Computing (2015) 97:509–537
DOI 10.1007/s00607-014-0435-4

Web article quality ranking based on web community knowledge

Jingyu Han · Kejia Chen · Jianing Wang

Received: 17 June 2013 / Accepted: 17 November 2014 / Published online: 26 November 2014
© Springer-Verlag Wien 2014

Abstract The Web article has been recognized as the most popular data source for its convenience and abundance of information. Yet its data quality is compromised as most of existing quality assessment approaches rely mainly on the syntax or lexicon, rather than the semantics. We propose a Fact-based Quality Assessment (FQA) approach, which captures the data quality based on content semantics by gleaning the Web community knowledge. The FQA can automatically rank the Web data quality in terms of the three most important quality dimensions *accuracy*, *completeness* and *freshness*. Furthermore, the semantic dimensions can well complement existing works based on syntactical or lexical features. Given one source article, the FQA starts with the identification of an alternative context by collecting articles of the same topics. Then, the dimension baselines of *accuracy*, *completeness* and *freshness* are extracted in the alternative context. Finally, the data quality is determined by comparing the semantic corpus of the source article with the established dimension baselines. The performance of our FQA is verified in the experiments.

This research is fully supported by National Natural Science Foundation of China under the grant numbers 61003040, 61100135.

J. Han (✉) · K. Chen
Nanjing University of Posts and Telecommunications,
P.O.X 139, Nanjing 210003, People's Republic of China
e-mail: hanjingyusky@yahoo.com

K. Chen
e-mail: kejia.chen@gmail.com

J. Wang
Department of Computer Science and Information Systems, Birkbeck College, London, UK
e-mail: jianing@dcs.bbk.ac.uk

Keywords Data quality · Web article · Accuracy · Completeness · Freshness · Semantic corpus · Fact

Mathematics Subject Classification 68U35 · 68T30

1 Introduction

With the rapid expansion of the Web, people are increasingly relying on the Web to gain knowledge because of its convenience and abundance of information. Web data quality, namely, how good the Web data is, receives more and more attention due to the following reasons. First, the Web has gained great popularity and become the most convenient information source. Second, in contrast to the relational database where the schema can prevent incorrect data, it is difficult to ensure Web data quality for lack of schema. Third, in Web 2.0 age, users can freely publish, edit and modify user-generated content, which leads to a variety of data quality levels. For instance, Wikipedia's articles exhibit a wide range of quality levels.

In this work, we focus on the Web article's data quality. Generally speaking, data quality is widely accepted as a multi-dimensional concept including accuracy, completeness, freshness, consistency, etc. Given a Web article, a specific subset of data quality dimensions should be taken into consideration. For instance, for a Wikipedia article, users pay attention to its *accuracy* and *completeness*. In contrast, for a news article, users may also concern whether the article can provide *fresh* information besides its *accuracy* and *completeness*.

Much work has been done on assessing Web articles' data quality [1–3]. Yet existing work mainly focuses on assessing the quality in terms of syntax or lexicon, rather than semantics. So, we propose a novel Fact-based Quality Assessment (FQA) approach to rate Web articles' data quality. We focus on three first-class quality dimensions, namely, accuracy, completeness and freshness [4–6]. Note that our FQA applies to factual articles, rather than opinion articles.

We base our work on the following observations. First, a Web article can be regarded as a collection of *facts*. Each fact is a tri-tuple (h, v, t), where h is a head element, v is a relationship predicate, and t is a tail element. It denotes how h and t are correlated via v. These facts constitute an article's semantic corpus. For instance, given one Web page of Charles Darwin,[1] its semantic corpus consists of 34 facts, which are illustrated in Table 1.

Second, one topic is usually described by many alternative articles on the Web and the alternative articles are semantically similar or complementary to each other. Note the alternative article is not copied from another. Given two articles, if two articles contain the same or similar content, they verify each other. If each article only discusses a certain fraction of one topic, the two articles complement each other. For instance, another page of Charles Darwin[2] also describe the same topic, and its 29 facts are illustrated in Table 2. We observe that the facts 1 and 3 in Table 2 verify the facts 3

[1] http://www.bbc.co.uk/history/historic_figures/darwin_charles.shtml.

[2] http://www.lucidcafe.com/library/96feb/darwin.html.

Table 1 Facts of the first example web article

No.	Fact
1	(Darwin, was, a British scientist)
2	(Darwin,laid,foundations of the theory of evolution)
3	(Charles Robert Darwin, was born on, 12 February 1809)
...	...
34	(Darwin, was buried in, Westminster abbey)

Table 2 Facts of the second example web article

No.	Fact
1	(Charles Robert Darwin, was born on, February 12 1809)
2	(Darwin, was, the fifth child)
3	(Darwin, was, the British naturalist)
...	...
29	(His daughter, was present at, his deathbed)

and 1 in Table 1, respectively. In contrast, the facts 2 and 29 in Table 2 do not exist in Table 1. Hence, Table 2 also complements Table 1.

Third, whether a factual statement is true or not can be automatically identified by comparing *all* the alternative articles.

In our quality assessment framework, the three quality dimensions, namely, *accuracy*, *completeness* and *freshness* are rated through the following three phases.

1. *Building alternative context*: Given a source Web article, its alternative context is constructed with two steps. First, relevant articles are retrieved using the source article's title words plus a set of extracted keywords , and the articles constitute a relevant space. Second, each article in the relevant space is compared with the source article in terms of both topics and lexicons, and the similar relevant articles with respect to the source article constitute the source article's alternative context. In particular, to measure topic similarities, the Latent Dirichlet Allocation (LDA) analysis is exploited [7]. To measure lexical similarities, the word n-gram model is used.
2. *Extracting dimension baselines*. From semantics point of view, each alternative article's semantic corpus is a collection of facts, and each fact is represented as a tri-tuple (h, v, t). Among all the semantic corpora, the accuracy baseline with respect to the source article is extracted by voting, the completeness baseline is synthesized by ranking graph vertices, and the freshness baseline is constructed by collecting the freshest information of every fact.
3. *Calculating quality dimensions*. Three quality dimensions are determined by comparing the source article's semantic corpus with the three dimension baselines.

The contributions of the paper are as follows. Firstly, we propose to gauge a Web article's data quality in its alternative context, which is a collective knowledge base consisting of a set of alternative articles. Secondly, we propose concrete approaches to extract the accuracy baseline, completeness baseline and freshness baseline based on the facts the alternative context contains. Thirdly, the extracted semantic dimen-

sions *accuracy* and *completeness* are fairly independent of syntactical features. These semantic dimensions can help improve existing quality assessment approaches which are based on lexical or syntactical features. Experiments and analysis are given, showing that our FQA is a promising approach for the data quality assessment of Web articles.

2 Related work

Data quality is an important issue to all the content repositories, and its evaluation approaches are divided into two categories. The first category focuses on *qualitatively* analyzing data quality dimensions [4–6,8]. The second category deals with how to *quantitatively* assess data quality. The most obvious quality assurance approach is grammar check. Foltz et al. [9] point out that cohesion is an important measurement of writing quality and propose to use Latent Semantic Analysis (LSA) to measure cohesion. The result shows that LSA could be used to achieve the human accuracy in holistic judgement of quality. But its limitation is that the domain must be well defined and a representative corpus of the target domain must be available.

The work closely relevant to ours is assessing Web data quality. Dalip et al. [1] explore a significant number of quality indicators to assess Wikipedia articles' quality. Recently, Dalip et al. [10] further propose to group quality indicators into the views of quality, which are combined by meta-learning to rate the data quality of Wikipedia articles. The work in [11] employs a Learning-to-Ranking (L2R) approach for ranking answers in the Q&A forum, which takes advantage of eight groups of features including user features, user graph features, review features, structure features, length features, style features, readability features and relevance features. Rassbach et al. [12] present a maximum entropy model to identify Wikipedia articles' quality. Stvilia et al. [3] discuss seven IQ metrics which can be evaluated automatically on the Wikipedia content. The work in [2,13] gives how to use revision history to assess the trustworthiness of articles. The work of [14] proposes a measure for estimating the lexical quality of Web articles by detecting their spelling errors. The work in [15] proposes to measure the data quality of Web articles by extracting their factual information. These methods mainly focus on analyzing different types of quality indicators. But they do not touch on how to identify quality levels from the semantic point of view. Another work relevant to ours is on extracting facts or relations from Web articles [16,17]. But they do not touch on how to extract semantic corpus of data quality dimensions.

3 Building alternative context

Given a source article P_s, its alternative context is defined as follows.

Definition 1 (*alternative context*) A source article P_s's alternative context is a collection of alternative articles $\{P_1, \ldots, P_i, \ldots, P_n\}$ (including P_s itself), each with a similarity $sim(P_s, P_i)$ satisfying $\vartheta < sim(P_s, P_i) \leq 1 (0 < \vartheta < 1)$. Here ϑ is a threshold set by sampling.

3.1 Collecting relevant articles

The relevant articles are collected by searching the Web using keywords extracted from the source article. Then, the duplicate articles are removed using the Locality-Sensitive Hashing [18].

If the source article has a title, stop words of the title are removed and the remaining words are kept, denoted by K^{tit}. We further extract a set of keywords from the text with the *KeyGraph* approach [19], denoted by K^{ext}. The terms K^{ext} have been sorted according to their importance. Once the K^{tit} and K^{ext} are determined, the set of final keywords K^{final} are determined as follows. Suppose we need to find K^n keywords, where the K^n is empirically determined. If $|K^{tit} \cap K^{ext}| \leq K^n$, all the terms in $K^{tit} \cap K^{ext}$ are included in K^{final}. Furthermore, the top $K^n - |K^{tit} \cap K^{ext}|$ terms in $K^{ext} - (K^{tit} \cap K^{ext})$ are also included in K^{final}. If $|K^{tit} \cap K^{ext}| > K^n$, only the top K^n terms in K^{ext} are included in K^{final}. The keywords K^{final} are used to search the Web, and the returned top $\mathcal{N}$ articles are regarded as relevant ones.

