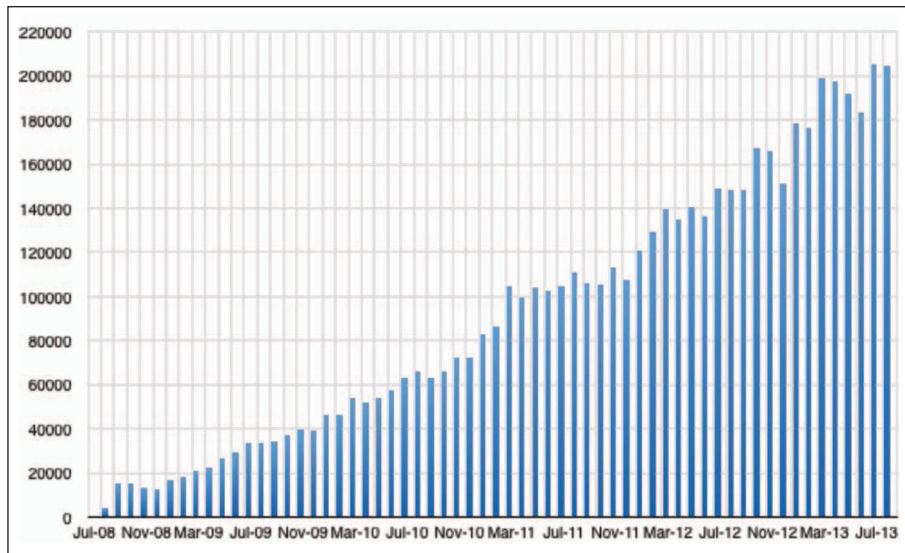


Figure 1: Number of questions by month (Ponzanelli et al. 2014).

is the process of getting work from a large group of people, especially from online communities, rather than from employees or suppliers. This model of contribution has been applied to a wide range of disciplines, from bioinformatics (Khare et al. 2015) to the digital humanities (Carletti et al. 2013). “The Wisdom of Crowds” (Surowiecki 2005) is a popular science work for a general introduction to the concept. “Crowdsourcing Systems on the World-Wide Web” (Doan, Ramakrishnan and Halevy 2011) and “Human Computation: A Survey and Taxonomy of a Growing Field” (Quinn and Bederson 2011) give a scholarly broad review of the field. The former is focused on the Web as the natural environment for crowdsourcing. The latter emphasizes the power of humans to undertake tasks that computers cannot yet do effectively.

SO relies on the crowd, and some authors underlined the soundness of this user-generated content model to provide quality solutions. Vasilescu and Serebrenik (2013) investigated the correlation between the activity of SO's users and their activity on GitHub, the largest social coding site on the Web. They demonstrated that the most productive programmers, in terms of amount and uniform distribution of their work, are the ones who answer more questions on the Q&A site. Therefore, a large number of answers on SO are presumably effective solutions, as they came from qualified programmers that incorporate good work practices.

Parnin, Treude and Grammel (2012) showed that companies like Google, Android, or Oracle also acknowledge the quality of contents produced on SO. These brands entrusted the documentation of their respective APIs – Google Web Toolkit, Android, and Java – to the SO community. The authors collected usage data using Google Code Search (currently shut down), and analyzed the coverage, quality, and dynamics of the SO documentation for these APIs. They found that the crowd is capable of generating a rich source of content with code examples and discussion that is more actively viewed and used than traditional API documentations.

The second pillar of this work is *machine learning*, a subfield of artificial intelligence that studies how to create algorithms that can learn and improve with experience. These algorithms learn from input observations, build a model that fits the observed data, and apply the model to new observations in order to make predictions on them. Machine learning is a transverse area used in a large number

of disciplines, and with multiple applications, from computer vision to speech and handwriting recognition. "Machine Learning" (Mitchell 1997) is a classic introductory textbook on primary approaches to the field.

A recurrent problem solved with machine learning is classification. This is the problem of identifying the category of a new observation, and belongs to the type of supervised methods, that is, the task of building a model from categorized data. As the goal of our study is to determine when a question has been correctly answered, we posed this question as a classification problem: "Is this answer (likely to be) the accepted answer for this question?" The possible options are "yes" or "no." Therefore, for every question each of its answers belongs to one of these two categories: "Yes" (it is the accepted answer) or "No" (it is not). The question seems trivial but even in many branches of pure mathematics, where specialists deal with objective and universal knowledge, it can be surprisingly hard to recognize when a question has, in fact, been answered (Wilf 1982).

While it is true that finding the "right" answer is ambitious, efforts to detect 'good enough' ones are underway. As already mentioned above, our approach is to extract features from questions, answers, and users, and apply classification to learn the relation between a question and its accepted answer. Many authors have applied machine learning techniques to provide the correct answer to a question by pursuing different objectives. For instance, some of them focused on directly identifying the best answer. Wang et al. understood questions and their answers on Yahoo! Answers as relational data (Wang et al. 2009). They assumed that answers are connected to their questions with various types of links that can be positive (indicating high-quality answers), negative (indicating incorrect answers) or user-generated spam. They proposed an analogical reasoning-based approach which measures the analogy between the new question-answer linkages and those of some previous relevant knowledge that contains only positive links. The answer that had the most analogous link to the supporting set was assumed to be the best answer. Shah and Pomerantz instead, evaluated and predicted the quality of answers, also on Yahoo! Answers (Shah and Pomerantz 2010). They extracted features from questions, answers, and the users that posted them, and trained different classifiers that were able to measure

the quality of the answers. The answer with the highest quality was considered the best one. Interestingly, the authors reported that contextual information such as a user's profile (they included information like the number of questions asked and the number of those questions resolved, for the asker; and the number of questions answered and the number of those answers chosen as the best answers, for the answerer) can be critical in evaluating and predicting content quality. This is actually a key finding in our experiment, as we will see in the results (section 5).

