Truthfulness Analysis of Fact Statements Using the Web

Xian Li[1], Weiyi Meng[1] Clement Yu[2]
[1]Department of Computer Science, Binghamton University, USA
{xianli, meng}@cs.binghamton.edu
[2]Department of Computer Science, University of Illinois at Chicago, USA
cyu@cs.uic.edu

Abstract

Web users increasingly treat the Web as a very large knowledge base and use it to acquire all kinds of knowledge. In this article, we introduce a method that leverages the information on the Web to determine the truthfulness of any statement that attempts to state a fact. This method accomplishes the goal by checking whether there is an alternative statement that is more likely to be truthful. As a result, this method can find an alternative truthful statement when the given statement is not truthful. Our experimental results show that the proposed method is effective. In this article, we also provide some insights on how to extend this work and how to leverage this technique to estimate the trustworthiness of web pages and websites.

1 Introduction

The Web is being used by millions of users to acquire knowledge. Indeed, people can use the Web to find lots of useful knowledge such as the longest river in the world and the religion of President Barack Obama. But due to the existence of significant amount of untruthful information on the Web for various reasons such as typos, obsolete information and rumors, it is not always easy to determine the correct fact. The search result in Figure 1 is an example of a rumor-spreading article that attempts to portrait Barack Obama as a Muslim. It is important to develop useful tools that can help web users differentiate truthful and untruthful information.

In this article, we first introduce a method [1] for determining the truthfulness of any fact statement with a specified doubt unit and then discuss how to extend this work to more general situations. In this article, a *fact statement* is defined as a statement that attempts to state a fact, not an opinion. For example, "United States has 51 states" is a fact statement although the stated fact is not correct. In contrast, "Stinky Tofu is the worst dish" expresses an opinion.

We developed a system called T-verifier [4] for verifying the truthfulness of any fact statement. A user submits a fact statement, whose truthfulness is uncertain to the user, to T-verifier and indicates which part of the statement he/she is not certain about. Such a statement is called a *doubtful (fact) statement* and the uncertain part is called the *doubt unit* of the statement. For example, a doubtful statement may be "United States has [51] states", where [ ] indicates the doubt unit. T-verifier determines the truthfulness of a doubtful statement DS in two phases: (1) *Alternative statements* that are on the same topic as DS are generated from the *search result records* (SRRs) retrieved from a search engine using a query derived from DS. (2) The alternative statements are ranked based on features extracted from newly retrieved SRRs using each alternative statement as a query and

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[1]This method was first reported in [4].

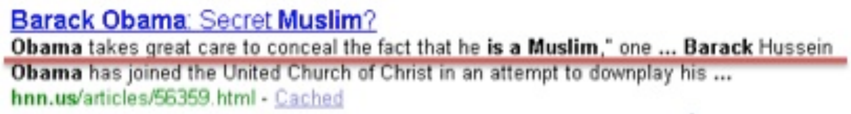

Figure 1: Results of “Barack Obama is Muslim” (Yahoo.com on 12/20/2009)

the top-ranked alternative statement is recognized as the truthful statement. In this way, T-verifier not only tries to determine the truthfulness of the doubtful statement but also tries to provide a relevant correct statement if the doubtful statement is not considered to be truthful.

The rest of the paper is organized as follows. In Section 2, we introduce a method to generate relevant alternative statements from a given doubtful statement. In Section 3, we discuss how to rank the alternative statements so as to determine the truthful statement. In Section 4, we report some experimental results. In Section 5, we briefly review some related works. In Section 6, we discuss some future extensions to the current work. Finally, a brief summary is given in Section 7.

2 Alternative Statements Generation

Let DS denote a given doubtful statement with a specified doubt unit DU. A doubt unit can be a single word, a phrase, a name entity, a number or a date, etc. Alternative statements to DS are very important to the truthfulness judgment, because they provide different versions of the fact and the truthful one is probably one of them. Different alternative statements should be different on the doubt unit. We call the term(s) in place of the doubt unit *alternative unit* (*alter-unit* for short). A relevant alternative statement should have the following properties:

- **Same Topic**. It should cover the same topic as the doubtful statement.
- **Term sense/type closeness**. Each alter-unit should have close word sense with the DU, in terms of both data type and semantic meaning.