We extract the plain text plus Web page's publishing time from all the relevant articles [20,21], and remove a standard list of stop words such as 'a' , 'an', 'the', 'of' etc. Then, we employ the Locality-Sensitive Hashing (*LSH*) to remove the duplicates[18]. We adopt a detecting-and-filtering policy to remove the duplicates. (1) *Detecting phase* after removing the punctuation marks and stop words in the text, we map the text into 3-gram space as follows. Given one article, we calculate its hash value by concatenating the values of 10 hashing functions, each of which is from the same LSH function family. Then, we map the article onto a bucket based on the hash value. The mapping is repeated l times by choosing different combinations of hash functions. (2) *Filtering phase* for the conflicted articles in the same bucket, the similarity of two articles is defined as the cosine of the frequency vectors of their words. If their similarity is 1, the duplicate is removed.

3.2 Deriving LDA model in relevant space

If two articles describe the same or similar topics, they should share some common lexical items and they cannot syntactically differ much from each other. Hence, we combine both the topic and lexical similarities to determine the alternative articles. To describe articles' topics, we employ the widely used Latent LDA analysis[7] to model a set of articles' topic distribution. The LDA model assumes that each article P_d is produced by a generative process.

Suppose that there are K topics. Variables $\alpha = (\alpha_1, \ldots, \alpha_K)$ and $\beta = (\beta_1, \ldots, \beta_K)$ are corpus-level parameters. Given a source article, we regard all the relevant articles (including the source article) as the corpus of LDA model, and use variational EM algorithm to determine corpus level parameters α and β [7]. Once α and β are determined, we employ a variational inference procedure as Algorithm 1 to determine the topic distribution with respect to the article P in the relevant space. As discussed by literature [22], combination of multiple models typically outperforms a single model and avoids the difficult task of setting an optimal K. In our task, the following numbers of topics are used: $K = 12, 24, 48$ and 96.

Algorithm 1: articleTopics

Input: Dirichlet distribution parameter $\alpha = (\alpha_1, \ldots, \alpha_k, \ldots, \alpha_K)$ where α_k is a scalar corresponding to topic k, word distributions of K topics $\beta = (\beta_1, \ldots, \beta_k, \ldots, \beta_K)$ where β_k is a vector, an article P of length N_P

Output: topic distribution of article P, denoted as Γ_P

foreach *topic* $k \in 1,\ldots,K$ **do**
  initialize $Y_k = \frac{N_P}{K}$;
end
repeat
  foreach *word* s_i $(1 \leq i \leq N_P)$ **do**
    foreach *topic* $k \in 1\ldots,K$ **do**
      $\phi_{ik} \leftarrow \beta_{ks_i} * exp(\Psi(Y_k))$;
    end
    normalize ϕ_i ;
  end
  foreach *topic* $k \in 1, \ldots, K$ **do**
    $Y_k \leftarrow \alpha_k + \sum_{i=1}^{N_P} \phi_{ik}$;
  end
until *convergence*;
normalize $(Y_1, \ldots, Y_K)$; **return** $\Gamma_P \leftarrow (Y_1, \ldots, Y_K)$;

3.3 Determining alternative articles

Alternative articles of a source article are determined based on their topic and lexical similarities. The similarity of two articles P_s and P_r is defined as

$$sim(P_s, P_r) = \eta \times sim_t(P_s, P_r) + (1 - \eta) \times sim_{lex}(P_s, P_r), \tag{1}$$

where $0 < \eta < 1$, $sim_t(P_s, P_r)$ and $sim_{lex}(P_s, P_r)$ are topic similarity and lexical similarity, respectively.

One article's topic similarity with respect to the source article is calculated with Algorithm 2. Note the $cos(t_s, t_r)$ is the *cosine* of two vectors t_s and t_r, where the $t_s(t_r)$ is the latent topics of article $P_s(P_r)$ and its component is the weight of corresponding latent topic.

Algorithm 2: computeTopicSimilarity

Input: source document P_s, relevant document P_r, a series of LDA models $\{(\alpha^K, \beta^K, K)\}$ $(K = 12, 24, 48, 96)$

Output: topic similarity $sim_{topi}(P_s, P_r)$

$ret \leftarrow \emptyset$;
foreach $K \in \{12, 24, 48, 96\}$ **do**
  $t_s \leftarrow$ articleTopics(α^K, β^K, P_s); $t_r \leftarrow$ articleTopics(α^K, β^K, P_r);
  $ret \leftarrow ret \cup \cos(t_s, t_r)$;
end
$sim_{topi} \leftarrow$ the largest $cosine$ value in ret; **return** sim_{topi} ;

Table 3 Text and words of the source article and other relevant articles

P_s	Text	Charles Darwin was the British scientist
	Words	{Charles, Darwin, British, scientist}
P_1	Text	Darwin was the British naturalist
	Words	{Darwin, British, naturalist}
P_2	Text	Charles Darwin was the British naturalist
	Words	{Charles, Darwin, British, naturalist}
P_3	Text	Darwin is the capital of the Northern Territory of Australia
	Words	{Darwin, capital, Northern, Territory, Australia}

Table 4 Latent topic vectors and topic similarities of Example 1

	$K = 3$		$K = 5$		$sim_t(P_s, P_r)$
	Topic vector	$cos(t_s, t_r)$	Topic vector	$cos(t_s, t_r)$	
t_s	(0.6, 0.1, 0.3)	–	(0.3, 0.2, 0.2, 0.1, 0.2)	–	–
t_1	(0.7, 0.15, 0.15)	0.968	(0.35, 0.15, 0.2, 0.15, 0.15)	0.978	0.978
t_2	(0.6, 0.15, 0.5)	0.95	(0.3, 0.2, 0.15, 0.15, 0.2)	0.95	0.95
t_3	(0.1, 0.6, 0.3)	0.456	(0.1, 0.3, 0.5, 0.05, 0.05)	0.733	0.733

Given two articles P_s and P_r, their lexical similarity is calculated in the n-gram space. Suppose that $lex(P_s) = (f_1^s, \ldots, f_i^s, \ldots, f_n^s)$, where f_i^s is the frequency of the n-gram i. Then, the lexical similarity between P_s and P_r is defined as

$$sim_{lex}(P_s, P_r) = \cos(lex(P_s), lex(P_r)). \tag{2}$$

Example 1 Suppose that the text of the source article P_s and other relevant articles are shown in Table 3. In other words, there are four articles in the relevant space of P_s, namely, $\{P_s, P_1, P_2, P_3\}$. After removing the stop words 'the', 'of', 'was' and 'is', the remaining words are also reported in the table. Suppose we set K=3, 5. The latent topic vector of each article, which is obtained with Algorithm 1, is illustrated in Table 4. The topic similarity between P_s and $P_r (r = 1, 2, 3)$ are also show in Table 4.

We then calculate the 3-gram frequency vector of every article. We have $\cos(lex(P_s), lex(P_1)) = 0.83$, $\cos(lex(P_s), lex(P_2)) = 1$, $\cos(lex(P_s), lex(P_3)) = 0.69$. If we set $\eta = 0.6$, $\vartheta = 0.75$, we have the following results. $sim(P_s, P_1) = 0.978 \times 0.6 + 0.83 \times 0.4 = 0.9188$, $sim(P_s, P_2) = 1 \times 0.6 + 1 \times 0.4 = 1$, and $sim(P_s, P_3) = 0.733 \times 0.6 + 0.69 \times 0.4 = 0.7158$. So, P_1, P_2 and P_s itself constitute the alternative context of P_s.

4 Extracting dimension baselines

First, we extract the semantic corpus of every alternative article.

Definition 2 (*article's semantic corpus*) Given an article P, its semantic corpus, denoted as $corp(P)$, is the set of facts it contains.

Second, dimension baselines are constructed from all the semantic corpora in the alternative context.

To avoid the ambiguity caused by pronouns when extracting the semantic corpora, we pre-process every alternative article with the Stanford Deterministic Coreference Resolution System[3] [23] to solve the co-reference problem. The task is to finding all the expressions refer to the same entity. For instance, suppose there are two sentences 'Charles Darwin was an English naturalist and geologist. He established the theory of evolution.'. After the pre-processing, the word 'he' is replaced by 'Charles Darwin'.

4.1 Extracting semantic corpus

From the semantics point of view, an article represents a set of facts. One fact is a tri-tuple (h, v, t). Here the head element h is a noun or a noun phrase. The tail element t is a noun (phrase), an adjective (phrase) or an adverb (phrase). The relationship predicate v connects the head element and the tail element, which is a verb or a verb phrase.

We use the ReVerb[4] and Part-Of-Speech (POS) tagger to extract all the facts from alternative article's text. During extracting facts, one thesaurus T is built to index synonyms and direct hyponyms for later processing. To determine the synonyms or direct hyponyms, WordNet[5] is consulted. Given two words or phrases, whether they are semantically equivalent is determined by the following rules.

Rule 1: If they are literally the same, they are semantically equivalent. Otherwise, go to *Rule 2*.

Rule 2: By consulting the local WordNet, we determine whether two words or phrases are synonymy or direct hyponym. We regard both the synonymy and direct hyponym as semantic equivalence although the latter represents a *weak* semantic equivalence.

Three hash tables, namely, Head Hash (HH), Verb Hash (VH) and Tail Hash (TH), are also built to index the facts based on the head element, predicate and tail element, respectively.