Another approach is to redirect the question to the best source of information. For example, Singh and Shadbolt matched questions on SO to the corresponding Wikipedia articles, so that the users could find out the answer by themselves (Singh and Shadbolt 2013). For this, they applied natural language processing tools to extract the keywords of the question, and then they matched them to the keywords of the pertinent Wikipedia article. Tian et al. addressed the problem of routing a question to the appropriate user who can answer it (Tian et al. 2013). They proposed an approach to predict the best answerer for a new question on SO. Their solution considered both the user's interests (topics learned by applying topic modeling to previous questions answered by the user) and the user's expertise (topics learned by leveraging the collaborative voting mechanism) relevant to the topics of the given question.

Similar to Q&A sites, we have Q&A systems. They are systems that receive a question in natural language and return small snippets of text that contain an answer to the question (Voorhees and Tice 1999). Related to these systems, Rodrigo et al. evaluated the correctness of answers as a classification problem (Rodrigo, Peñas and Verdejo. 2011). They studied two methods, F-score and ROC, and compared both evaluation approaches. Harabagiu and Hickl considered that the semantics of an answer is a logical entailment of the semantics of its question, and used this idea to enhance the accuracy of current open-domain automatic Q&A systems (Harabagiu and Hickl 2006).

Less relevant to our work, machine learning can be applied to classify questions instead of answers. Ponzanelli et al. presented an approach to automate the classification of questions on SO according to their quality (Ponzanelli et al. 2014). They investigated how to model and predict the quality of a question by

considering both the contents of a post (from simple textual features to more complex readability metrics) and community-related aspects (for example, the popularity of a user in the community) as features. Other authors analyzed the problem of question classification by topic. Miao et al. compared two approaches, K-means and PLSA, to automatically classify questions on Yahoo! Answers, and reported the importance of incorporating user information to improve both methods (Miao et al. 2010). Blooma et al. also applied a machine learning solution but a different classifier, Support Vector Machine (SVM) (Blooma et al. 2008). Li and Roth developed a semantic hierarchical classifier guided by an ontology of answer types (Li and Roth 2006).

Experiment and methodology

As stated before, we formulated the idea of identifying valuable knowledge on SO as a classification problem in machine learning. In particular, we made use of SO's reputation system to predict if an answer will be the accepted answer for a question. Let us have a question, q_p , that has k answers, $a_{i1}, a_{i2}, \dots, a_{ik}$, and none of them has been marked as accepted yet. Let us pair the question with each of its answers; we obtain k pairs: $\langle q_p, a_{i1} \rangle, \langle q_p, a_{i2} \rangle, \dots, \langle q_p, a_{ik} \rangle$. The problem we want to solve is: "what question-answer pair contains the answer that will be (likely to be) accepted?" In the context of this paper, we defined the next concepts:

- **Information unit** or **quan**: a quan is a question-answer pair. Each answer on SO has a minimum level of accuracy with respect to its question; in any other case, it would have been removed by the community. Therefore, every quan provides always valid information.

$$quan = \langle question, answer \rangle$$

In the previous example, $\langle q_p, a_{i1} \rangle, \langle q_p, a_{i2} \rangle, \dots, \langle q_p, a_{ik} \rangle$ are quans.

- **Knowledge unit** or **kuan**: a kuan is a particular quan formed by a question and its accepted answer. The accepted answer solved the asker's question. Hence, a kuan is a source of valuable knowledge for other users that could face the same problem in the future.

$$kuan = \langle question, accepted_answer \rangle$$

In the previous example, if the first answer (for instance), a_{i1} , was the accepted answer, then from the list of quans, $\langle q_i, a_{i1} \rangle$, $\langle q_i, a_{i2} \rangle$, ..., $\langle q_i, a_{ik} \rangle$, only $\langle q_i, a_{i1} \rangle$ would be also a kuan.

Formally, let Q be the set of questions and n its cardinality, i.e., let us have n questions:

$$Q = \{q_1, q_2, \dots, q_n\}$$

Let A be the set of answers and m its cardinality, i.e., let us have m answers (for this definition, we are considering the total answers independently from their questions):

$$A = \{a_1, a_2, \dots, a_m\}$$

And let ans be the function that returns the answers for a question and k_i the number of answers for the i th question (q_1 has k_1 answers, q_2 has k_2 answers, etc.), i.e., ans is a function that takes a question from the set of questions, Q , and returns all the answers from the set of answers, A , that are the answers of the question:

$$ans: Q \rightarrow A^k$$

$$ans(q_i) = \{a_{i1}, a_{i2}, \dots, a_{ik_i}\} \mid \forall i: 1 \leq i \leq n$$

Let us group the total m answers in n subsets of answers (one subset for each of the n questions). Each question may have a different number of answers with respect to the other questions:

$$ans(q_1) = \{a_{11}, a_{12}, \dots, a_{1k_1}\}$$

$$ans(q_2) = \{a_{21}, a_{22}, \dots, a_{2k_2}\}$$

...

$$ans(q_n) = \{a_{n1}, a_{n2}, \dots, a_{nk_n}\}$$

Then, A can be redefined as follows:

$$A = \{ans(q_1), ans(q_2), \dots, ans(q_n)\}$$

Let QA be the set of quans (i.e., QA is the set of pairs formed by each question and each of its answers):

$$QA = \{\langle q, a \rangle \mid q \in Q \wedge a \in ans(q)\}$$

Or equivalently:

$$QA = \{\langle q_1, a_{11} \rangle, \dots, \langle q_1, a_{1k_1} \rangle, \langle q_2, a_{21} \rangle, \dots, \langle q_2, a_{2k_2} \rangle, \dots, \langle q_n, a_{n1} \rangle, \dots, \langle q_n, a_{nk_n} \rangle\}$$

Let us note that the total number of quans is equals to the number of answers:

$$|QA| = |A| = m$$

Let KA be the set of kuans (i.e., the set of quans that contain the accepted answer):

$$KA = \{qa \mid qa \in QA \wedge qa.answer.accepted = True\}$$

Or equivalently, if a_{ji} is the accepted answer for the question q_i , then KA can be redefined as follows:

$$KA = \{\langle q_1, a_{1j_1} \rangle, \langle q_2, a_{2j_2} \rangle, \dots, \langle q_n, a_{nj_n} \rangle\}$$

Let us note that the KA is a subset of QA, i.e., only some quans are kuans; and that there are, at most, as many kuans as questions (every question can has 0 or 1, but no more accepted answers):

$$KA \subseteq QA$$

$$|KA| \leq |Q| = n$$

The following example illustrates the previous concepts. Let us have two questions, $Q = \{q_1, q_2\}$, two answers for q_1 , $ans(q_1) = \{a_{11}, a_{12}\}$, and three answers for q_2 , $ans(q_2) = \{a_{21}, a_{22}, a_{23}\}$; then we obtain five quans: $QA = \{\langle q_1, a_{11} \rangle, \langle q_1, a_{12} \rangle, \langle q_2, a_{21} \rangle, \langle q_2, a_{22} \rangle, \langle q_2, a_{23} \rangle\}$. Let a_{11} and a_{22} be the accepted answers for q_1 and q_2 , respectively; then we obtain two kuans: $KA = \{\langle q_1, a_{11} \rangle, \langle q_2, a_{22} \rangle\}$.