We turn the problem of finding alternative statements into the problem of looking for relevant alter-units. To find relevant alter-units, we partition DS into two parts: *doubt unit* and *topic units*. The topic units contain content words in DS except the doubt unit. The topic units provide the appropriate context for finding relevant alter-units. We use the topic units to form a query (called *topic query*) and submit it to a search engine (T-verifier currently uses Yahoo!) to retrieve a set of SRRs. Each SRR consists of a title and a snippet. Let D be the set of N SRRs collected. Every term (including phrase) in these SRRs is considered a candidate alter-unit and we rank all candidates using seven features to select the alter-units from the top ranked candidates.

Our observation shows that there are two types of co-occurrence information with alter-units that can be used. First, relevant alter-units often co-occur with the topic units as the topic units provide the appropriate context for an alter-unit to be relevant. Second, relevant alter-units often co-occur with the doubt unit. This is because when people have doubt about a fact, they often mention other possible answers to the fact. Among the following seven features used in T-verifier, the first four capture the co-occurrence relationship between the topic units (i.e., the topic query Q) and a term T from SRRs (i.e., a candidate alter-unit) and the last three capture the relevance between a term T and the given doubt unit DU.

- **Result coverage (RC)**: It is the percentage of SRRs in D that contain T. If T appears in a higher percentage of the SRRs, it is more likely to be a good alter-unit. RC measures how frequently T co-occurs with the topic units.
- **Result Query Relevance (RQR)**: It is a measure of the relevance of the SRRs with respect to Q that contain T. If T appears in more relevant SRRs, T is more likely to be a good alter-unit. An SRR that contains more topic units is considered to be more relevant to Q.
- **SRR ranking (Rrank)**: It is a measure reflecting the ranks of the SRRs that contain T. Intuitively, if T appears in higher ranked SRRs, it is more likely to be a good alter-unit.

- **Term distance (TD)**: Intuitively, terms that appear closer to the topic units are more likely to be related to the topic units. To capture the proximity information between terms, we consider the size of the smallest window of consecutive words in each SRR (title and snippet are considered separately) that covers all the topic units contained in the SRR as well as T. The TD measure considers the length of the minimum window as well as the number of topic units covered in this window. When none of the topic units appears in the same sentence with T, we consider its TD for this SRR to be 0.
- **Data type matching (DM)**: Each valid alter-unit is required to have the same data type as the doubt unit in T-verifier. Several data types are supported in T-verifier, including date, time, telephone number, email address, person name, place name (e.g., name of state, city and attractions), and number. All others are considered as strings.
- **Sense closeness (SC)**: Intuitively, terms that have a closely related meaning as the doubt unit are good candidate alter-units. T-verifier utilizes WordNet to capture the sense closeness between two terms. Three special relations are considered in T-verifier (direct hypernym/hyponym, instance hyponym, and sibling) and they are treated differently when sense closeness is calculated [4].
- **Term local correlation (TLC)**: It measures the correlation between T and DU. As mentioned earlier, relevant alter-units are often mentioned together with the doubt unit. T-verifier uses the correlation coefficient formula in [9] to measure the correlation strength between T and DU.

T-verifier selects alter-units by combining the seven features described above in two steps. It first filters the candidate terms by data type matching (DM), i.e., only those terms that match the data type of the DU will be considered further. In the second step, it ranks each remaining candidate term using the other six features. The ranking score of T is a weighted sum of the scores based on these six features. The best values for these weights can be determined empirically. Note that if DU is one of the terms in the retrieved SRRs, it will participate in the ranking just like other terms in these SRRs.

As an example, for the doubtful statement ”Barack Obama is a Muslim” with DU = ”Muslim”, the top-5 ranked alter-units found by the above method are ”christian”, ”muslim”, ”president”, ”black”, and ”senator”. Note that ”president” is the alter-unit with the highest result coverage, however, it is not the top-ranked alter-unit when other features are considered (it has low sense closeness with ”Muslim”).

According to our experiment, the above alter-unit ranking method performs well on our test dataset. For each of the 50 sample statements tested, this method always ranks the truthful alter-unit among the top five results. Furthermore, in 31 cases, the truthful alter-unit is ranked at the top. Each of the top-5 ranked alter-units is used to generate one alternative statement by taking the place of the doubt unit in the doubtful statement. As a result, we only need to consider these five alternative statements in the statement truthfulness verification phase for each doubtful statement. Note that DU may or may not be one of the top-5 alter-units depending on whether it appears in the retrieved SRRs and what its ranking score is.

