Verification of Fact Statements with Multiple Truthful Alternatives

Xian Li[1], Weiyi Meng[1] and Clement Yu[2]
[1]*Computer Science Department, Binghamton University, Binghamton, NY, U.S.A.*
[2]*Computer Science Department, University of Illinois at Chicago, Chicago, IL, U.S.A.*

Keywords: Web Text Mining, Truth Finding.

Abstract: When people are not sure about certain facts, they tend to use the Web to find the answers. Two problems make finding correct answers from the Web challenging. First, the Web contains a significant amount of untruthful information. Second, currently there is a lack of systems/tools that can verify the truthfulness or untruthfulness of a random fact statement and also provide alternative answers. In this paper, we propose a method that aims to determine whether a given statement is truthful and to identify alternative truthful statements that are highly relevant to the given statement. Existing solutions consider only statements with a single expected correct answer. In this paper, we focus on statements that may have multiple relevant alternative answers. We first present a straightforward extension to the previous method to solve such type of statements and show that such a simple extension is inadequate. We then present solutions to two types of such statements. Our evaluation indicates that our proposed solutions are very effective.

1 INTRODUCTION

Many users use the Web to find answers when they are not sure about certain facts. However, there are problems that make finding correct answers from the Web challenging. First, currently there is a lack of high quality systems/tools that can verify the truthfulness or untruthfulness of a random fact statement and also provide alternative answers, although there is an increasing interest in developing such tools (e.g., *Yahoo! Answers* and *Answers.com*). Second, the Web contains a significant amount of untruthful information, ranging from unintended errors (e.g. typo), obsolete information, misconception spread from the past, and intentional rumors.

It is fairly easy to find examples where top search results provide contradictory information regarding the same fact. Figure 1 shows the top three results of searching "Barack Obama was born in" on Yahoo!. The first two search result records (each consists of a title and a snippet), SRR for short, claimed President Obama was born in Honolulu, Hawaii, whereas the third record claimed that he was born in Kenya. With the Web containing mixed truthful and untruthful information, effective methods that can distinguish truthful information from untruthful ones are needed.

In paper (Li et al., 2011), a system known as T-verifier was introduced. It allows a user to submit a doubtful statement $\mathcal{S}$ together with a part of $\mathcal{S}$ the user has doubt on (called **doubt unit**). An example of a doubtful statement is "Barack Obama was born in [Kenya]", where [ ] indicates the doubt unit. T-verifier aims to determine whether $\mathcal{S}$ is truthful and give the most likely truthful alternative statement if $\mathcal{S}$ is untruthful. In a nutshell, T-verifier works as follows. It first tries to find alternative statements of the same topic as the doubtful statement from the search result records (SRRs) retrieved from a search engine using a **topic query** (it is formed from the doubtful statement by removing the doubt unit from it). Terms replacing the doubt unit in alternative statements are called *alternative units* or alter-units (Li et al., 2011). Then, it ranks alternative statements based on analyzing new SRRs retrieved by a search of each alternative statement and considers the top ranked statement as truthful. T-verifier achieved a 90% precision on the doubtful statements in the experiment reported in (Li et al., 2011). However, a significant limitation of T-verifier is that it can only process doubtful statements that have a **s**ingle **t**ruthful **a**lternative (we will call such statements as ***STA statements*** in this paper). That is, T-verifier cannot evaluate doubtful statements with **m**ultiple **t**ruthful **a**lternatives. We will call these statements as ***MTA statements***.

In reality, many fact statements have more than one truthful alternative statement. For example, state-

Li, X., Meng, W. and Yu, C.
Verification of Fact Statements with Multiple Truthful Alternatives.
In *Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016) - Volume 2*, pages 87-97
ISBN: 978-989-758-186-1

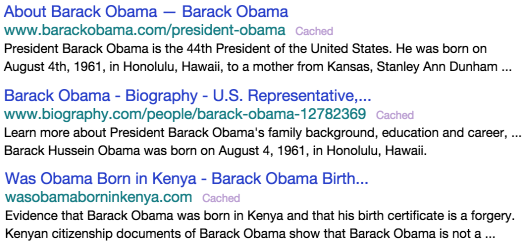

Figure 1: President Obama's Birthplace Contradiction (collected from Yahoo! on 05/31/2015).

ment "Barack Obama was born in [Kenya]" has multiple truthful alternatives, each with [Kenya] being replaced by one of the following: "Honolulu", "Hawaii", "Honolulu, Hawaii", and "United States". Processing doubtful statements with multiple truthful alternatives is to identify **all** truthful alternatives. In this paper, we focus on MTA statements.