The rephrased goal now is to answer the question: “*what quans are kuans?*” In order to address this experiment, we combined different tools: the Ubuntu command line, Python, and in particular its Element Tree XML API, Pandas, and Scikit Learn. Python is a programming language that allowed us to work quickly and integrate systems more effectively. The Element Tree module implements a simple and efficient API for parsing and creating XML data. Pandas is an open source, BSD-licensed library that provides high-performance, easy-to-use data structures and data analysis tools for Python. Scikit Learn is a collection of simple and efficient tools for data mining, data analysis, and machine learning. Our experiment was performed in several steps:

a) SO's data dump

The first step was to dump SO's data on posts and users. With the Ubuntu command line, we split the posts into questions and answers, and then used the Element Tree XML API to parse and extract the data from questions, answers, and users. **Table 1** summarizes the information of the total data dump.

b) Experiment dataset

Secondly, we selected a subset of questions with their respective answers and their authors (users that posted these questions and answers). The subset of questions ranged from January 1 to January 10, 2015, while we collected their answers posted during the whole month of January, 2015. One month from posting a question is enough time to get most of their answers. In fact, 63.5% of questions on SO are answered in less than an hour (Bhat et al. 2014). We used Pandas to process this large data set and store it in an easy to use format: CSV.

c) Feature selection

Third, we created the set of quans from the selected dataset. We needed to transform our data into a suitable representation that the classifier could process, so to accomplish this task, every quan was represented as a vector of features (these came from questions' and answers' attributes as represented in **Table 3**). **Table 2** summarizes the information of our new dataset.

We selected the features based on similar previous works. Features 1 and 3 were proposed by Anderson et al. for questions, and we extended them for answers in 2 and 4 (Anderson et al. 2012). Shah and Pomerantz (2010) suggested the use of 5. Feature 6 was proposed by several sources (Ponzanelli et al. 2014; Movshovitz-Attias et al. 2013; Anderson et al. 2012). Features 7 and 9 were also suggested by others (Shah and Pomerantz 2010; Ponzanelli et al. 2014; Bhat et al. 2014); then, we extended them

Table 1: SO's database dump (2008–2015).

data	#	size (GB)	format
questions	~9,500,000	20.1	XML
answers	~15,800,000	14.5	XML
users	~4,300,000	1.1	XML

Table 2: Experiment dataset (as in January, 2015).

data	#	size (MB)	format
questions	~50,000	36.0	CSV
answers	~70,000	34.8	CSV
users	~50,000	5.7	CSV
quans	~70,000	72.1	CSV

Table 3: Features.

Question-based	Answer-based
1. q score	2. a score
3. qer reputation	4. aer reputation
5. % qer answered questions	6. % aer accepted answers
7. # q comments	8. # a comments
9. # q code lines	10. # a code lines
11. # views	12. # answers
	13. accepted
Description	
1.	Question's score (difference between up and downvotes)
2.	Answer's score (difference between up and downvotes)
3.	Asker's reputation
4.	Answerer's reputation
5.	Asker's percentage of answered questions
6.	Answerer's percentage of accepted answers
7.	Question's total number of comments
8.	Answer's total number of comments
9.	Question's total number of code lines
10.	Answer's total number of code lines
11.	Count of total number of views for the question
12.	Count of total number of answers for the question
13.	The answer is marked as accepted by the asker

for 8 and 10. Lastly, feature 11 was proposed by Anderson et al. (2012), and 12, by Anderson et al. (2012), Shah and Pomerantz (2010), and Wang et al. (2009).

d) Feature extraction

Next, we needed to classify the set of quans into two categories: “Yes” (as in “Yes, it contains the accepted answer”) for kuans and “No” (as in “No, it does not contain the accepted answer”) for the rest of the quans that are not kuans. Feature extraction consisted in transforming arbitrary data, such as text or images, into numerical features usable for machine learning. In our case, a quan is represented by a Python-dictionary-like object composed of 13 feature-value pairs. As previously defined, if QA is the set of quans and m its cardinality, then:

$$QA = \{qa_i \mid \forall i : 1 \leq i \leq m\}$$

Where qa_i is each of the total m quans, decomposed into its thirteen features with their respective values:

$$qa_i = \langle \langle f_1, v_{1i} \rangle, \langle f_2, v_{2i} \rangle, \dots, \langle f_{13}, v_{13i} \rangle \rangle$$

As an instance, this is the quan number 44,366:

```
qa44,366 = <<q score, 5>, <a score, 3>, <qer reputation, 105>, <aer reputation,
45,971>,
<% qer answered questions, 100%>, <% aer accepted answers, 31%>,
<# q comments, 8>,
<# a comments, 0>, <# q code lines, 22>, <# a code lines, 30>, <# views,
308>, <# answers, 2>,
<accepted, True>>
```

e) K-fold cross-validation

The aforementioned set of feature vectors is required to train the classifier. In the basic approach, the total set is split into two sets: training set (usually 90% of the original set) and testing set (the remaining 10%). Then, the classifier is trained with the training set and tested with the testing set. In the k-fold cross-validation approach, the training set is split into k smaller sets, and the following procedure is followed for each of the k folds: first, a model is trained using k-1 of the folds as training data; and second, the resulting model is validated on the remaining part of the data (i.e., it is used as a test set to compute performance measures such as accuracy). The performance measure reported by k-fold cross-validation is then the average of the k values computed in the loop.

f) Classifier training and testing

Finally, we did 10-fold cross-validation on our set of 72,847 quans and got 10 subsets of 7,285 quans. Approximately, one third of the quans are kuans (containing an accepted answer). We trained and tested several classifiers with different parameters. These classifiers were: Ridge, Perceptron, Passive-Aggressive, Stochastic Gradient Descent (SGD), Nearest Centroid, Bernoulli Naive Bayes (NB), and Linear Support Vector Classification (SVC). Of these, the classifier with best performance was Linear SVC.