4.2 Extracting quality dimension baselines

To extract quality dimension baselines from semantic corpora, we only take into account the facts that occur in more than one article. We define *support* of a fact f as

$$sup(f) = |equ(f)|, \tag{3}$$

[3] http://nlp.stanford.edu/software/dcoref.shtml.

[4] http://reverb.cs.washington.edu/.

[5] http://wordnet.princeton.edu/.

Algorithm 3: filterFacts

Input: thesaurus T, head hash table HH, article's semantic corpus $corp(P)$, support threshold χ
Output: refined semantic corpus

```
foreach f_s ∈ corp(P) do
    H ← retrieve synonyms or hypononyms of f_s.h from T; S ← ∅;
    foreach e ∈ H do
        S_temp ← retrieve facts containing e from HH; S ← S ∪ S_temp;
    end
    foreach f_x ∈ S do
        if sim_fact(f_s, f_x) < 1 then
            remove f_x from S;
        end
    end
    if |S| < χ then
        remove f_s from corp(P);
    end
end
```

where $|equ(f)|$ is the number of articles that contain the semantic equivalents of f. Only the fact with *support* above a given threshold χ counts. Given an article, the non-qualified facts are filtered out with Algorithm 3, in which the fact similarity is defined as follows.

Definition 3 (*fact similarity*) Given two facts $\overline{f}(\overline{h}, \overline{v}, \overline{t})$, $\underline{f}(\underline{h}, \underline{v}, \underline{t})$, their similarity is

$$sim_{fact}(\overline{f}, \underline{f}) = \omega_h * es(\overline{h}, \underline{h}) + \omega_p * ps(\overline{v}, \underline{v}) + \omega_t * es(\overline{t}, \underline{t}) \tag{4}$$

where es is the element similarity, ps is the predicate similarity, and $\omega_h + \omega_p + \omega_t = 1$ holds for the head element weight, predicate weight and tail element weight. Here the weights ω_h, ω_p, and ω_t are set by sampling.

Definition 4 (*element similarity* (*es*)) Given two head (or tail) elements he_1 and he_2, their element similarity is calculated according to the following rules.

Rule 1: If he_1 and he_2 are literally the same, we have $es(he_1, he_2) = 1$. To identify whether two facts are literally the same, we need to perform stopping and stemming on each component of one fact. If he_1 is different from he_2, go to Rule 2.

Rule 2: If $he^1 = \langle w_1^1, \ldots, w_n^1 \rangle$ and $he^2 = \langle w_1^2, \ldots, w_n^2 \rangle$ exhibit the same sequence of part of speech, their element similarity is defined as

$$es(he^1, he^2) = \frac{1}{n} \sum_{i=1}^{n} sim^{se}(w_i^1, w_i^2), \tag{5}$$

where $sim^{se}(w_i^1, w_i^2)$ is the semantic similarity of two words, which is defined as

$$sim^{se}(w_1, w_2) = \begin{cases} 1 & \text{if } w_1 \text{ and } w_2 \text{ are semantically equivalent} \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

If $he^1 = \langle w_1^1, \ldots, w_n^1 \rangle$ and $he^2 = \langle w_1^2, \ldots, w_n^2 \rangle$ do not exhibit the same sequence of part of speech, go to Rule 3.

Rule 3: If he_1 is a single word w, and he_2 is a noun phrase, adjective phrase or adverb phrase, we calculate $es(he_1, he_2)$ based on the semantic similarity between w and the core constituents of phrase he_2.

Rule 4: If both he_1 and he_2 are phrases, we calculate $es(he_1, he_2)$ based on phrase's type and phrase's core constituents.

In particular, if the head or tail element contains numbers, dates (time), they should be pre-processed as follows. First, every occurrence of numbers is identified by hand-tuned rules. Then, the components of dates (time) are extracted by two steps, i.e., the expression identification and component segmentation.

In the expression identification step, the sequence of tokens (words, spaces and marks) containing dates (time) is identified by a trained naive Bayes classifier according to the *left context*, *right context* as well as *format and style*. The left context characterizes the left context of the expressions containing dates (time). The right context characterizes the right context of the expressions containing dates (time). The format and style characterizes the format and style of the expressions containing dates (time). We do not address it in detail due to space.

In the component segmentation step, the identified expressions are first segmented into different components by the delimiters such as '/', ':' and spaces. Then, every component is classified into one of the six categories, i.e., year, month, day, hour, minute and second. This is also achieved by training a naive Bayes classifier according to the left context, right context as well as the format and style of the components.

Once the dates (time) or numbers are identified and segmented, the two elements are compared according to the following rules.

Rule 1: In this case, two elements contain the same type of data (date, time or number) and no other words. The similarity of dates (time) or numbers, denoted by es^{dtn}, is calculated as follows. If the two dates (time) or numbers are different, the similarity of the two elements is 0. If they are the same, the similarity is 1.

Rule 2: In this case, both elements contain the same type of data (date,time or number) and some other words. We first calculate the similarity of dates (time) or numbers, also denoted by es^{dtn}. Then, we calculate the similarity of reduced elements (excluding the dates, time or numbers) according to the Definition 4, denoted by es^{com}. Finally, the element similarity is

$$\omega^{dtn} \times es^{dtn} + \omega^{com} \times es^{com} \tag{7}$$

where the weights ω^{dtn} and ω^{com} satisfy $\omega^{dtn} + \omega^{com} = 1$. Here $\omega^{dtn} = \frac{len^d}{len^d + len^c}$ and $\omega^{com} = \frac{len^c}{len^d + len^c}$ hold, where len^d is the minimum of the numbers of components in the two date (time or number) expressions and len^c is the the minimum of the numbers of words in the two remaining expressions.

Definition 5 (*predicate similarity* (*ps*)) Given two relationship predicates rp_1 and rp_2, $ps(rp_1, rp_2)$ is calculated according to the following rules.

Rule 1: If rp_1 and rp_2 are literally the same, we have $ps(rp_1, rp_2) = 1$. Otherwise, go to Rule 2.

Rule 2: If rp_1 and rp_2 each are single verbs, their predicate similarity is calculated according to Eq. 6. Otherwise, go to Rule 3.

Rule 3: Suppose that one predicate, say rp_1, is a single verb v, and the other, namely rp_2, is a verb phrase. If rp_1 is an equivalent of rp_2 by consulting the thesaurus, $ps(rp_1, rp_2) = 1$. Otherwise, we assume that each relationship predicate is categorized into four patterns including *V*, *VP*, *VW*P* and multiple adjacent verb phrases [24]. We extract all the verbs in rp_2. Furthermore, if rp_2 contains light verb construction (LVC) [24], we also extract all the nouns in the LVC. The verbs and nouns are stemmed and merged into one set, denoted as $VN = \{vn_1, \ldots vn_m\}$. We have

$$ps(rp_1, rp_2) = max_{1 \le i \le m}\{sim^{se}(v, vn_i)\}. \tag{8}$$

Rule 4: Suppose that both rp_1 and rp_2 are verb phrases. If rp_1 is an equivalent of rp_2 by consulting the thesaurus, $ps(rp_1, rp_2) = 1$. Otherwise, we assume each predicate contains at most two contiguous verb phrases, and their similarity falls into the following cases. First, if rp_1 is one verb phrase vp^1, and rp_2 is one verb phrase vp^2, we have $ps(rp_1, rp_2) = sim^{vp}(vp^1, vp^2)$. Here $sim^{vp}(vp^1, vp^2)$ is semantic similarity between two verb phrases. Second, if rp_1 is one verb phrase vp^1, and rp_2 contains two contiguous verb phrases vp_1^2 and vp_2^2, we have $ps(rp_1, rp_2) = max(sim^{vp}(vp^1, vp_1^2), sim^{vp}(vp^1, vp_2^2))$. Third, if rp_1 contains two contiguous verb phrases vp_1^1 and vp_2^1, and rp_2 contains two contiguous verb phrases vp_1^2 and vp_2^2, $ps(rp_1, rp_2)$ is defined as

$$max\left\{\frac{sim^{vp}(vp_1^1, vp_1^2) + sim^{vp}(vp_2^1, vp_2^2)}{2}, sim^{vp}(vp_2^1, vp_2^2)\right\}. \tag{9}$$

Example 2 Let $\omega_h = 0.3, \omega_p = 0.4$ and $\omega_t = 0.3$. Suppose there are two facts (Darwin, was born on, 1854) and (Darwin, was born on, 12 february 1809). By applying the trained naive Bayes classifiers, we recognize the first tail element '1854' represents a year. So we tagged the first tail element as $(year\| - \| - \| - \| - \|-)$. Likewise, we tagged the second tail element as $(day\|month\|year\| - \| - \|-)$. Without doubt, es('Darwin','Darwin')=1 and ps('was born on', 'was born on')=1 hold. Since the two tail elements represent different dates by analyzing their components, es('1854',' 12 February 1809')=0. According to the Definition 3, the similarity of the two facts is 0.7.

Again, suppose there are two facts (farmer, has, 7 hares) and (farmer, has, 8 rabbits). Then, es('farmer','farmer')=1 and ps('has', 'has')=1 hold. In the tail elements, since the two numbers are different, $sim^{se}(7, 8) = 0$ holds. Also, sim^{se}(hare, rabbit) = 1 holds. As every element contains one number and one other word, the length of number expression $len^d = 1$ holds and the length of remaining words $len^c = 1$ also

holds. So, es('7 hares','8 rabbits')= $\frac{1}{2} \times 0 + \frac{1}{2} \times 1 = \frac{1}{2}$. According to the Definition 3, the similarity of the two facts is 0.85.