The problem of processing MTA statements is significantly more challenging than that of processing STA statements. For STA statements, it is often sufficient to rank all alternative statements and select the top-ranked one as the truthful statement. For MTA statements, a possible straightforward extension to the previous method is to first rank all alternative statements, and then determine an integer k and consider the top-k ranked alternative statements as truthful. We call this method the *Top-k method* in this paper.

This paper makes the following contributions:

- We present and evaluate the Top-k solution to the MTA statements. We show that the Top-k solution is inadequate in processing various MTA statements, which makes it necessary to develop new solutions for MTA statements.
- We propose several solutions to process two types of MTA statements (*compatible concepts* and *multi-valued attributes*). Our solutions explore semantic and statistical relationships among alternative answers and use them to derive a set of inference rules for inferring the truthfulness of one alter-unit from that of another. These inference rules are represented in two matrices.
- We conduct extensive experiments to evaluate the proposed solutions and the effectiveness of several important solution components (e.g., the effectiveness of local correlation versus that of global correlation). The experimental results show that these solutions outperform the Top-k solution significantly. The accuracy (both recall and precision) of our best solution is above 90%.

The rest of the paper is organized as follows. In Section 2, we present and evaluate the Top-k solution. In Section 3, we introduce two types of MTA statements. In Section 4, we discuss useful relationships between alter-units and the inference rules derived from these relationships. In Section 5, we present three algorithms for processing two types of MTA statements. In Section 6, we present the experimental evaluation. In Section 7, we review related works. We conclude the paper in Section 8.

Table 1: Performance of Top-k method.

	# Total selected	# Total alternatives	Precision	Recall	F1
LSG	140	107	0.76	0.74	0.75
LPG	182	125	0.68	0.68	0.76
FSG	178	128	0.71	0.88	0.79

2 THE Top-k SOLUTION

We first describe several variations of this solution and then provide an evaluation of these variations.

The key problem for the Top-k method is how to determine the appropriate value for k. A reasonable way to determine k is to analyze the distribution of the ranking scores of the alternative statements that are used to rank the alternative statements. We implemented the same algorithm as that used in (Li et al., 2011) to compute the ranking scores of the alternative statements for this experiment. The value for k should be chosen such that the ranking scores of the top-k ranked alternative statements are close to each other but the ranking score of the $(k+1)$-th ranked alternative statement is "significantly lower" than that of the k-th ranked one.

In this paper, we consider the following three ways to determine the value for k.

1. *Largest Score Gap (LSG)*: Compare the gap between the scores of each pair of consecutively ranked alternative statements and choose the largest gap as the cut-off point. Specifically, let G_i be the difference of the ranking scores of the i-th and the $(i+1)$-th ranked alternative statements. Then this method sets $k = argmax_i\{G_i\}$ (if there are multiple such k, use the smallest one).
2. *Largest Percentage Gap (LPG)*: LPG is similar to LSG except that percentage gap of scores is used. The percentage gap between two ranking scores S_i and S_{i+1} of two consecutively ranked alternative statements is defined to be $(S_i - S_{i+1}) \,/\, S_i$.
3. *First Significant Gap (FSG)*: We define significant score gap in terms of the logarithm of the ratio of the scores of two consecutively ranked alternative statements. Specifically, if $log_b(S_i/S_{i+1}) > 1$, the score gap is considered to be significant, and k is set to be the smallest (i.e., the first) i that satisfies

this condition. We tune the base b of the logarithm from 1.3 to 5 and choose the one that achieves the best F-score.