Linear SVC belongs to the family of SVM, a set of supervised learning methods used for classification, among other applications. SVM has shown good performance for many natural language related applications, such as text classification (Joachims 2002), and has been used in multiple studies relating to question classification (Bloomer et al. 2008; Tamura, Takamura and Okumura 2005; Solorio et al. 2004; Zhang and Lee 2003).

Results

Table 4 summarizes the performance of all the classifiers evaluated in our experiment. The results show that Linear SVC was the classifier with highest performance (for the four metrics) in comparison to the rest.

Table 5 displays the Linear SVC performance in detail. The evaluation was done on the testing set (7,285 quans). We can see that the classifier performed slightly better for “No” than for “Yes” (first and second rows respectively). The third row shows the average values for the total testing set of quans. Our model gave an accuracy of 88%, that is, it predicted correctly the accepted answer for 88% of the questions. This result is superior to others reported in similar works. Shah and Pomerantz (2010) measured the quality of answers on Yahoo! Answers with an accuracy of 84%, while Wang et al. (2009) identified the best answer for new questions, also on Yahoo! Answers, with a precision of 78%. **Table 6** is the confusion matrix of the model. It represents the real number of “Yes” and “No”, and how Linear SVC classified them.

Table 4: Classifier performance comparison.

<u>classifier</u>	<u>accuracy</u>	<u>precision</u>	<u>recall</u>	<u>f1-score</u>
Ridge	0.63	0.59	0.63	0.51
Perceptron	0.53	0.69	0.53	0.51
Passive-Aggressive	0.80	0.80	0.80	0.80
SGD	0.50	0.72	0.49	0.44
Nearest Centroid	0.62	0.59	0.62	0.58
Bernoulli NB	0.78	0.79	0.78	0.78
Linear SVC	0.88	0.88	0.88	0.88

Table 5: Linear SVC performance.

set	#	accuracy	precision	recall	f1-score
No	4,590	–	0.89	0.90	0.90
Yes	2,695	–	0.84	0.82	0.83
Total	7,285	0.88	0.87	0.87	0.87

Table 6: Confusion matrix.

	Predicted No	Predicted Yes	Total True
True No	4,121	441	4,562
True Yes	485	2,238	2,723
Total Predicted	4,608	2,677	7,285

For visualization purposes, we also calculated the two most informative features (the two most helpful features for the classifier to classify the quans) and they were *percent_answered_questions_q* and *percent_accepted_answers_a*, i.e., the percentage of questions that were answered to this question's asker (with respect to the total number of questions posted by this asker) and the percentage of answers that were accepted for this answer's answerer (with respect to the total number of answers posted by this answerer), respectively. We trained a new classifier with only these two features and obtained an average accuracy of 82% (86% for "No" and 76% for "Yes"). This result is unsurprisingly inferior to the previous one (88%) as we left out other important features, but it demonstrated that the two most informative features alone provide enough information to the classifier for it to classify the quans with fairly high accuracy. These results are shown in **Figure 2**.

Figure 2 displays the space of quans created by the new limited model. Quans (dots) are drawn in a space that is divided into two physically differentiated areas: red for "Yes" and blue for "No." The red area corresponds to high values for the two features – *percent_answered_questions_q* and *percent_accepted_answers_a* – and takes up approximately one third of the total space, while the blue area represents lower values for the features (the origin is the bottom-left corner). Analogously, every dot (quan) is either red (for "Yes") or blue (for "No"). This color indicates the true category of the quan, whereas its position in the space means the category assigned

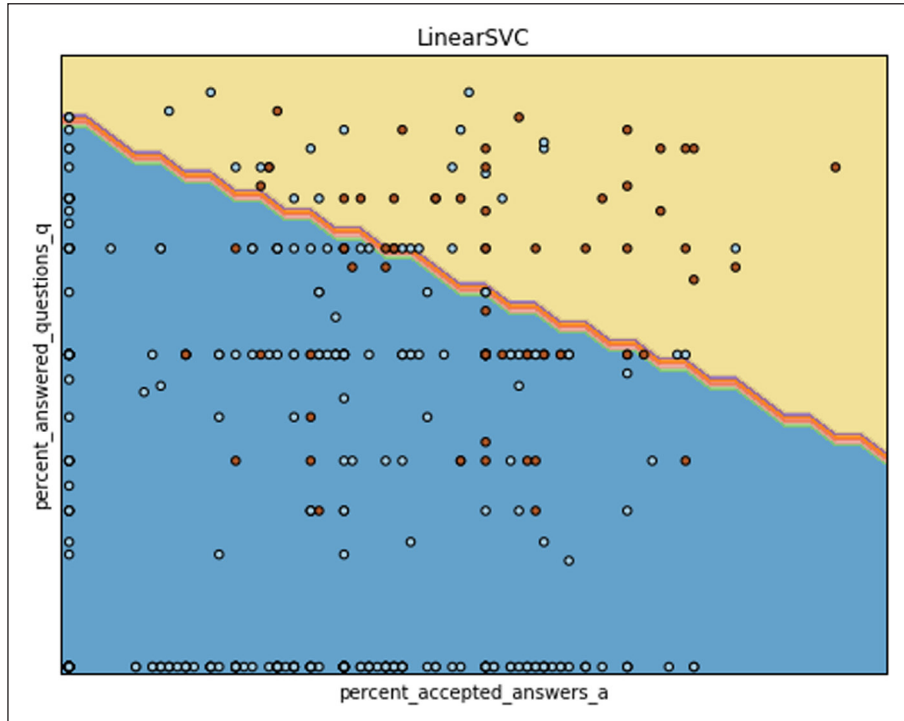


Figure 2: Space of quans: truth vs. prediction.

by the classifier. The higher the values for the features, the higher their probability to fall into the red area. Thus, in an ideal scenario (accuracy of 100%), every quan would have fallen into its correct category (red dots and blue dots into red and blue areas respectively). However, the image shows that only 86% of blue dots fell into the blue area and, similarly, only 76% of red dots do it in the red area. This asymmetry was already observed in **Table 5**. Let us remember that this model only uses the two most informative features, and that the performance of the classifier improves (up to 88%) when we consider the whole set of features.