3 Alternative Statements Truthfulness Verification

In T-verifier, the verification phase is carried out by ranking the top five alternative statements generated in the previous phase and selecting the top-ranked statement as the truthful one. The phase described in Section 2 does provide a ranking of all the alter-units which can be regarded as the ranking of the corresponding alternative statements. Unfortunately this ranking is not sufficiently accurate for practical use as it ranks only 62% of the truthful alter-units at the top among the 50 doubtful statements we tested. Thus a better solution is needed. In this section, we introduce the truthfulness verification process for a given group of alternative statements as used in T-verifier. This process has three steps: first, send every alternative statement as a query to a search engine and collect relevant SRRs; second, employ a number of basic rankers and use each of them to generate a ranking list of the alternative statements based on the newly collected SRRs; third, use a rank merging algorithm to merge the rank lists into a combined list.

Intuitively, verifying the truthfulness of a statement requires relevant evidence. For each alternative statement, T-verifier submits it as a query to a search engine to retrieve the top 200 SRRs and uses the information from these SRRs to rank the alternative statements. An implicit assumption underpinning this approach is that truthful information is usually more widespread than untruthful ones on the Web and tends to be consistent.

The Basic Rankers

Given a set of alternative statements and the retrieved SRRs, the basic rankers used in T-verifier to rank the alternative statements are introduced below.

- **Alter-Unit Ranker (AUR)**: In Section 2, we discussed a method for selecting alter-units from the SRRs retrieved by the topic query. The alter-units are ranked based on a number of features collected from SRRs. Since each alter-unit corresponds to an alternative statement, this ranking of the alter-units from this method can be considered as a ranking of the alternative statements, and we call it Alter-Unit Ranker (AUR).
- **Hits Ranker (HR)**: A seemingly reasonable method is to rank the alternative statements by the number of hits they retrieve from a search engine. Web users often use this method to do quick truthfulness verification. We call this method the Hits Ranker. One potential problem of this ranker is that it implicitly assumes that all the SRRs retrieved by an alternative statement support this alternative statement. However, it is possible that some of the SRRs are actually against it (i.e., saying it is not true). As a result, the number of hits may not be a reliable indicator for judging the truthfulness of an alternative statement.
- **Text Feature Rankers (TFR)**: Text feature rankers are a set of 4 rankers measuring the relevance between the alternative statements and each SRR. In Section 2, we discussed several text features used to measure the relevance between an SRR and the topic query. There are mainly four features involved: Result Coverage (RC), Result Query Relevance (RQR), SRR ranking (Rrank), and Term Distance (TD). In TFR, we reuse these features to rank the alternative statements. However, the rankings generated by TFR here are different from those by the alter-unit generation phase because these four features are now used against a different set of SRRs, which is retrieved by a different query (i.e., the full alternative statement under consideration).
- **Domain Authority Ranker (DAR)**: Some researchers have observed that web pages published by certain domains are more likely to be truthful, such as ".gov", ".edu", etc [2]. This is because websites in these domains claim responsibility for their published materials. Based on this observation, a higher weight is assigned to a domain that is considered to be more trustworthy. In T-verifier, the weight for each domain is learned from the SRRs in this domain that are retrieved by sample truthful and untruthful alternative statements [4]. The rank position given by DAR for an alternative statement is determined by the number of SRRs from each domain retrieved by the alternative statement and the weights of these domains.

Different basic rankers may produce different rankings of the alternative statements. Table 1 shows the ranks of five alternative statements corresponding to the alter-units listed in the first column by the seven basic rankers. Thus, a method is needed to combine the different rankings into a single overall ranking. Based on the evaluation result of several rank merging techniques (including a probabilistic method, a machine learning method, several variations of the Borda method and several variations of the Condorcet method (see [4] for more details)), a variation of the Borda method called Weighted Positional Borda is found to have the best performance and is used in T-verifier. This method is described below.

The basic Borda algorithm [1] works as follows. Suppose n candidates (they are the top ranked alternative statements in our case) are being ranked. If an alternative statement is ranked at the i-th position, it receives a score of $n - i + 1$, $1 \leq i \leq n$. The combined score of an alternative statement is the sum of the scores it receives from all the basic rankers. Finally, the alternative statements are ranked in descending order of the combined scores. One weakness of this algorithm is that it treats all basic rankers equally, ignoring the fact that some basic rankers may be better than others.