We handle the negation [25] to more precisely calculate two facts' similarity. If there is one negation word in the head (tail) element or relationship predicate, we assume that it negates the whole fact. Given two facts $\overline{f}$ and $\underline{f}$, we handle the negation as follows.

(1) *Negation in head elements* if either head element begins with 'no', 'not', 'few' or 'little', we first extract the reduced facts by excluding the negative words, denoted by $\overline{f'}$ and $\underline{f'}$. Suppose that the fact similarity between $\overline{f'}$ and $\underline{f'}$ is x satisfying $0 < \mu \leq x \leq 1$, where μ is a negation threshold set by sampling. If only one head element contains the negative word, we have $sim_{fact}(\overline{f}, \underline{f}) = 1 - x$. If both head elements contain the negative words, $sim_{fact}(\overline{f}, \underline{f}) = x$ holds.

(2) *Negation in relationship predicates* if either relationship predicate contains adverbs 'not' or 'never', we extract the reduced facts by excluding the adverbs 'not' and 'never'. Suppose that the similarity of two reduced facts $\overline{f'}$ and $\underline{f'}$ is $x(0 < \mu \leq x \leq 1)$. If only one relationship predicate contains the negation adverbs, we have $sim_{fact}(\overline{f}, \underline{f}) = 1 - x$. If both relationship predicates contain the negation adverbs, $sim_{fact}(\overline{f}, \underline{f}) = x$ holds.

(3) *Negation in tail elements* if either tail element contains the adjectives 'no', 'few' or 'little', we first extract the reduced tail elements by excluding the negative words. Suppose the similarity of two reduced facts is $x(0 < \mu \leq x \leq 1)$. If only one tail element contains the negative words, we have $sim_{fact}(\overline{f}, \underline{f}) = 1 - x$. If both tail elements contain the negative words, the similarity of two facts is x.

Note only when the similarity of two reduced facts is higher than the negation threshold μ, we take into account the negation effect.

Example 3 Let us illustrate how to calculate the fact similarity when the negation effect is taken into account. Let negation threshold $\mu = 0.9$. Suppose two sentences are 'Few books talk about Charles Darwin' and 'The book talks about Charles Darwin'. The corresponding two facts are $\overline{f}$ = (Few books, talk about, Charles Darwin) and $\underline{f}$ = (book, talks about,Charles Darwin). The reduced facts by excluding the negative words are $\overline{f'}$ = (books, talk about, Charles Darwin) and $\underline{f'}$ = (book, talks about, Charles Darwin). As $sim_{fact}(\overline{f'}, \underline{f'}) = 1$, which is greater than μ, and only one head element contains the negative words, $sim_{fact}(\overline{f}, \underline{f}) = 1 - 1 = 0$ holds.

4.2.1 Constructing accuracy baseline

Given a source article, its accuracy baseline is extracted from the candidate facts.

Definition 6 (*candidate facts*) Given a source fact f_s and a size threshold Q, f_s's candidate baseline facts, denoted by $\Upsilon(f_s)$, are the top Q facts from $\{f_1, \ldots, f_i, \ldots, f_n\}$ based on their fact similarities with respect to f_s. Each similar fact f_i satisfies that at least two components of its are the equivalents of corresponding components of f_s.

In the above definition, f_s's candidate facts always include f_s itself.

Algorithm 4: query

Input: thesaurus T, hash tables T_1, T_2, source fact f_s, f_s's head(tail) elements or predicates c_1 and c_2
Output: similar facts with respect to f_s

```
F ← ∅; S1 ← retrieve c1' equivalents from T; S2 ← retrieve c2' equivalents from T;
foreach s1 ∈ S1 do
    F1 ← retrieve facts from T1 or T2 based on s1;
    foreach s2 ∈ S2 do
        F2 ← retrieve facts from T1 or T2 based on s2; F ← F ∪ (F1 ∩ F2);
    end
end
return F;
```

Algorithm 5: extractCandiFacts

Input: thesaurus T, fact hash tables HH, VH, TH, source fact f_s
Output: top Q candidate facts

```
ret ← ∅; (h, v, t) ← fact components of f_s; set1 ← query(T, HH, VH, f_s, h, v); set2 ←
query(T, HH, TH, f_s, h, t); set3 ← query(T, VH, TH, f_s, v, t); ret ← set1 ∪ set2 ∪ set3;
sort all facts in ret in descending order based on similarity with respect to f_s;
return top Q ones in ret;
```

Definition 7 (*accuracy baseline fact*) Given one fact f_s in the source article P_s, its accuracy baseline fact is the fact from $\Upsilon(f_s)$ with the largest confidence among the set of candidate facts of P_s.

Each accuracy baseline fact corresponds to one fact of the source article. It is extracted by two phases, i.e., collecting candidate facts and identifying targets.

In the phase of collecting candidate facts, all the candidate facts similar to the source fact are collected. Each time we retrieve candidate facts based on the source fact's two components, which is detailed in Algorithm 4. When all component pairs of one fact have been searched for, retrieved facts are sorted and the top Q ones are chosen. The whole procedure for collecting candidate facts of one source fact is described in Algorithm 5. The running time of Algorithm 4 is determined by the number of I/O operations for accessing the thesaurus and fact hash tables. Suppose that the size of T is M. We use the binary-search to find the word's equivalents in T. So, the time complexity of Algorithm 5 is $O(log_2 M)$.

In the phase of identifying targets, the accuracy baseline fact of a source fact is identified by voting in the set of candidate facts. Given a source fact f_s, we regard the candidate fact supported most in the alternative context as f_s's baseline fact. To this aim, we use the confidence to measure the extent to which a fact's components are confirmed by all the candidate facts. Given one fact $f(h, v, t)$, its confidence is a combination of the head confidence, predicate confidence and tail confidence. Formally, $conf_h(f) = \frac{h^{num}}{|\Upsilon|}$, $conf_v(f) = \frac{v^{num}}{|\Upsilon|}$, and $conf_t(f) = \frac{t^{num}}{|\Upsilon|}$ hold. Here $h^{num}(v^{num}, t^{num})$ is the number of literal appearances of $h(v, t)$ in the head elements (predicates, tail elements) in the set of candidate facts Υ. Then, the confidence of f is

Algorithm 6: identifyTarget

Input: candidate facts Υ, source fact f_s, confidence threshold δ
Output: most accurate fact

```
if δ > conf(f)(∀f ∈ Υ \ f_s) then
    return null;
end
else
    conflist ← sorts all facts ∈ Υ based on their confidence; f_top ← top fact of conflist; return f_top;
end
```

Algorithm 7: constructAccuracyBaseline

Input: source article corpus $corp(P)$, thesaurus T, fact hash tables HH, VH, TH, confidence threshold δ
Output: accuracy baseline of P

```
Π ← ∅; // Initialize the accuracy baseline
foreach f_s ∈ corp(P) do
    Υ ← extractCandiFacts(T, HH, VH, TH, f_s); f_acc ← identifyTarget(Υ, f_s, δ);
    Π ← Π ∪ f_acc;
end
return Π;
```

Table 5 Candidate facts and confidence

Fact no.	Facts	Confidence
f_s	(Charles Darwin, was, British scientist)	$\frac{1}{3}(\frac{3}{4}+1+\frac{1}{4}) = 0.67$
f_1	(Charles Darwin, was, British naturalist)	$\frac{1}{3}(\frac{3}{4}+1+\frac{2}{4}) = 0.75$
f_2	(Darwin, was, British naturalist)	$\frac{1}{3}(\frac{1}{4}+1+\frac{2}{4}) = 0.58$
f_3	(Charles Darwin, was, scientist)	$\frac{1}{3}(\frac{3}{4}+1+\frac{1}{4}) = 0.67$

$$conf(f) = \frac{conf_h(f) + conf_v(f) + conf_t(f)}{3}. \tag{10}$$

Based on the confidence, the most accurate representation of one source fact is obtained by Algorithm 6. The accuracy baseline of one source article is constructed by Algorithm 7.

Example 4 Suppose that the source fact f_s is (Darwin, was, British scientist). Its four candidate facts (including itself) and the confidence are shown in Table 5. From the table, we can see the confidence of f_1 is the largest. So. the accuracy baseline fact of f_s is f_1.

4.2.2 Constructing completeness baseline

The completeness baseline is represented as an undirected graph, each vertex of which corresponds to one distinct fact in the alternative context. The completeness baseline is constructed with the following two steps.

(1) *Constructing Initial Graph* each distinct and qualified fact in alternative context acts as one vertex with an initial completeness score $s(f_i, 0) = popu(f_i)$, where $popu(f_i)$ is the number of alternative articles where the fact f_i appears. Weight edges are added to connect every fact pair (f_i, f_j) whose similarity satisfies $0 < sim_{fact}(f_i, f_j) < 1$.
(2) *Refining Completeness Score* each vertex's completeness score is calculated by iteration. The vertex score is calculated with equation 11 until it reaches a fixed point.

$$s(f_i, t) = s(f_i, t-1) - \frac{1}{2^t} \sum_{j \in con(f_i)} \frac{s(f_j, t-1)}{|con(f_i)| + \sum_{f_k \in con(f_j)} \frac{1}{sim_{fact}(f_k, f_j)}} \quad (11)$$

Here $s(f_i, t)$ is the score of f_i at the t-th iteration, $sim_{fact}(f_k, f_j)$ is the fact similarity between f_k and f_j, and $con(f_i)$ is the vertices that are directly connected to vertex f_i. Given a convergence threshold $\rho (0 < \rho < 1)$, if the average variation of all vertices between two successive iterations is within ρ, the computation ends.

Theorem 1 *Each vertex's completeness score converges to a value between 0 and 1.*

Proof : Given a vertex f_i, its iteration process is a sequence of values $\langle s(f_i, 1), \ldots, s(f_i, n), s(f_i, n+1), \ldots \rangle$, and the sequence satisfies the following two conditions.