Finally, our experiment confirms Shah and Pomerantz's (2010) results. They reported that contextual information such as a user's profile (information like the number of questions asked and the number of those questions resolved, and the number of questions answered and the number of those answers chosen as the best answers) can be critical in evaluating and predicting content quality.

Discussion

The election of SO as a case study was not arbitrary. It could be argued that, against the backdrop of an online gastronomic encyclopedia, we should have looked into projects such as Wikipedia, Allrecipes, Epicurious, or Cookpad, that are, at a first glance, more relevant to Bullipedia than a programming forum like SO. Wikipedia is the most representative example of social knowledge on the Web. Its members self-organize and knowledge emerges from the bottom-up as a result of their interactions. It has a quality mechanism based on constant supervision by its own community, which ensures that contents are accurate at all times. On Wikipedia, articles get better and better over time as everybody contributes their knowledge, and therefore editions are less and less frequent as the content achieves a certain degree of consensus. This is what we call *knowledge maturity*: when the frequency of editions is lower than some threshold.

Despite its suitability, Wikipedia does not provide enough quantifiable data about contents and users. On one hand, the number of article views could be used as an indicator of the popularity of the topic but not for the quality of its content. On the other hand, user pages describe their remarkable activities and other recognitions, but their contribution for a particular article is difficult to measure. A good indicator for quality could be the life cycle of the contents in an article; however, this is not a trivial task. We would have to examine the revision history page, track its passages, calculate for how long every passage was or has been alive, and then create a measurement based on the passages that are currently active and belong to the article.

Allrecipes, Epicurious, and Cookpad are cooking websites where users create and publish their own recipes. Although these sites should have all the ingredients to be our model, they actually lack the social dimension needed for this study. Cookpad let the user favorite a recipe and comment on it, but there is no other consequence from this interaction beyond the favorite count for the recipe. Epicurious allows one to rate and write a review for a recipe but no other user can evaluate these reviews; therefore, there is no way to know how reliable they are. The same thing occurs on

Allrecipes. Without a mechanism to evaluate the goodness of contributions, users' feedback are just opinions – there is no evidence that they help to improve the original recipe.

SO is, however, a place where real problems meet their solutions. Although Q&A sites have a different nature from an online encyclopedia, there are some significant reasons that make SO relevant to the case of Bullipedia:

1. **Topic.** Food and computing seem to belong to distant worlds; however, there are important analogies between recipes and algorithms (Bultan 2012). Numerous researchers have proposed methods for converting a recipe text to a work flow by following different approaches – recipe feature extraction (Yamakata et al. 2013), machine learning (Mori et al. 2012), or domain constraints (Hamada et al. 2000).
2. **Popularity.** SO is one of the most popular Q&A based knowledge sharing communities on the web (Movshovitz-Attias et al. 2013), from which we can obtain valuable lessons for the Bullipedia.
3. **Reputation system.** A huge, active, and engaged community would be desirable to build up trust and quality content for the future Bullipedia, and a SO-like reputation system can help to fit this need. Recognition and reputation are intrinsically rewarding factors that motivate the community to collaborate (Herzberg 2008). Also, active users and reputed users encourage other users to contribute (Jiménez-Mavillard and Suárez 2015). Furthermore, building up trust is one of the major motivations for information exchange (Barachini 2009; Krogh, Roos and Kleine 1998). Thanks to this involvement, good questions and answers are easily identified by the community itself, and unhelpful posts are removed. This keeps the quality of content very high (Anderson et al. 2012). At the same time, the level of trust in a community and the value of the knowledge that it generates are also important factors in users' willingness to collaborate (Krogh, Roos and Kleine 1998; Tsai 2000). All of this activity combines into a self-perpetuating cycle (**Figure 3**).

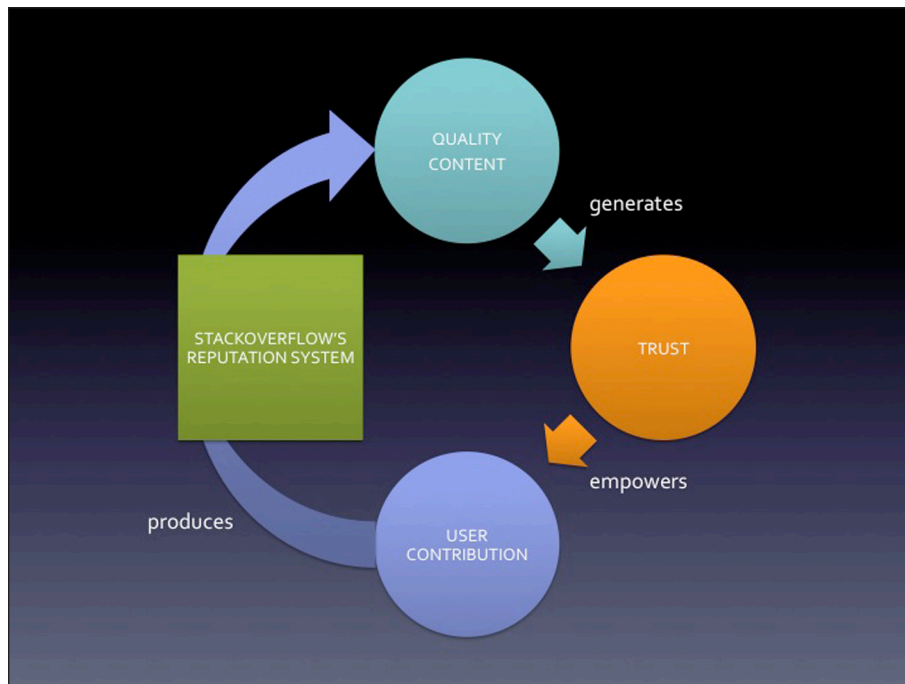


Figure 3: SO's reputation system, contribution, quality, and trust.