In order to evaluate the precision of the Top-k method, we need a set of MTA statements with specified doubt units. We use the factoid questions from TREC-8, TREC-9 and TREC 2001 Question Answering Track as the experimental data repository which contains a large number of fact questions as well as standard answers. In our experiment, we select 50 questions with multiple answers from the QA track and transform them into doubtful statements with either correct or incorrect doubt units. Besides, we identify all (actually up to 5) truthful alternatives for each of the MTA statements. Overall, we find a total of 143 truthful alternative statements for the 50 MTA statements. The largest number of truthful alternative statements belonging to one MTA statement is 5 and smallest is 2. On average, each MTA statement has approximately 2.8 truthful alternatives. Section 6.1 provides more details about this dataset.

We evaluated the three methods, i.e. LSG, LPG and FSG, and their precisions, recalls and F-scores are shown in Table 1. For method FSG, the logarithm base used is 1.5. The second column shows the total number of alternative statements each method selects as truthful. The third column shows the number of selected alternative statements that are actually truthful. From Table 1, we can see that the FSG method has the best overall performance but it still has a somewhat low F-score at 0.79.

3 TWO TYPES OF MTA STATEMENTS

While all MTA statements share the common property of having multiple truthful alternative statements, further analysis of sample MTA statements reveals that MTA statements can be classified into several different types (due to space limitation, they will not be discussed here). In this section, we provide our analysis of the two types of MTA statements.

- *Type 1: Compatible Concepts (CC)*

For each MTA statement of this type, its truthful alter-units are compatible to each other. Usually, these alter-units either are equivalent to each other (i.e., synonyms) or correspond to the same basic concept but with different specificity/generality (i.e., hyponyms/hypernyms) or with different granularity (i.e., one is a part of another).

Consider the first example in Table 2. We know that Barack Obama was born in Honolulu, Hawaii.

Table 2: Example of MTA statements.

Type	Doubtful statements	Truthful alternatives
CC	Barack Obama was born in [Kenya].	Honolulu, Hawaii, United States
MVA	[Edwin Krebs] won Nobel Prize in medicine in 1992.	Edwin Krebs, Edmond Fischer

Therefore, both "Honolulu" and "Hawaii" are truthful alter-units. They both refer to the same basic concept "place" and "Honolulu" is a part of Hawaii as Honolulu is a city in Hawaii. Note that compatible concept covers many practical situations, including location (see the above example), time (e.g., "2015" is more general than "2015 July"), many types of product (e.g., "Toyota" is more general than "Toyota Camry"), etc. An example for equivalent alter-units in a doubtful statement is "Queen Elizabeth II resided in [United Kingdom]". Correct alter-units include "United Kingdom", "England" and "Great Britain".

- *Type 2: Multi-Valued Attributes (MVA)*

The truthful alter-units of this type of MTA statements correspond to different values of a *multi-valued attribute* in a database. A multi-valued attribute may have multiple values for a given entity (record). Examples of multi-valued attributes include multiple authors of a book, co-stars of the same movie, and multiple official languages of a country. In the second example in Table 2, two US biochemists "Edwin Krebs" and "Edmond Fischer" shared the 1992 Nobel Prize in medicine (they are values of the multi-valued attribute "Recipients" of a Nobel Prize record); therefore both of them are truthful alter-units.

4 ALTER-UNITS RELATIONSHIPS AND INFERENCE RULES

T-verifier (Li et al., 2011) processes each doubtful statement $\mathcal{S}$ in two steps: it first forms a topic query by removing doubt unit from $\mathcal{S}$, submits the query to a search engine (e.g., Yahoo!), extracts the alter-units that are possibly truthful from the retrieved SRRs, and forms the alternative statements from these alter-units by replacing the doubt unit in $\mathcal{S}$ by one of the alter-units); then it sends each alternative statement to the search engine, collects relevant SRRs and uses them to rank the alternative statements. In this section, we assume that a list of ranked alter-units corresponding to the ranked alternative statements has been obtained using the method in (Li et al., 2011). Let $L_{AU} = (AU_1, AU_2, \ldots, AU_n)$ denote this list of the ranked alter-units

for the $\mathcal{S}$ under consideration. In this work, we always include the doubt unit into L_{AU}.

When a doubtful statement has multiple truthful alter-units, these alter-units are usually related in certain ways. Exploring the relationships is critical in finding out all truthful alter-units.