One way to improve the basic Borda algorithm is to differentiate the importance of different basic rankers and assign a weight to each ranker, as was also done in [1]. In T-verifier, the weight for each basic ranker is

Table 1: Ranks of Alternative Statements by Basic Rankers

Alter-units	AUR	HR	RC	RQR	Rrank	TD	DAR
christian	1	3	1	1	1	1	2
muslim	2	2	2	3	2	2	1
president	3	1	3	2	3	5	3
black	4	4	4	4	4	3	5
senator	5	5	5	5	5	4	4

obtained through training [4]. After the weight for a basic ranker BR is obtained, the ranking score an alternative statement receives from BR in the basic Borda algorithm is revised by multiplying the weight of BR. Specifically, if the weight of BR is w, then the alternative statement ranked at the i-th position by BR will receive a score of $(n - i + 1) * w$. This version of the Borda algorithm is called the Weighted Borda algorithm [4].

Another weakness of the basic Borda algorithm (as well as that of the Weighted Borda algorithm) is the way it assigns a score to each ranking position. While it is reasonable to assign a larger score to a higher position (the top-ranked result has the highest position), there is little scientific justification on why each next lower position should get exactly one point less than the position above. In T-verifier, the score assigned to each position by a basic ranker is the estimated probability that an alternative statement assigned to this position by this basic ranker is truthful. This probability is called the *position probability* of the basic ranker and it can be estimated from training i.e., from ranking the alternative statements of a set of sample doubtful statements with known truthful answers [4]. This variation of the Borda algorithm is called the Positional Borda in [4].

Finally, the Weighted Positional Borda algorithm used in T-verifier is a combination of the Weighted Borda algorithm and the Positional Borda algorithm. For a basic ranker BR with weight w, the final score received by its i-th ranked alternative statement is $p_i * w$, where p_i is BRs position probability for the alternative statement ranked at position i. Once again, the combined score of each alternative statement is the sum of the final scores it receives from all the basic rankers. The alternative statement with the highest combined score is selected as the truthful statement for the given doubtful statement.

For the ranks by individual basic rankers shown in Table 1, all three Borda-based rank merging algorithms rank "Barack Obama is a Christian" at the top among the five alternative statements.

4 Experimental Evaluation

T-verifier was evaluated using 50 doubtful statements [2] transformed from 50 factoid questions from TREC-8 and TREC-9 Question Answering track [10]. Half of these statements are truthful while the other half are untruthful. The answer part in each statement is specified as the doubt unit. The proposed methods described in Sections 2 and 3 are used to determine the truthfulness of each statement and compare the result with the ground truth to compute the precision of the proposed methods.

To evaluate the alternative statement generation method described in Section 2, half of the 50 doubtful statements are randomly selected as training set and the remaining half as testing set. The method is performed on the top 200 SRRs for each statement. The results in Table 2 show that the method described in Section 2 can always rank all truthful alternative statements among the top 4 candidates for our dataset. A 10-fold cross-validation yielded similar result [4].

Based on the result on alterative statements generation, it is sufficient to compare the top-5 alternative statements to determine which one is most likely to be truthful and consequently decide whether the doubtful statement is truthful. The precisions of the seven basic rankers introduced in Section 3 vary from 0.2 to 0.66 (AUR 0.62, HR 0.2, TFR(TD) 0.32, TFR(RC) 0.66, TFR(RQR) 0.6, TFR(Rrank) 0.62, and DAR 0.2) based on the 50 doubtful statements. Overall, the precisions of individual basic rankers are not very good. The hit ranker and the domain authority result ranker performed especially poorly for our dataset. Combining ranks from the basic

[2] This dataset is available at http://cs.binghamton.edu/∼xianli/doubtful_statements.htm.

Table 2: Performance of Alternative Statements Generation

Training dataset		Testing dataset	
Total cases	25	Total cases	25
Truthful one as top 1st	17	Truthful one as top 1st	14
Truthful one as top 2nd	7	Truthful one as top 2nd	9
Truthful one as top 3rd	0	Truthful one as top 3rd	1
Truthful one as top 4th	1	Truthful one as top 4th	1
Truthful one as top 5th	0	Truthful one as top 5th	0

rankers using the Weighted Positional Borda algorithm as employed by T-verifier yields a precision of 0.9 (or 0.94 if multiple correct answers are allowed for a doubtful statement) based on 10-fold cross-validation.

5 Related Works

The "Honto?Search" system [11, 12] is another system aimed at verifying the truthfulness of fact statements. The basic idea of this system is similar to that of T-verifier in that it looks for alternative statements via a search engine and finds most likely truthful one from them. The method is based on hits numbers, temporal information (i.e., when a page was published) [11] and sentimental factors [12].