The viability of Bullipedia depends on how we face some big challenges, namely, how to engage the community that will build the content, how to make this content quality knowledge, or how to compete with established recipe websites. As mentioned above, the reputation system fits the quality knowledge needs and the rewarding factors that motivate people to contribute; but social factors play a role equally decisive. Some of these are simply the common good or less altruistic motivators such as recognition and career advancement. Reputation is translated into expertise, which is very valuable for the development industry in the case of SO. Only time will tell if a similar reputation system on Bullipedia will be helpful in identifying the next generation of best chefs in the world. Bullipedia will likely also lure users away from other recipe websites, since Bullipedia has an important unfair advantage in its connection to elBulli and Ferran Adrià's trademark. This is a project that has raised high expectations since its conception among culinary professionals and media. Moreover, Bullipedia is different from its main competitors because it will be more than a recipe exchange website – it will be a platform for recipe and

knowledge exchange, and a forum for people to solve their questions, with a true social component incentivized with gamification.

Conclusions and future work

In this work, we have studied SO, a popular Q&A site for programming that owes its success to its committed community. This commitment is achieved by means of an elaborate reputation system and its triple role: 1) it is an implementation of the gamification employed by the site to encourage participation, 2) it is a collective mechanism to control the quality of the crowdsourced contents, and 3) it stores a reputation score for every user, that is, a measurement of their expertise on the site. These scores provide useful metrics to evaluate the quality of the contents. Thus, we contemplated using this reputation system to prognosticate the quality of the answers. In particular, we wondered whether it was possible to predict the likely *accepted* answer (from a set of candidate answers) for a yet unresolved question.

We formulated this issue as a machine learning classification problem: “for every quan (question-answer pair), is that answer (likely to be) the accepted answer for that question?” We gave two possible categories: “Yes” or “No”. In order to solve this problem, we first represented every quan as a feature vector; and then did a 10-fold cross-validation on our set of quans and trained and tested several classifiers with different parameters. The classifier that performed best was Linear SVC. Our key finding shows that this model correctly predicted the accepted answer for every question with an accuracy of 88%. This result is substantially higher than others reported in similar studies. Shah and Pomerantz (2010) measured the quality of answers on Yahoo! Answers with an accuracy of 84%, while Wang et al. (2009) identified the best answer for new questions, also on Yahoo! Answers, with a precision of 78%.

Our prediction is based on the thirteen extracted features for each quan, but the two most informative features were *percent_answered_questions_q* and *percent_accepted_answers_a* (the percentage of resolved questions posted by the asker and the percentage of accepted answers posted by the answerer, respectively). These two metrics suggest the importance of asking clear question in obtaining an answer and giving good answers in having them accepted. When a question is asked by a *good*

asker and the answer is provided by a *good answerer*, the probability for the question and the answer to form a kuan (question-accepted_answer pair) increase rapidly (to 82% according to our second experiment and up to 88% according to our first one). These results reaffirm Shah and Pomerantz's (2010) findings – they reported that users' information is critical in evaluating and predicting content quality.

A question and its accepted answer constitute reliable knowledge, as it provides the solution for a specific problem that a user had in the past and that other users may face in the future. From this reliable knowledge (by identifying questions and their accepted answers) we can build a repository that contains exclusively quality knowledge on SO. This idea can be extrapolated to every possible domain. We have demonstrated that Linear SVC is suitable for Q&A classification, so, if we implemented a similar forum and reputation system on the future Bullipedia, it would be possible to apply this same idea to predict the best answer for unsolved questions on the gastronomic encyclopedia. Questions like “*Is it safe to cook chicken in a slow-cooker?*” or “*What's a good dressing for a salmon salad?*” would have a best answer because our classifier would select it from all the supplied answers.

We believe that a Q&A forum with gamification implemented by a reputation system, plus elBulli and Adrià's trademark, will form a delightful combination that would raise high expectations and engage a large enthusiastic community. This engagement would create quality knowledge and increase the potential of the Bullipedia project, from which the next generation of best chefs in the world might arise. In the future, we are planning to research if it is also feasible to use the reputation system to identify influential users in the community, like well-known chefs, other users with high reputations, *good askers* or *good answerers*, and improve the process of knowledge creation by analyzing the habits of these key actors.

Competing Interests

The authors have no competing interests to declare.

References

- Anderson, Ashton, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec.** 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, 850–858. KDD 2012. New York: ACM. DOI: <https://doi.org/10.1145/2339530.2339665>

Atwood, Jeff. 2008. "Introducing Stackoverflow.com." *Horror Coding*, April 16. <https://blog.codinghorror.com/introducing-stackoverflow-com/>.

Barachini, Franz. 2009. "Cultural and Social Issues for Knowledge Sharing." *Journal of Knowledge Management* 13(1): 98–110. DOI: <https://doi.org/10.1108/13673270910931198>

Bhat, Vasudev, Adheesh Gokhale, Ravi Jadhav, Jagat Pudipeddi, and Leman Akoglu. 2014. "Min(e)d your Tags: Analysis of Question Response Time in StackOverflow." In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 328–335. IEEE, 2014. DOI: <https://doi.org/10.1109/ASONAM.2014.6921605>

Blooma, Mohan J., Dion Hoe-Lian Goh, Alton Y. K. Chua, and Zhiquan Ling. 2008. "Applying Question Classification to Yahoo! Answers." In *Applications of Digital Information and Web Technologies*, 229–234. ICADIWT 2008. IEEE, 2008. DOI: <https://doi.org/10.1109/ICADIWT.2008.4664350>

Bultan, Tevfik. 2012. "Recipes for Computing." *What Is Computing?* May 2. <http://whatiscomputing.blogspot.ca/2012/05/recipes-for-computing.html>.

Carletti, Laura, G. Giannachi, D. Price, D. McAuley, and S. Benford. 2013. "Digital Humanities and Crowdsourcing: An exploration." In *MW2013: Museums and the Web 2013*. Portland: Museums and the Web. <https://ore.exeter.ac.uk/repository/handle/10871/17763>.