4.1 Relationships among Alter-units

Alter-units can be concepts (categories) themselves or instances of some concepts. For example, the truthful alter-units for the first example in Table 2 are the instances of concepts City, State and Country, respectively. As another example, for doubtful statement "Duke Ellington is a [composer]", both "composer" and "musician" are truthful alter-units and they are both concepts. Based on our analysis of many examples, we found that the following relationships among alter-units are most useful for inferring new truthful alter-units from known truthful alter-units:

- *Synonym Relationship*: Clearly, for two alter-units that are synonyms, if one of them is truthful, then the other should also be truthful.
- *Is_a Aelationship*: This relationship usually occurs between two concepts. For example, "composer" has an is_a relationship with "musician".
- *Part_of Relationship*: This relationship may occur between two concepts or between the instances of two concepts. An example of the former: "State" is part of "Country"; an example of the latter is: "Hawaii" is part of "US".
- *Instance_of Relationship*: This relationship indicates whether an alter-unit is an instance of a concept.
- *Correlation Relationship*: This relationship is a measure on how much two alter-units are correlated in terms of their co-occurrences in a dataset. The dataset can be the set of retrieved SRRs or the set of all documents indexed by a search engine. The correlation computed from the former will be called *local correlation* and that from the latter will be called *global correlation*.

The first four types of relationships can generally be obtained from semantic dictionaries (e.g., WordNet[1]), general concept hierarchies (e.g., ODP[2]) and some existing knowledge bases (e.g., Probase[3], 50states.com[4]). In this paper, we assume that these types of relationships can be readily obtained from existing sources.

For the rest of this subsection, we discuss how we compute the correlations between two alter-units AU_1 and AU_2.

- *Global Correlation*

The global correlation of AU_1 and AU_2 is a correlation of AU_1 and AU_2 among the indexed documents of the search engine used. We to compute the global correlation as follows. First, submit AU_1 and AU_2 as two separate queries to the search engine. Let $Hits(AU)$ denote the number of results (hits) returned for alter-unit AU. Next, submit "AU_1 and AU_2" to the search engine to find the number of results that contain both AU_1 and AU_2. Let this number be denoted by $Hits(AU_1, AU_2)$. Finally, use the following formula (a variation of the PMI suggested in (Magnini et al., 2002)) to compute the global correlation of AU_1 and AU_2:

$$G_{corr}(AU_1, AU_2) = \frac{Hits(AU_1, AU_2)^2}{Hits(AU_1) * Hits(AU_2)}. \quad (1)$$

The global correlation computed above is unlikely to be very useful because it does not take the right context of AU_1 and AU_2 into consideration. For example, "Edwin" and "Edmond" may appear in the same document for various reasons. But if we talk about these two names in the context of "Nobel Prize in Medicine", it's much more likely that these two names refer to "Edwin Krebs" and "Edmond Fischer". To increase the likelihood that we compute the global correlation of AU_1 and AU_2 in the right context, we add the topic query keywords into each of the above three queries (they are AU_1, AU_2 and "AU_1 and AU_2") to form three new queries. In these queries, the content words in the topic units and the used alter-units are required terms. Thus, $Hits(AU_1)$, $Hits(AU_2)$ and $Hits(AU_1, AU_2)$ now denote the numbers of results retrieved by the three new queries, respectively.

- *Local Correlation*

The following three sets of SRRs can be used to compute the local correlation for alter-units AU_1 and AU_2: the first set (denoted $P_1(SRR)$) consists of the 200 SRRs that were retrieved by the topic query; the second and the third sets (denoted $P_{21}(SRR)$ and $P_{22}(SRR)$, respectively), each consists of the 100 SRRs that were retrieved by the alternative statement corresponding to each of AU_1 and AU_2, respectively. All these 400 SRRs are in the right context for computing local correlations as they are all retrieved with the topic units as part of the query.

We consider two options to utilize the above SRRs in local correlation computation. The first is to use

[1] http://wordnet.princeton.edu
[2] http://www.dmoz.org
[3] http://research.microsoft.com/en-us/projects/probase/
[4] http://www.50states.com

them together and the second is to use the three sets separately.