Boundness: $\forall t \geq 1, 1 \geq s(f_i, t) > 0$ holds. We use induction to prove that, $\forall t \geq 1, 1 \geq s(f_i, t) \geq \frac{1}{2^t}$ holds.

(1) When $t = 1, s(f_i, 1) = 1 - \frac{1}{2} \sum_{j \in con(f_i)} \frac{1}{|con(f_i)| + \sum_{f_k \in con(f_j)} \frac{1}{sim_{fact}(f_k, f_j)}} \geq \frac{1}{2}$.
(2) Suppose that when $t = n, 1 \geq s(f_i, n) \geq \frac{1}{2^n}$ holds. When $t = n + 1$,

$$\begin{aligned} s(f_i, n+1) &= s(f_i, n) - \frac{1}{2^{n+1}} \sum_{j \in con(f_i)} \frac{s(f_j, n)}{|con(f_i)| + \sum_{f_k \in con(f_j)} \frac{1}{sim_{fact}(f_k, f_j)}} \\ &\geq s(f_i, n) - \frac{1}{2^{n+1}} \sum_{j \in con(f_i)} \frac{1}{|con(f_i)| + \sum_{f_k \in con(f_j)} \frac{1}{sim_{fact}(f_k, f_j)}} \\ &\geq \frac{1}{2^{n+1}}. \end{aligned}$$

Monotonicity during the iteration, completeness score decreases monotonically. In other words, $\forall t \geq 1, s(f_i, t) \geq s(f_i, t+1)$ holds. This is because

$$s(f_i, t) - s(f_i, t+1) = \frac{1}{2^{n+1}} \sum_{j \in con(f_i)} \frac{s(f_j, t)}{|con(f_i)| + \sum_{f_k \in con(f_j)} \frac{1}{sim_{fact}(f_k, f_j)}} \geq 0.$$

Hence, we know $s(f_i, t)$ must converge to a value between 0 and 1.

4.2.3 Constructing freshness baseline

Given one source article, its freshness baseline consists of the freshness information of all the facts. We take into consideration two factors for one fact's freshness.

(1) *Occurrence time* when one fact first appears on the Web or it occurs in the real world. The earlier the occurrence time is, the less fresh the fact is.
(2) *Content uniqueness* if one article gives some facts that are rarely touched on by other articles, the article's freshness is high. In contrast, if the article's facts have been talked about by many other articles, its freshness is low.

Given one source article with m facts $\{f_1, \ldots, f_i, ..., f_m\}$, its freshness baseline is a set of m tuple, and each tuple corresponds to one source fact f_i. Each tuple takes the form $(t_{\mathcal{F}}, u_{cont})$, where $t_{\mathcal{F}}$ is the *occurrence time*, and u_{cont} is the fact's *content uniqueness* defined as

$$u_{cont}(f) = 1 - \frac{sup(f)}{\zeta}, \tag{12}$$

where ζ is the size of the alternative context. Given one source article, its freshness baseline is constructed with Algorithm 8.

Algorithm 8: constructFreshnessBaseline

Input: alternative context $\mathcal{C}$, source article P_s, thesaurus T, fact tables HH, VH, TH
Output: P_s's freshness baseline $\mathcal{F}$

$\mathcal{F} \leftarrow \emptyset$;
extract the publishing time of every article $P \in \mathcal{C}$ and assign it to every fact $f \in P$;
foreach *fact* $f_s \in corp(P)$ **do**
  $F_{list} \leftarrow$ extractCandiFacts(T, HH, VH, TH, f_s); $f_s.t_{\mathcal{F}} \leftarrow f_s$'s published time; $S_{sup} \leftarrow \emptyset$;
  foreach $f \in F_{list}$ **do**
    if $sim_{fact}(f, f_s) = 1$ **then**
      if $f.t < f_s.t_{\mathcal{F}}$ **then**
        $f_s.t_{\mathcal{F}} \leftarrow f.t$;
      end
      $S_{sup} \leftarrow S_{sup} \cup$ article id of f;
    end
  end
  $f_s.u_{cont} \leftarrow 1 - \frac{|S_{sup}|}{\zeta}$; $\mathcal{F} \leftarrow \mathcal{F} \cup (f_s.t_{\mathcal{F}}, f_s.u_{cont})$;
end
return $\mathcal{F}$;

5 Calculating quality dimensions

Accuracy gives to what extent the data is close to its truth.

Definition 8 (*accuracy*) Given a source article P_s with n facts, its accuracy is

$$acc(P_s) = \frac{1}{n} \sum_{j=1}^{n} sim_{fact}(f_j, f_b) \tag{13}$$

where f_b is f_j's corresponding baseline fact in the accuracy baseline. In particular, if a fact f_j's corresponding baseline fact is null, we assume that $sim_{fact}(f_j, null) = 0$.

Completeness gives to what extent related facts are described in a source article, which is the ratio of the amount of information in the source article to that in its completeness baseline. To take into consideration the semantic overlap when calculating the amount of information of the source article, we also perform the iterations on the fact graph of the source article P_s until the iteration stops. Formally, the completeness is defined as follows.

Definition 9 (*Completeness*) Given the source article P_s and its completeness baseline $\mathcal{B}$, both of which are represented as weighted graphs, completeness of P_s is defined as

$$comp(P_s) = \frac{\sum_{f_i \in P_s} s(f_i)}{\sum_{f_j \in \mathcal{B}} s(f_j)}, \tag{14}$$

where $s(f_i)$ and $s(f_j)$ are the final completeness scores of vertex f_i and f_j, respectively.

Freshness gives the extent to which data represents fresh information. Given one source article P_s with n facts and its freshness baseline $\mathcal{F}$, its freshness is defined as

$$fresh(P_s) = \frac{1}{n} \sum_{j=1}^{n} \frac{f_j.t - f_j.t_{\mathcal{F}}}{t_{cur} - f_j.t_{\mathcal{F}}} * f_j.u_{cont}, \tag{15}$$

where $f_j.t$ is the publication time of f_j, $f_j.t_{\mathcal{F}}$ is f_j's baseline time, t_{cur} is current time, and $f_j.u_{cont}$ is f_j's *content uniqueness*.

6 Experimental evaluation

Our experiments ran on a laptop with Intel Core i3 M370@2.4GHz and RAM 2048M. We collected three datasets. The first is a collection of Wikipedia articles describing scientist, denoted by *SCT*, which contains 200 source articles describing biologists, chemists, earth scientists, physicists, psychologists and economists.[6] The articles have been assigned quality class labels, namely, Featured Article (FA), Good Article (GA), B-Class (B), C-Class (C), Start-Class (ST) and Stub-Class (SU), according to the Wikipedia community quality grading scheme.[7] The quality class labels were extracted from the discussion pages. For every source article, we used 5 (i.e., the value of $|K^{final}|$) keywords to search the Web for its relevant articles. The top 60 articles returned by Google are regarded as the relevant ones. The second dataset is also a collection of Wikipedia articles, which span a variety of topics including history, art, geography, society, culture, technology, religion, people, mathematics and natural science, denoted by *MT*. We collected 15 articles for every topic, totalling 150 articles.

[6] http://en.wikipedia.org/wiki/Scientist.

[7] http://en.wikipedia.org/wiki/Wikipedia:Version_1.0_Editorial_Team/Assessment.

The third dataset consists of 100 source articles, which are the news of Syria civil war. They were manually chosen from www.yahoo.com and www.foxnews.com.

Every source article's relevant articles were pre-processed as follows. We first extracted the plain text and publishing time. Then, we removed the duplicate articles in the relevant space by setting $l = 6$, which were determined by sampling. Finally, we used the Stanford Deterministic Coreference Resolution System to replace every pronoun with its entity name, thus facilitating the disambiguation of two facts in the later processing.

When calculating the fact similarity, we set $\eta = 0.5$ by tuning. The context thresholds of two Wikipedia datasets were tuned with the following three steps. (1) We randomly chose 25 % articles for each quality class, and all the chosen articles constituted the sample set. (2) We decreased the context threshold ϑ from 0.9 to 0.3 by an interval 0.05, and plotted the curve of average performance measurements(defined later) with respect to the context threshold. (3) We chose the first point with the second derivative being 0 as the threshold, where the maximum value of measurements is generated. Finally, we set $\vartheta = 0.52$ for the SCT dataset and $\vartheta = 0.56$ for the MT dataset. Similarly, the value of ϑ for the news dataset is set as 0.59.

Before extracting dimension baselines, we first trained naive Bayes classifiers for identifying and segmenting the dates (time) or numbers in the head or tail elements by randomly choosing 25 % articles as training set. The negation threshold $\mu = 0.86$ is also set by randomly sampling 25 % articles. During extracting dimension baselines, we set support $\chi = 2$.