Deterding, Sebastian. 2012. "Gamification: Designing for Motivation." *Interactions* 19(4): 14–17. DOI: <https://doi.org/10.1145/2212877.2212883>

Doan, Anhai, Raghu Ramakrishnan, and Alon Y. Halevy. 2011. "Crowdsourcing Systems on the World-Wide Web." *Communications of the ACM* 54(4): 86–96. DOI: <https://doi.org/10.1145/1924421.1924442>

Ha, Anthony. 2010. "Stack Overflow Raises \$6 M to Take Its Q&A Model Beyond Programming." *The New York Times*, May 5. Accessed Nov. 17, 2016. <http://www.nytimes.com/external/venturebeat/2010/05/05/05venturebeat-stack-overflow-raises-6m-to-take-its-qa-mode-31396.html>.

- Hamada, Reiko, Ichiro Ide, Shuichi Sakai, and Hidehiko Tanaka.** 2000. "Structural Analysis of Cooking Preparation Steps in Japanese." In *Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages*, 157–64. IRAL '00. New York: ACM. DOI: <https://doi.org/10.1145/355214.355237>
- Harabagiu, Sanda and Andrew Hickl.** 2006. "Methods for Using Textual Entailment in Open-domain Question Answering." In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, 905–912. ACL-44. Stroudsburg: Association for Computational Linguistics. DOI: <https://doi.org/10.3115/1220175.1220289>
- Herzberg, Frederick.** 2008. "One More Time: How Do You Motivate Employees?" *Harvard Business Review*, September–October.
- Janes, Joseph.** 2003. "The Global Census of Digital Reference." *Paper presented at the 5th Annual VRD Conference*. San Antonio, Texas, November 17–18.
- Jiménez-Mavillard, Antonio, and Juan-Luis Suárez.** 2015. "From Taste of Home to Bullipedia: Collaboration, Motivations and Trust." *Digital Studies/Le Champ Numérique*. Accessed December 11, 2016. https://www.digitalstudies.org/ojs/index.php/digital_studies/article/view/311/408.
- Joachims, Thorsten.** 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Norwell: Kluwer Academic Publishers. DOI: <https://doi.org/10.1007/978-1-4615-0907-3>
- Khare, Ritu, Benjamin M. Good, Robert Leaman, Andrew I. Su, and Zhiyong Lu.** 2015. "Crowdsourcing in Biomedicine: Challenges and Opportunities." *Briefings in Bioinformatics*, 1–10. DOI: <https://doi.org/10.1093/bib/bbv021>
- Krogh, Georg von, Johan Roos, and Dirk Kleine.** 1998. *Knowing in Firms: Understanding, Managing and Measuring Knowledge*. London: SAGE Publications Ltd. <http://knowledge.sagepub.com/view/knowning-in-firms/SAGE.xml>. DOI: <https://doi.org/10.4135/9781446280256>
- Li, Xin, and Dan Roth.** 2006. "Learning Question Classifiers: The Role of Semantic Information." *Nat. Lang. Eng.* 12(3): 229–249. DOI: <https://doi.org/10.1017/S1351324905003955>

- Liu, Qiaoling, Eugene Agichtein, Gideon Dror, Evgeniy Gabrilovich, Yoelle Maarek, Dan Pelleg, and Idan Szpektor.** 2011. "Predicting Web Searcher Satisfaction with Existing Community-based Answers." In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 415–424. SIGIR '11. New York: ACM. DOI: <https://doi.org/10.1145/2009916.2009974>
- Liu, Yandong, Jiang Bian, and Eugene Agichtein.** 2008. "Predicting Information Seeker Satisfaction in Community Question Answering." In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 483–490. SIGIR '08. New York: ACM. DOI: <https://doi.org/10.1145/1390334.1390417>
- Miao, Yajie, Lili Zhao, Chunping Li, and Jie Tang.** 2010. "Automatically Grouping Questions in Yahoo! Answers." In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)* 1: 350–357. IEEE, 2010. DOI: <https://doi.org/10.1109/WI-IAT.2010.157>
- Mitchell, Tom M.** 1997. *Machine Learning*. McGraw-Hill.
- Mori, Shinsuke, Tetsuro Sasada, Yoko Yamakata, and Koichiro Yoshino.** 2012. "A Machine Learning Approach to Recipe Text Processing." In *Proceedings of The CookingwithComputersWorkshop(CWC)*, edited by Amélie Cordier, and Emmanuel Nauer, 29–35. Accessed December 12, 2016. http://www.lirmm.fr/ecai2012/images/stories/ecai_doc/pdf/workshop/W29_proceedingsCWC2012.pdf.
- Movshovitz-Attias, Dana, Yair Movshovitz-Attias, Peter Steenkiste, and Christos Faloutsos.** 2013. "Analysis of the Reputation System and User Contributions on a Question Answering Website: StackOverflow." In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 886–893. ASONAM '13. New York, NY, USA: ACM. DOI: <https://doi.org/10.1145/2492517.2500242>
- O'Reilly, Tim.** 2005. "What Is Web 2.0." *O'Reilly*, September 30. <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>.
- Pantaleoni, Ana.** 2013. "Ferran Adrià Codificará la Cocina en la Bullipedia." *El País*, 18 Marzo. http://ccaa.elpais.com/ccaa/2013/03/18/catalunya/1363637632_675586.html.