Fact statement truthfulness verification discussed in this article is related to answer verification in question-answering (QA) systems. The method proposed in [5, 6] also uses co-occurrence information between the keywords in a question and each extracted candidate answer to rank different candidate answers. To the best of our knowledge, very few papers address the answer verification problem directly, although many systems incorporate answer verification into their candidate answer ranking component. The ranking component scores candidate answers based on certain features [8, 13], and considers the answer with the highest score as the most likely correct one [3].

T-verifier differs from the existing solutions in the following aspects. (1) T-verifier operates in two carefully designed phases (i.e., alternative statement generation and statement truthfulness verification) which explore different sets of features, including new features not used before such as the semantic closeness between the doubt unit and each alter-unit. None of the existing solutions used all of these features in a single solution and used them like in T-verifier. None of them has studied both phases as comprehensively as in T-verifier. In fact, most of them studied only one of these phases. (2) Our alternative statement generation method is able to rank the truthful alter-unit among the top 5 results in our experiment. To the best of our knowledge, similar results have not been reported before. (3) Based on seven basic rankers and the Weighted Positional Borda rank merging method, T-verifier is able to achieve a precision of about 90%.

Yahoo!Answers (http://answers.yahoo.com/) and answers.com (http://www.answers.com/) are two Web QA systems based on completely different technology. They ask users to post questions, provide answers to any question, and vote for existing answers. This type of systems will not be able to provide answers to questions that have not been posted and answered. Furthermore, for answers with few votes, the accuracy is likely to be low. We tested the 50 doubtful statements in our dataset on these two systems in May 2011. For answers.com, there were no answer for 6 statements and for 4 additional statements, the answers were wrong. For Yahoo!Answers, there were no answer for 18 statements and for 12 additional statements, the answers were wrong.

It is worth noting that fact statements as studied in this article and questions in QA systems have significant differences. First, there is a specified doubt unit in each doubtful statement which enables certain features to be utilized such as the sense of the doubt unit. In contrast, no specific doubt unit is given in questions. Second, it is possible to expand the doubtful statements to allow multiple doubt units to be specified (see Section 6). Questions in QA systems, on the other hand, can target only one doubt at a time. Third, many fact statements cannot be easily converted to WH-questions used in QA systems. These differences lead to different solutions for the two types of systems (truthfulness verification and QA).

6 Future Directions

The work introduced in this article can be extended in a number of ways to either handle more general doubtful statements or to make the technique more useful to web users. Some ideas are introduced below.

Process doubtful statements that have multiple truthful results: For many doubtful statements, the truthful alter-unit is not unique. For example, since time-sensitive statements often have different truthful results at different times, they can be considered as a special case of doubtful statements with multiple truthful results. Many time-insensitive statements may also have multiple results. For example, doubtful statement "Barack Obama was born in [Kenya]" has multiple truthful units: "Honolulu", "Hawaii", "Honolulu, Hawaii", "the United States", etc. Challenging issues here include how to identify doubtful statements that have multiple truthful results and how to find all the truthful results accurately for such statements.

Allow multiple doubt units in a doubtful statement: Sometimes a user may specify multiple doubt units in a doubtful statement. For example, in "President Nixon visited [South Africa] in [1970]", two doubt units "South Africa" and "1970" are specified. An interesting issue is how to process such type of doubtful statements. Clearly processing each doubt unit independently will not ensure the generation of statements that are truthful on all doubt units. For this type of doubtful statements, we need to select the alter-units for different doubt units in synchronization but also deal with the possibility that multiple combinations of alter-units are truthful.

Process doubtful statements that have no specified doubt unit: Such statements may either be entered by users or extracted directly from web pages. The ability to determine the truthfulness of any fact statement without a pre-specified doubt unit can open the door for examining claims made in web pages automatically. A naive method is to treat each content word or phrase in such a statement as a doubt unit, one at a time, and process the corresponding doubtful statement using the technique introduced in this article. We can also generate doubtful statements by treating combinations of content words and/or phrases as doubt units. While this method may work, it is not at all efficient, especially for long fact statements as the number of content words/phrases and their combinations can be quite large. An interesting issue is how to automatically and efficiently identify worthy doubt unit(s) in any given fact statement using the data on the Web?