6.1 Performance of the FQA

6.1.1 Precision of constructed dimension baselines

On the two Wikipedia datasets, we evaluate the precision of constructed baselines with respect to the two dimensions *accuracy* and *completeness*. We developed a tool to help users manually identify the gold standard facts. To identify the gold facts of accuracy baseline, each source fact and its candidate facts were displayed. We asked 5 users to manually choose the most accurate representation of the source fact. The fact which is chosen by the largest number of users is the gold standard fact of the accuracy baseline. To identify the gold facts of completeness baseline, we displayed each source article and all the facts in its alternative context, and asked 5 users to choose the facts that should be covered. If one fact was chosen by at least 3 users, it was regarded as one fact in the completeness baseline. Given one baseline B, its precision is defined as

$$prec(B) = \frac{|B^{FQA} \cap B^{man}|}{|B^{man}|}, \tag{16}$$

where B^{FQA} represents the set of facts identified by FQA, and B^{man} represents the set of facts identified by human. Figure 1 gives the average precision of accuracy baselines for every quality class. Figure 2 gives the average precision of completeness baselines on every quality class. We can observe that our FQA can effectively find

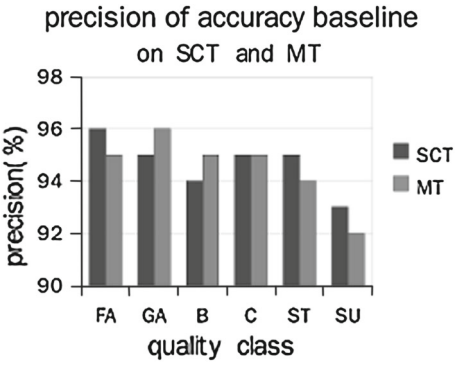

Fig. 1 Precision of accuracy baselines

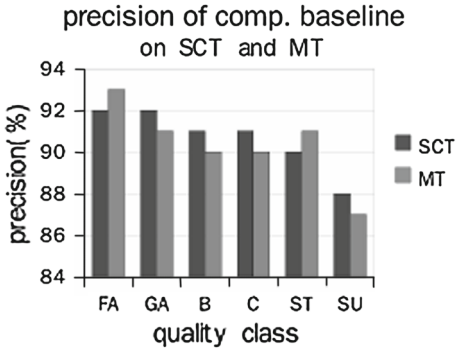

Fig. 2 Precision of completeness baselines

the baseline facts for both the accuracy and completeness. The precision of accuracy baselines on the SCT dataset ranges between 93 and 96 %, and that on the MT dataset ranges between 92 and 96 %. The precision of completeness baselines on the SCT dataset ranges between 88 and 92 %, and that on the MT dataset ranges between 87 and 93 %. We also notice that the precision of the SU class on both datasets is the lowest. This is mainly due to that the FQA cannot determine an appropriate alternative context for too short articles.

6.1.2 Effectiveness of quality dimension scores

We evaluate the effectiveness of the quality scores on the two Wikipedia datasets. Then, we evaluate the effectiveness of the quality scores on the news dataset.

I. Effectiveness on the wikipedia datasets

The FQA gives accuracy scores, completeness scores and freshness scores to all the source articles. To measure the precision of the ranking based on the scores, we borrow the Kendall correlation coefficient. Suppose that there are N articles $\{P_1, \ldots, P_i, \ldots, P_N\}$. The articles are ordered as $S^b = \{P_1^b, \ldots, P_i^b, \ldots, P_N^b\}$ according to the gold fact baselines, and ordered as $S^F = \{P_1^F, \ldots, P_i^F, \ldots, P_N^F\}$ according to the FQA. The ordered set of N objects can be decomposed into

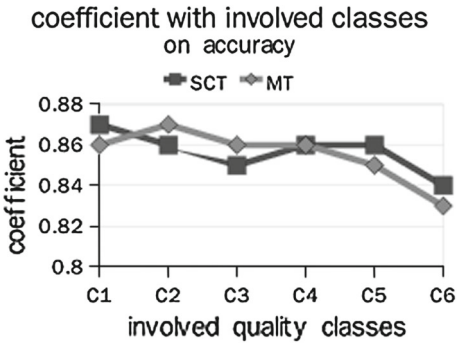

Fig. 3 Kendall coefficient on accuracy

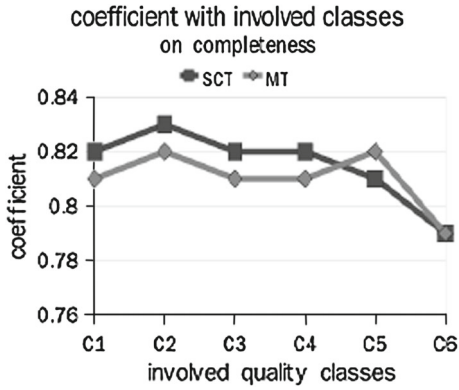

Fig. 4 Kendall coefficient on completeness

$\frac{1}{2}N \times (N-1)$ ordered pairs. Let the set of ordered pairs of S^b be OS^b, and the set of ordered pairs of S^F be OS^F. Then, the Kendall correlation coefficient is defined as

$$\tau = 1 - \frac{2 * d_\Delta(OS^b, OS^F)}{N \times (N-1)}, \tag{17}$$

where $d_\Delta(OS^b, OS^F)$ is the symmetric difference distance between OS^b and OS^F, i.e., the number of ordered pairs that belong to only one set. The coefficient measures the degree of correspondence between two rankings, whose value ranges between –1 and 1.

Figure 3 reports the values of the coefficient based on the ranking of accuracy scores when different numbers of quality classes are involved. We assume an order of ⟨FA, GA, B, C, ST, SU⟩. We use C1 denotes the set of articles in FA class, and C2 denotes the set of articles in both the GA and FA classes, and so on. Figure 4 reports the values of coefficient based on the completeness. We can observe that the values of coefficients based on the accuracy are always higher than that based on the completeness. On the SCT dataset, the values of coefficients range between 0.84 and 0.87 for assessing the accuracy, and the values of coefficients for assessing the completeness range between 0.79 and 0.83. On the MT dataset, the values of

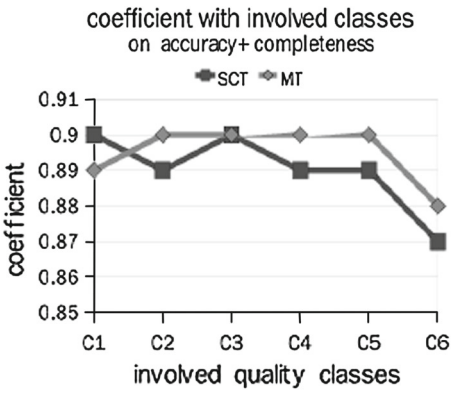

Fig. 5 Kendall coefficient on accuracy + completeness

coefficients for assessing the accuracy range between 0.83 and 0.87, and the values of coefficients for assessing the completeness range between 0.79 and 0.82.

Figure 5 reports the Kendall correlation coefficients based on the combination of accuracy and completeness. Note here the weights for accuracy and completeness are tuned as 0.64 and 0.36, respectively. Comparing Fig. 5 with Figs. 3 and 4, we can find that the FQA approach yields the best performance when the accuracy and the completeness are combined. Specifically, the values of coefficients range between 0.86 and 0.90 on the SCT dataset, and range between 0.88 and 0.91 on the MT dataset. From these figures, we can observe that our FQA performs steadily when new quality classes are included. We also notice that the values of coefficients drop slightly when the SU class is included. This is because the articles in the SU class are too short, which in turn affects the building of the alternative context.

II. Effectiveness on the news dataset

As news is sensitive to time, we report the quality assessment performance by taking into consideration the accuracy, completeness and freshness, individually or collectively. We manually identified each source article's quality score with the following steps. (1) We asked five users to weight every dimension based on its importance, and we take the average dimension weight. (2) We asked each user to give each dimension a score between 0 and 1. Then, three dimension scores are combined by each dimension's weight to produce the combined score. Here the scoring for the accuracy and completeness takes the Wikipedia quality grading scheme as references. Specifically, in reference to the six quality classes of FA, GA, B, C, ST and SU, the score of accuracy or completeness should fall into one of the six intervals [0, 0.167], [0.167, 0.334], [0.334, 0.5], [0.5, 0.667], [0.667, 0.83] and [0.83, 1]. As for the guidelines of scoring freshness, we do not report them due to space. (3) We averaged the scores given by five users for three dimensions, and they are denoted by acc, $comp$ and fre, respectively. We also calculate the average of the five combined scores, and it is denoted by acf.

We divided all the articles into 10 groups, and each group contains 10 source articles. In each group, we calculated two rankings. One is based on the scores given by the FQA, and the other is based on the scores manually given. Then, we calculated the four Kendall correlation coefficients, τ^{acc}, τ^{comp}, τ^{fre} and τ^{acf}. Here

Table 6 Kendall correlation coefficients on news dataset

Group	τ^{acc}	τ^{comp}	τ^{fre}	τ^{acf}	Group	τ^{acc}	τ^{comp}	τ^{fre}	τ^{acf}
1	0.60	0.45	0.46	0.76	6	0.64	0.69	0.50	0.83
2	0.52	0.60	0.55	0.78	7	0.61	0.64	0.46	0.82
3	0.58	0.75	0.52	0.79	8	0.53	0.56	0.41	0.76
4	0.51	0.59	0.45	0.78	9	0.67	0.55	0.47	0.84
5	0.70	0.54	0.51	0.84	10	0.60	0.67	0.40	0.75

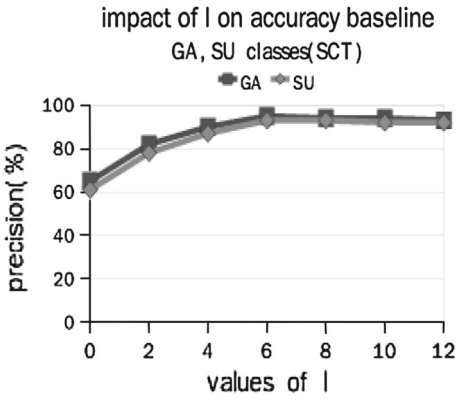

Fig. 6 Impact of l on accuracy baselines

τ^{acc}(τ^{comp}, τ^{fre}, τ^{acf}) is the coefficient based on the accuracy (completeness, freshness, combined score).