(a) Using all SRRs Together. Let *RS* denote the 400 SRRs. Rather than simply using Equation 1 to compute the local correlation by treating the SRRs in *RS* as the set of documents indexed by the search engine, in our work, we extend this basic formula by also taking into consideration the proximity between AU_1 and AU_2 in each SRR. The basic idea is that when AU_1 and AU_2 appear closer in an SRR, they contribute more to the correlation. Specifically, when computing the proximity score of AU_1 and AU_2 within an SRR r, denoted as $Prox(r,AU_1,AU_2)$, the following three cases are considered:

- If one of AU_1 and AU_2 does not appear in r, $Prox(r,AU_1,AU_2) = 0$.
- If one of AU_1 and AU_2 only appears in the title of r and the other only appears in the snippet of r, $Prox(r,AU_1,AU_2) = \alpha$, where $\alpha > 0$ is used to give a minimum proximity score when AU_1 and AU_2 appear in the same SRR. The value of α is determined empirically.
- If AU_1 and AU_2 both appear in the title or both appear in the snippet of r, the proximity score is computed by:

$$Prox(r,AU_1,AU_2) = max \{\beta, 1 - min(\frac{dist_t(AU_1,AU_2)}{len(r_t)-2}, \frac{dist_s(AU_1,AU_2)}{len(r_s)-2})\}. \quad (2)$$

where $dist_t(AU_1,AU_2)$ and $dist_s(AU_1,AU_2)$ are the numbers of words between two closest appearances of AU_1 and AU_2 in the title and snippet of r, respectively; $len(r_t)$ and $len(r_s)$ are the numbers of words in the title and snippet of r, respectively. If AU_1 and AU_2 do not both appear in title, then $dist_t(AU_1,AU_2) = len(r_t)$; if AU_1 and AU_2 do not both appear in snippet, then $dist_s(AU_1,AU_2) = len(r_s)$. $\beta > \alpha$ is used to guarantee that in Case 3 the proximity score will be higher than that in Case 2. The value of β is also determined empirically. In our current work, $\beta = 1.5\alpha$ is used.

We now introduce our formula for computing the local correlation of AU_1 and AU_2. Let $C(r,AU)$ be a sign function indicating whether alter-unit AU is contained in SRR r. That is, $C(r,AU) = 1$ if AU is contained in r and $C(r,AU) = 0$ otherwise. The local correlation of AU_1 and AU_2 is computed:

$$L_{corr}(r,AU_1,AU_2) = \frac{(\sum_{r\in RS} C(r,AU_1) * C(r,AU_2) * Prox(r,AU_1,AU_2))^2}{(\sum_{r\in RS} C(r,AU_1)) * (\sum_{r\in RS} C(r,AU_2))} \quad (3)$$

Note that $\sum_{r\in RS} C(r,AU_1)$ is the number of SRRs in *RS* that contain AU_1. The value of the correlation varies from 0 (when two alter-units never co-occur in any retrieved SRRs) to 1 (when the two alter-units always co-occur and appear next to each other either in the title or in the snippet).

(b) Using $P_1(SRR)$, $P_{21}(SRR)$ and $P_{22}(SRR)$ Separately. By substituting *RS* in Option (a) by each of the three sets, a different local correlation can be computed. We take the maximum of the three local correlations as the final local correlation in this case. The reason we consider this case is that the three sets of SRRs have different characteristics. $P_1(SRR)$ is retrieved using the topic units only and these SRRs are not specifically targeting AU_1 or AU_2. The SRRs in $P_{21}(SRR)$ ($P_{22}(SRR)$, respectively) are all related to AU_1 (AU_2) and the corresponding correlation essentially indicates (if proximity information is not considered) what percentage of the SRRs that contain AU_1 (AU_2) also contain AU_2 (AU_1).