- Parnin, Chris, Christoph Treude, and Lars Grammel.** 2012. "Crowd Documentation: Exploring the Coverage and the Dynamics of API Discussions on Stack Overflow." Georgia Institute of Technology. *Citeseerx*. <http://citeseerx.ist.psu.edu/viewdoc/similar;jsessionid=AB198D100393A78A4639D4099DCBA6B6?doi=10.1.1.371.6263&type=cc>.
- Ponzanelli, Luca, Andrea Mocci, Alberto Bacchelli, and Michele Lanza.** 2014. "Understanding and Classifying the Quality of Technical Forum Questions." In *2014 14th International Conference on Quality Software (QSIC)*, 343–352. IEEE. DOI: <https://doi.org/10.1109/QSIC.2014.27>
- Quinn, Alexander J., and Benjamin B. Bederson.** 2011. "Human Computation: A Survey and Taxonomy of a Growing Field." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1403–1412. CHI '11. New York: ACM. DOI: <https://doi.org/10.1145/1978942.1979148>
- Rodrigo, Álvaro, Anselmo Peñas, and Felisa Verdejo.** 2011. "Evaluating Question Answering Validation as a Classification Problem." *Language Resources and Evaluation* 46(3): 493–501. March 19 DOI: <https://doi.org/10.1007/s10579-011-9143-2>
- Shah, Chirag, and Jefferey Pomerantz.** 2010. "Evaluating and Predicting Answer Quality in Community QA." In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 411–418. SIGIR '10. New York, NY, USA: ACM. DOI: <https://doi.org/10.1145/1835449.1835518>
- Singh, Priyanka, and Nigel Shadbolt.** 2013. "Linked Data in Crowdsourcing Purposive Social Network." In *Proceedings of the 22nd International Conference on World Wide Web*, 913–918. WWW '13 Companion. New York, NY, USA: ACM. DOI: <https://doi.org/10.1145/2487788.2488079>
- Solorio, Thamar, Manuel Pérez-Coutiño, Manuel Montes-y-Gómez, Luis Villaseñor-Pineda, and Aurelio López-López.** 2004. "A Language Independent Method for Question Classification." In *Proceedings of 20th International Conference on Computational Linguistics*. Geneva, Switzerland, August 23–27. Stroudsburg, PA, USA: Association for Computational Linguistics.


- Spolsky, Joel.** 2008. "Stackoverflow.com." *Joel on Software*. Accessed December 15, 2016. <http://www.joelonsoftware.com/items/2008/04/16.html>.
- Stack Overflow.** 2016. "About Stack Overflow." Accessed July 18, 2016. <http://stackoverflow.com/company/about>.
- Surowiecki, James.** 2005. *The Wisdom of Crowds*. New York: Anchor.
- Tamura, Akihiro, Hiroya Takamura, and Manabu Okumura.** 2005. "Classification of Multiple-sentence Questions." In *Natural Language Processing – IJCNLP 2005*, edited by Robert Dale, Kam-Fai Wong, Jian Su, and Oi Yee Kwong, 426–437. Lecture Notes in Computer Science 3651. Springer Berlin Heidelberg, 2005. http://link.springer.com/chapter/10.1007/11562214_38.
- Tian, Yuan, Pavneet Singh Kochhar, Ee-Peng Lim, Feida Zhu, and David Lo.** 2013. "Predicting Best Answerers for New Questions: An Approach Leveraging Topic Modeling and Collaborative Voting." In *Social Informatics*, edited by Akiyo Nadamoto, Adam Jatowt, Adam Wierzbicki, and Jochen L. Leidner, 55–68. Lecture Notes in Computer Science 8359. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/978-3-642-55285-4_5.
- Treude, Christoph, Ohad Barzilay, and Margaret-Anne Storey.** 2011. "How Do Programmers Ask and Answer Questions on the Web?: NIER Track." In *2011 33rd International Conference on Software Engineering (ICSE)*, 804–807. ACM. DOI: <https://doi.org/10.1145/1985793.1985907>
- Tsai, Wenpin.** 2000. "Social Capital, Strategic Relatedness and the Formation of Intraorganizational Linkages." *Strategic Management Journal* 21(9): 925–939. September 1. DOI: [https://doi.org/10.1002/1097-0266\(200009\)21:9<925::AID-SMJ129>3.0.CO;2-I](https://doi.org/10.1002/1097-0266(200009)21:9<925::AID-SMJ129>3.0.CO;2-I)
- Vasilescu, Bogdan, Vladimir Filkov, and Alexander Serebrenik.** 2013. "StackOverflow and GitHub: Associations Between Software Development and Crowdsourced Knowledge." In *2013 International Conference on Social Computing (SocialCom)*, 188–195. IEEE. DOI: <https://doi.org/10.1109/SocialCom.2013.35>
- Voorhees, Ellen M, and Dawn M. Tice.** 1999. "The TREC-8 Question Answering Track Evaluation." In *Text Retrieval Conference TREC-8*, 83–105. http://trec.nist.gov/pubs/trec8/t8_proceedings.html.

- Wang, Xin-Jing, Xudong Tu, Dan Feng, and Lei Zhang.** 2009. "Ranking Community Answers by Modeling Question-answer Relationships via Analogical Reasoning." In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 179–186. SIGIR '09. New York, NY, USA: ACM. DOI: <https://doi.org/10.1145/1571941.1571974>
- Wilf, Herbert S.** 1982. "What is an Answer?" *The American Mathematical Monthly* 89(5): 289–292. May 1. DOI: <https://doi.org/10.2307/2321713>
- Williams, Greg.** 2012. "Staying Creative." *Wired UK Edition*, October.
- Yamakata, Yoko, Shinji Imahori, Yuichi Sugiyama, Shinsuke Mori, and Katsumi Tanaka.** 2013. "Feature Extraction and Summarization of Recipes Using Flow Graph." In *Social Informatics*, edited by Adam Jatowt, Ee-Peng Lim, Ying Ding, Asako Miura, Taro Tezuka, Gaël Dias, Katsumi Tanaka, Andrew Flanagin, and Bing Tian Dai, 241–254. Lecture Notes in Computer Science 8238. Springer International Publishing. http://link.springer.com/chapter/10.1007/978-3-319-03260-3_21.
- Zhang, Dell and Wee Sun Lee.** 2003. "Question Classification Using Support Vector Machines." In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 26–32. SIGIR '03. New York: ACM. DOI: <https://doi.org/10.1145/860435.860443>

How to cite this article: Jimenez-Mavillard, Antonio and Juan-Luis Suárez. 2018. Community Engagement and Quality Knowledge with Stackoverflow's Reputation System: Applications to Bullipedia. *Digital Studies/Le champ numérique* 8(1): 2, pp. 1–30, DOI: <https://doi.org/10.16995/dscn.293>

Submitted: 05 November 2017 **Accepted:** 06 November 2017 **Published:** 23 January 2018

Copyright: © 2018 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

 *Digital Studies/Le champ numérique* is a peer-reviewed open access journal published by Open Library of Humanities. **OPEN ACCESS** 