From Table 6, we can observe that the quality scores given by the FQA and those given by human have a high correlation. The values of Kendall correlation coefficients based on the accuracy scores range between 0.51 and 0.70, and those based on the completeness scores range between 0.45 and 0.75. For freshness, the values of coefficients range between 0.40 and 0.55. When three dimensions are combined, the values of coefficients range between 0.75 and 0.84. As we can observe, the FQA can effectively predict news articles' quality ratings based on the accuracy, completeness or freshness, individually. Also, it can yield the best performance when the three quality dimensions are combined.

6.2 Impact of de-duplication and context threshold on quality assessment

6.2.1 Parameters of de-duplication on accuracy, completeness and freshness

In our de-duplication approach, the parameter l has a direct effect on the performance. We evaluate the impact of l on the SCT dataset. Figure 6 reports the accuracy baseline precision for GA and SU quality classes when we varied l from 0 to 12. Figure 7 reports the completeness baseline precision for GA and SU quality classes. On average, the precision with the processing of de-duplication is higher than that without

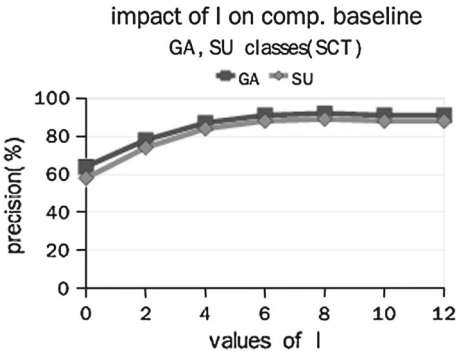

Fig. 7 Impact of l on completeness baselines

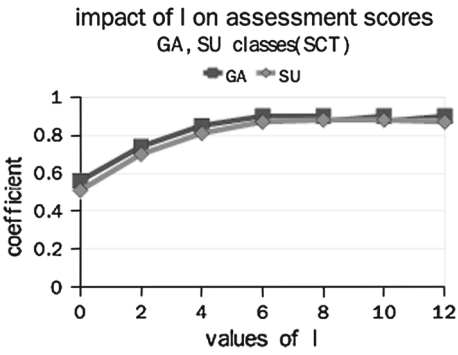

Fig. 8 Impact of l on quality assessment scores

the processing of de-duplication by a percent between 13 and 32. This is because the near-duplicate articles can distort the choice of baseline facts, thus degrading the precision. We also notice that the precision increases first rapidly then slowly with the increase of l. This is because, with the increase of l, the false negatives can be greatly reduced at first. But, when l is large, the decrease of the false negative rate is becoming very little with the increase of l. The results on other data quality classes exhibit the same trend, and we do not report them again.

Figure 8 reports the Kendall correlation coefficients based on the scores of combined accuracy and completeness by varying l from 0 to 12. We observe that, the values of coefficients with the de-duplication is higher than those without the de-duplication by between 0.18 and 0.37. This can be explained by the fact that the de-duplication helps find a more precise baselines, which in turn improve the performance of quality assessment scores. The results on other data quality classes exhibit the same trend, and we do not report them again.

6.2.2 Impact of context thresholds on the assessment of accuracy and completeness

The context threshold ϑ determines how many articles are in the alternative context, and in turn affects the calculated accuracy score and completeness score.

We varied ϑ from 0.9 to 0.3 on the SCT dataset. Figures 9 and 10 report the average context size, i.e., the number of alternative articles, with respect to the context threshold

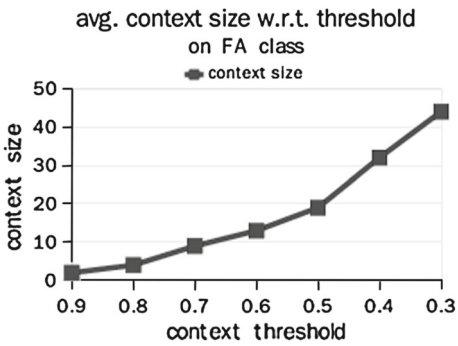

Fig. 9 Context size w.r.t. context threshold (FA)

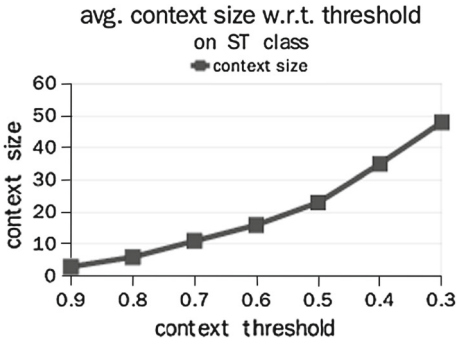

Fig. 10 Context size w.r.t. context threshold (ST)

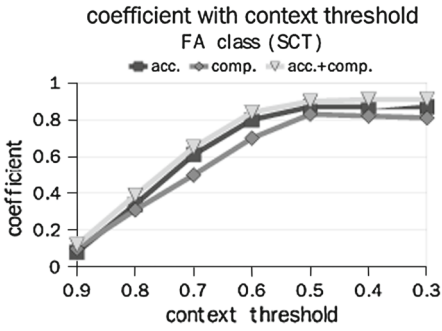

Fig. 11 Kendall coefficient w.r.t. context threshold (FA)

on the FA and ST classes, respectively. Figure 11 reports the average Kendall correlation coefficients based on different dimensions with respect to the context threshold on FA class. Figure 12 reports the average Kendall correlation coefficients based on different dimensions with respect to the context threshold on ST class. We can observe that, with the decrease of the context threshold, the coefficient is becoming larger. When the context threshold ϑ is large, the coefficient is very small. This is because the alternative context contains only a few alternative articles, and we lack enough articles to support the candidate facts of accuracy and completeness baselines. We also notice that when the context threshold is small, the decrease of the context threshold almost has no effect on the accuracy assessment, but leads to a slight decrease of

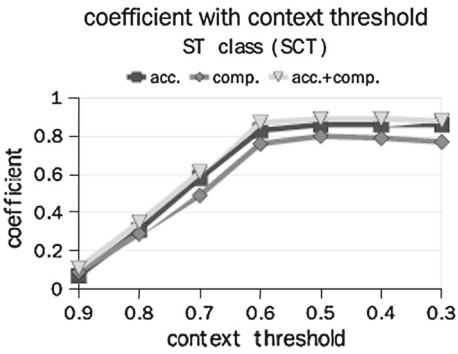

Fig. 12 Kendall coefficient w.r.t. context threshold (ST)

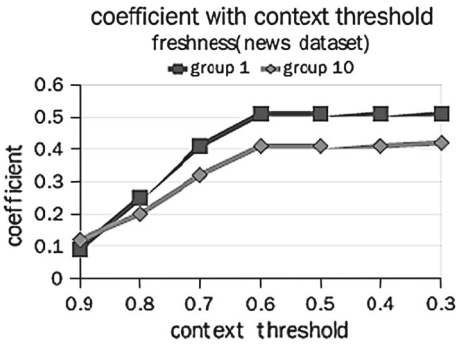

Fig. 13 Kendall coefficient of freshness w.r.t. context threshold

the coefficients of completeness. This can be explained by the fact that inclusion of more relevant articles cannot change the voting result, but it may somewhat distort the completeness baseline.

6.2.3 Impact of context thresholds on the assessment of freshness

This experiment is performed on the groups 1 and 10 of the news dataset. We concern how the Kendall correlation coefficients based on the scores of freshness change with context thresholds. Figure 13 reports the variation of the coefficient values when we varied the context threshold from 0.9 to 0.3 on the groups 1 and 10. We can observe, with the decrease of the context threshold, the coefficient first increases very rapidly then slowly. The reason is that, the larger the size is, the more likely the correct time can be captured on the Web. However, with the increase of the context size, the incremental chance of finding out the correct publishing time is becoming less and less. The results on other groups exhibit the same trend, and we do not report them again.

6.3 Comparison and complement w.r.t. previous work

6.3.1 Comparison with previous work

Our FQA approach gives data quality scores from the semantic point of view. To the best of our knowledge, most existing work assesses the Web article's data quality

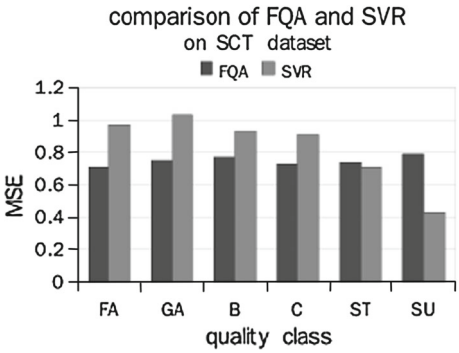

Fig. 14 Comparison of FQA and SVR based on MSE measurements

based on its lexical or syntax features. For instance, one of the state-of-the-art work uses Support Vector Regression (SVR) approach [1] to rate Wikipedia articles' quality. It first learns a SVR model, then uses it to predict an article's quality. The SVR approach uses three categories of features including text feature, review feature and network feature. In particular, text feature is divided into length feature, structure feature, style feature and readability feature. To be fair, we implemented SVR approach using the combination of Structure and Style features, which was reported to perform best.

The SVR approach gives articles quality values from 0 (SU class) through 5 (FA class). We identify the correspondence between the scores of our FQA and the Wikipedia quality classes as follows. Suppose that the experiment dataset consists of N articles, and $\sum_{i=1}^{6} N_i = N$ holds, where $N_1, N_2, \ldots, N_6$ denote the numbers of articles in FA, GA, B, C, ST, and SU classes, respectively. Again, the order based on the FQA scores is $\langle R_1, R_2, \ldots, R_N \rangle$. This means the first N_1 articles belong to FA class, the next N_2 articles belong to GA class, and so on. We use the Mean Squared Error (MSE) to measure the performance of SVR and FQA. The MSE is defined as

$$MSE = \frac{1}{N} \sum_{i=1}^{N} e^2, \tag{18}$$

where e is the error value.