- *Combined Correlation*

There are different possible ways to combine/aggregate local correlation and global correlation. In this paper, we combine them using the maximum function because it performed well in our preliminary test (not reported here).

$$Comb_{corr}(r,AU_1,AU_2) = max\{G_{corr}(AU_1,AU_2), L_{corr}(AU_1,AU_2)\}$$

4.2 Inference Rules among Alter-units

When two alter-units have certain relationship, it may become possible to infer the truthfulness of one alter-unit from that of another. For example, if "Honolulu" and "Hawaii" are both alter-units for "Kenya" in "Barack Obama was born in [Kenya]", then knowing "Honolulu" is truthful and "Honolulu" is part of "Hawaii", we can infer that "Hawaii" is also truthful. On the other hand, if we already know that "Hawaii" is untruthful, we can infer that "Honolulu" is also untruthful.

In our work, we employ a number of truthfulness inference rules. We divide these rules into four categories: *synonym* rule, *instanceOf* rules, *partOf* rules and *correlation* rules. To enable the *instanceOf* rules, we assume that an *IS_A* Concept Hierarchy (ICH) is available. To enable the *partOf* rules, we assume that a *Part_Of* Concept Hierarchy (PCH) is available.

Due to space limitation, we will not present the details of these rules in this paper. Instead, we use two matrices CC-matrix and CO-matrix to summarize the inference relationships among all extracted alter-units $L_{AU} = (AU_1, AU_2, \ldots, AU_n)$. The CC-matrix represents the synonyms, instanceOf and partOf rules and

CO-matrix represents the correlation rule. For a pair of alter-units (AU_i, AU_j), we define its corresponding value in CC-matrix (CC-matrix[i, j]) as the probability that AU_j can be inferred from AU_i. The value in CC-matrix is defined as follows:

1. If AU_i and AU_j are synonyms, then one's truthfulness can be directly inferred from the truthfulness of the other. In this case, CC-matrix[i, j] = CC-matrix[j, i] = 1.0.
2. If AU_i is a descendent of concept AU_j in ICH or AU_i is a part of AU_j in PCH, CC-matrix[i, j] = 1.0 according to the instanceOf or partOf generation rules.
3. If AU_i is the ancestor of concept AU_j in ICH or AU_i includes AU_j as a part according to PCH, CC-matrix[i, j] = $1/N$, where N is the number of alter-units as children of AU_i.
4. For all other situations, CC-matrix[i, j] = 0.

The values of CO-matrix entries are defined as the combined correlation of each pair of alter-units, i.e. CO-matrix[i, j] = $Comb_{corr}(AU_i, AU_j)$. A higher correlation between AU_i and AU_j indicates a higher probability that the alter-units have the same truthfulness.

In next section, we propose three algorithms exploring the inference rules in different ways.

5 IDENTIFY MULTIPLE TRUTHFUL ALTERNATIVE STATEMENTS

In this section, we present three algorithms for truthful alternative statements identification.

5.1 Top Alter-unit Expansion (TAE)

For any given doubtful statement, we implement the method in (Li et al., 2011) to rank all alternative statements and recognize the top-ranked alternative statement as truthful. Our TAE algorithm selects the alter-unit of the top-ranked alternative statement as the seed truthful alter-unit and tries to identify other truthful alter-units from it.

The basic idea of the inference process of the TAE algorithm is as follows. Let T_{AU} denote the set of computed truthful alter-units for the given doubtful statement. Let AU_{top} denote the alter-unit of the top-ranked alternative statement produced by T-verifier. Initially AU_{top} is the only alter-unit in T_{AU}. For every alter-unit not in T_{AU}, we check if it can be inferred from the alter-units in T_{AU} using the inference rules introduced in Section 4.2. If the result is positive, add this alter-unit to T_{AU}. This process is repeated until no alter-unit can be added to T_{AU}. All alter-units in the final T_{AU} are considered to be truthful.

In Section 4.2, we introduced CC-matrix and CO-matrix to represent the inference rules and probabilities. Two different probability thresholds are used in our current implementation, θ_1 for the CC-matrix and θ_2 for the CO-matrix, to determine if the truthfulness of one alter-unit can be inferred from another.

TAE algorithm is easy to implement. The main limitation of this algorithm is that it is highly dependent on the accuracy of the top-ranked alter-unit. When the top-ranked alter-unit is not actually truthful, the alter-units inferred from it are also unlikely to be truthful. Recall that about 90% of the top-ranked alter-units by the method in (Li et al., 2011) are truthful which essentially makes 90% the upper bound for the accuracy of the TAE.