Determine the truthfulness of fact statements on the Web: It would be nice if the truthfulness of every fact statement on the Web can be verified (or corrected) as this can benefit all web users. The problem is that the number of fact statements that exist in web pages is huge and new fact statements will keep popping up at rapid speed, and the verification process described in this article is complex. As a result, it is not realistic to use the described method to process all fact statements independently. What we need is an automatic, effective and scalable solution. One idea is to focus only on interesting fact statements, rather than all fact statements. How to identify all interesting fact statements is an open research problem. One type of interesting fact statements is *controversial fact statements*. A fact statement FS is a controversial fact statement if there exists another fact statement FS* on the Web such that FS and FS* state contradictory facts. Another idea is to save all fact statements whose truthfulness or untruthfulness has been verified and then use these statements to determine the truthfulness of new fact statement whenever possible before trying the more expensive method.

Estimate the trustworthiness of web pages/websites: The ability to determine the truthfulness of fact statements opens up a new way to estimate the trustworthiness of web pages and websites. Specifically, given a web page or a website, we can estimate its trustworthiness by estimating the percentage of the fact statements in the web page or websites that are truthful. An interesting issue here is how to take into consideration the situations where different fact statements and different types of "errors" (e.g., typo vs. obsolete information) may contribute to trustworthiness differently? Another issue is how to combine different trust models into an integrated model? More specifically, there are other techniques to estimate the trustworthiness of web pages and websites (such as using PageRank [7]/TrustRank [2] and considering the domains, e.g., documents/sites in the

.edu domain are often considered to be more trustworthy than those in the .com domain [2]). We need to figure out how to combine these existing trust models with the statement truthfulness based model.

7 Summary

In this article, we introduced a method to analyze the truthfulness of fact statements with a specified doubt unit using the data on the Web. This method consists of two main phases, with the first trying to identify the most likely relevant alternative statements and the second trying to rank them. The T-verifier system implemented based on the introduced method has an accuracy of about 90% according to our experimental evaluation. We also introduced several future research directions which can lead to a system capable of processing more general fact statements and to applications that can improve the chance for ordinary users to obtain more reliable and trustworthy information on the Web.

Acknowledgements

This work is supported in part by the following NSF grants: IIS-0842608, IIS-0842546 and CNS-0958501.

References

[1] J. A. Aslam, M. H. Montague: Models for Metasearch. ACM SIGIR Conference, pp.275-284, 2001.

[2] Z. Gyngyi, H. Garcia-Molina, J. Pedersen: Combating Web Spam with TrustRank. International Conference on Very Large Data Bases (VLDB), pp.576-587, 2004.

[3] C. T. Kwok, O. Etzioni, D. S. Weld: Scaling question answering to the web. In ACM Transactions on Information Systems, 19(3): 242-262, 2001.

[4] X. Li, W. Meng, C. Yu. T-verifier: Verifying Truthfulness of Fact Statements. IEEE ICDE Conference, pp.63-74, 2011.

[5] B. Magnini, M. Negri, R. Prevete, H. Tanev: Is It the Right Answer? Exploiting Web Redundancy for Answer Validation. ACL 2002: 425-432.

[6] B. Magnini, M. Negri, R. Prevete, H. Tanev: Mining Knowledge from Repeated Co-Occurrences: DIOGENE at TREC 2002. TREC 2002.

[7] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bring Order to the Web. Technical Report, Department of Computer Science, Stanford University, 1998.

[8] X. Qiu, B. Li, C. Shen, L. Wu, X. Huang, Y. Zhou: FDUQA on TREC2007 QA Track. Sixteenth Text REtrieval Conference (TREC), 2007.

[9] C. Silverstein, M. R. Henzinger, H. Marais, M. Moricz: Analysis of a Very Large Web Search Engine Query Log. SIGIR Forum 33(1): 6-12, 1999.

[10] http://trec.nist.gov/data/qamain.html

[11] Y. Yamamoto, T. Tezuka, A. Jatowt, K. Tanaka: Honto?Search: Estimating Trustworthiness of Web Information by Search Results Aggregation and Temporal Analysis. APWeb/WAIM 2007: 253-264.

[12] Y. Yamamoto, T. Tezuka, A. Jatowt, K. Tanaka: Supporting Judgment of Fact Trustworthiness Considering Temporal and Sentimental Aspects. In WISE 2008: 206-220.

[13] Z. Zheng: AnswerBus Question Answering System. HLT 2002: 399-404.