We only report the comparison results on the SCT dataset as the result on other dataset behaves the same. Figure 14 reports the comparison results of SVR and FQA based on the MSE measurement. We can observe that, in general, the FQA outperforms the SVR for most of the quality classes. Specifically, from FA through C classes, the error value of FQA is smaller than that of SVR by from 0.16 to 0.28. For the ST class, the performance of FQA is fairly comparable to that of SVR. For the SU class, the SVR has a better performance. From the comparison, we find that the FQA not only assesses the data quality with the smaller error, but also gives a relatively steady performance for all the quality classes.

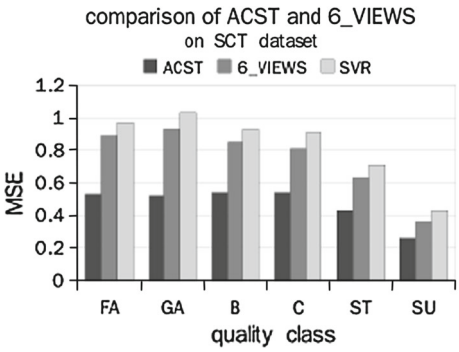

Fig. 15 Comparison of ACST and 6_VIEWS on MSE measurements

6.3.2 Evaluation of combining semantic dimensions with non-semantic features

We also notice, recently, ensemble methods are employed to enhance quality assessments by combining different views (groups) of non-semantic features [10,11]. These methods are clearly more successful if the combined views of features are independent of each other. Fortunately, the accuracy and completeness can well complement these types of methods by regarding the semantic dimensions as views of features. We evaluate an approach by combining the Accuracy, Completeness, Structure and sTyle (abbreviated by ACST). The ACST also employs the support vector regression (SVR) to train the assessment model as the 6_VIEWS proposed in [10]. In the first phase (learning level 0), we predict a source article's accuracy score and completeness score by comparing the source article with the constructed dimension baselines, and predict the structure score and style score using the support vector regression (SVR). In the second phase (learning level 1), the final quality score is obtained by the SVR according to the four views, i.e., accuracy, completeness, structure and style. In contrast, the 6_VIEWS calculates the final quality score according to six views, i.e., the length, style, structure, text readability, review history, and citation graphs.

On the SCT dataset, we evaluate the performance of 6-VIEWS and ACST in terms of the MSE measurement. Figure 15 reports the values of MSE for every quality class. To give a comprehensive view, we also include the SVR results in the figure. We observe that the ACST outperforms the 6_VIEWS for all quality classes with a great margin. Specifically, the error values of ACST are smaller than those of 6_VIEWS by from 0.1 to 0.36. All the error values are almost inferior to 0.5. This shows that the intrinsic quality dimensions can well complement the non-semantic features for quality rating.

We further investigate how the semantic dimensions, i.e., the accuracy and completeness, and the syntactical features, i.e., the structure and style, have an impact on the performance of the ACST. Table 7 reports the MSE values of ACST by excluding one view (semantic dimensions or syntactical features) in turn. Observing the table, we can find the accuracy plays a more important role than any other views on the FA, GA, B and C classes. On the SU class, the increase of the error ranges between 0.11 and 0.14 while we exclude one of the two semantic dimensions. In contrast, the increase of the error ranges between 0.26 and 0.31 while we exclude one of the two

Table 7 MSE values of ACST by excluding one quality view

	FA	GA	B	C	ST	SU
ACST	0.53	0.52	0.54	0.54	0.43	0.26
−A	0.68	0.69	0.65	0.70	0.64	0.37
−C	0.63	0.65	0.61	0.62	0.51	0.40
−S	0.59	0.61	0.59	0.63	0.68	0.57
−T	0.54	0.58	0.53	0.61	0.67	0.52

views of non-semantic features. This further verifies that the semantic dimensions and non-semantic features can complement each other well.

7 Conclusions and discussion

To make use of the vast amount of Web data, the data quality assessment of Web articles is a pressing concern. To handle this problem, we propose to assess the quality of Web articles in terms of semantics by collecting relevant knowledge in the Web community. The main contributions of the paper are as follows. First, our FQA approach is an automatic Web quality ranking solution with little human interaction by leveraging related Web knowledge. Second, it provides a viable means to assess a Web article's data quality in terms of semantics, rather than syntax, which can give a more precise quality rating. Third, the extracted semantic dimensions, i.e., the accuracy and completeness, are fairly independent of non-semantic features, which can well complement existing quality assessment works.

The FQA is a general approach to identify the Web article's quality. It is not limited to a specific type of Web articles as long as we can can find the Web article's alternative context. The experiment demonstrates the FQA can achieve a favourable performance. But, the FQA also leaves room for further improvements. First, the performance of FQA is somewhat affected by whether we can find a good alternative context on the Web. If we cannot find relevant and independent articles, the performance will deteriorate. Second, new facts cannot be much supported on the Web at first, which may more or less distort the constructed accuracy baselines. However, this situation is also compromised by the fact that, in the Web Age, the information is disseminated and transferred by various channels dramatically rapidly.

In future, we plan to take advantage of the reputation of data sources to support the baseline facts with more trusty credential, and to investigate the life cycle of facts during the evolution of articles.

Acknowledgments We sincerely thank Professor Alexandra Poulovassilis from London Knowledge Lab(LKL) for her valuable suggestions, and the anonymous reviewers for their valuable comments.

References

1. Dalip DH, Cristo M, Calado P (2009) Automatic quality assessment of content created collaboratively by web communities: a case study of wikipedia. In: Proceedings of JCDL '09, pp 295–304

2. Zeng H, A.Alhossaini M, Ding L (2006) Computing trust from revision history. In: Proc. of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services
3. Stvilia B, Twidle MB, Smith LC (2005) Assessing information quality of a community-based encyclopedia. In: Proceedings of the international conference on information quality, pp 442–454
4. Wang RY, Kon HB, Madnick SE (1993) Data quality requirements analysis and modeling. In: Proceedings of the 9th international conference on data engineering, pp 670–677
5. Louis DA, Perrochon L (1993) Towards improving data quality. In: Proceedings of the international conference on information systems and management of data, pp 273–281
6. Bouzeghoub M, Peralta V (2004) A framework for analysis of data freshness. In: Proceedings of 2004 international information quality conference on information system, pp 59–67
7. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
8. Wand Y, Wang RY (1996) anchoring data quality dimensions in ontological foundations. Communications of the ACM 39(11)
9. Foltz PW, Gilliam S, Kendall S (2000) Supporting content-based feedback in on-line writing evaluation with lsa. Interact Learn Environ 8(2):111–127
10. Dalip DH, Gonalves MA, Cristo M, Calado P (2012) On multiview-based meta-learning for automatic quality assessment of wiki articles. In: Proceedings of the 2nd international conference on theory and practice of digital libraries, pp 234–246
11. Dalip DH, Gonalves MA, Cristo M, 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, pp 543–552
12. Rassbach L, Pincock T, Mingus B (2008) Exploring the feasibility of automatically rating online article quality
13. Zeng H, Alhossaini MA, Fikes R, McGuinness DL (2006) mining revision history to assess trustworthiness of article fragments. In: Proceedings of the 2006 international conference on collaborative computing networking applications and worksharing, pp 1–10
14. Baeza-Yates R, Rello L (2012) on measuring the lexcial quality of the web. In: Proceedings of the 2nd joint WICOW/AIRWeb workshop on web quality, pp 1–6
15. Lex E, Voelske M, Errecalde M (2012) Measuring the quality of web content using factual information. In: Proceedings of the 2nd joint WICOW/AIRWeb workshop on web quality, pp 7–10
16. Li X, Meng W, Yu C (2011) T-verifier: verifying truthfulness of fact statements. In: Proceedinds of ICDE 2011:63–74
17. Parameswaran A, Rajaraman A, Garcia-Molina H (2010) Towards the web of concepts: extracting concepts from large datasets. In: Proceedings of 2010 VLDB. 3:566–577
18. Gionis A, Indyk P, Motwani R (1999) Similarity search in high dimensions via hashing. In: Proceedings of 25th international conference on very large data bases, Morgan Kaufmann, pp 518–529
19. Ohsawa Y, Benson NE, Yachida M (1998) Keygraph: automatic indexing by co-occurrence graph based on building construction metaphor. In: Proceedings of the IEEE international forum on research and technology advances in digital libraries, pp 12–18
20. Lu Y, Meng W, Zhang W, Liu KL, Yu C (2006) Automatic extraction of publication time from news search results. In: Proceedings of the 22nd international conference on data engineering workshops, p 50
21. Chen Z, Ma J, Cui C, Rui H, Huang S (2010) Web page publication time detection and its application for page rank. In: Proceedings of SIGIR'10, pp 859–860
22. Si X, Chang EY, Gyongyi Z, Sun M (2010) Confucius and its intelligent disciples: integrating social with search. In: Proceedigs of the 36th VLDB, pp 1505–1516
23. Lee H, Chang A, Peirsman Y, Chambers N, Surdeanu M, Jurafsky D (2013) Deterministic coreference resolution based on entity-centric, precision-ranked rules. Comput Linguist 39(4):885–916
24. Etzioni O, Fader A, Christensen J, Soderland S, Mausam (2011) Open information extraction:the second generation. In: Proceedings of twenty-second international joint conference on artificial intelligence, pp 3–10
25. Blanco E, Moldovan D (2011) Semantic representation of negation using focus detection. In: Proceedings of the 49th annual meeting of the association for computational linguistics, pp 581–589

Copyright of Computing is the property of Springer Science & Business Media B.V. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.