5.2 Truthfulness Group (TG)

The idea of the TG algorithm is to first divide the set of alter-units of a doubtful statement into multiple groups and then select one group as the truthful group (i.e., all alter-units in this group are recognized as truthful).

Alter-units are grouped based on their compatibility and correlation. Any pair of alter-units in a group should satisfy one of the following conditions: one can be inferred from the other; one is highly correlated with the other. Note that one alter-unit may be included in multiple groups (see the discussion below). The TG algorithm consists of two steps, i.e., *alter-unit grouping* and *group selection*. These two steps are described below.

Alter-unit Grouping. There are three sub-steps. First, form initial groups by putting alter-units that are synonyms together. Second, use the concept hierarchies ICH and PCH to expand each initial group. Specifically, find all alter-units that are not ancestors of any other alter-units, treat each of them, say AU^*, together with its synonyms, as a group, denoted as G^* (it is one of the initial groups), and add each alter-unit that is an ancestor of any alter-unit in G^* to G^*. Note that it is possible for the same alter-unit to be added to multiple groups in this step. For example, if an alter-unit is an ancestor of two alter-units in different initial groups, this alter-unit will be added to both of these two groups. Third, apply agglomerative clustering to the groups from the second sub-step and merge two groups at a time using correlation information. There are several ways to define the correlation between two groups G_1 and G_2. The following three methods are considered in this paper.

1. *Alter-unit based*: Define the group correlation as the largest combined correlation between each pair of alter-units, one from each group. As we have mentioned above, some alter-units may appear in multiple groups. When computing the correlation between two groups, we do not consider any pair of alter-units that are in fact the same alter-unit appearing in the two groups (this pair will always has correlation 1.0).
2. *Group based*: Conceptually treat the alter-units in each group as a single virtual alter-unit such that an occurrence of any of the alter-units in the group is counted as an occurrence of the virtual alter-unit. Then we define the group correlation as the combined correlation of the two virtual alter-units for G_1 and G_2.
3. *Synonym based*: For each group, treat the alter-units that are synonyms as a virtual alter-unit. Then apply the method (1) to find the largest combined correlation between each pair of alter-units (including virtual alter-units), one from each group.

During the agglomerative clustering process, each time the two groups with the highest correlation are considered for merging. The merging process stops when the highest correlation between any two of the remaining groups does not exceed a pre-set threshold T_{TG}, which is determined empirically. The three methods for computing group correlation will be compared in Section 6.

Group Selection. For each group, we add the ranking scores of the alternative statements with alter-units in the group and treat the sum as the ranking score of the group. The group with the largest score is selected as the truthful group.

Algorithm 1: TG algorithm.

```
Input  : CC-matrix, CO-matrix, L_AU
Output : TopG
Groups ← {}
foreach AU_i ∈ L_AU do
    foreach AU_j ∈ L_AU do
        if AU_i, AU_j are synonyms then
            AU_i^* ← merge(AU_i, AU_j)
Update L_AU with AU_i^*
foreach AU_i^* ∈ L_AU do
    if AU_i^* has no descendants then
        G^* ← {AU_i^*}; Groups ← Groups ∪ {G^*}
AggrCluster(Groups, T_TG)
foreach G^* ∈ Groups do
    SG(G^*) ←calGroupScore(G^*)
TopG← argmax{SG(G^*)}
```

5.3 Truthfulness Propagation (TP)

TG performs one-step inference based on the initial ranking score of each alternative statement. Compared to the static nature of the TG, the TP algorithm employs dynamic truthfulness propagation among the alter-units. It treats each alter-unit as a node and the inference probability and/or correlation score between each pair of alter-units as the link connecting the nodes. The truthfulness is then propagated among the nodes along the links in a process similar to the computation of PageRank (Brin and Page, 1998)

Suppose the vector $\vec{r}^{(0)} = (s_1, s_2, \ldots, s_n)$ is a list of scores assigned to the alternative statements. In our case, s_i is the score of alternative statement that contains AU_i and the score was obtained from the statement verification algorithm in (Li et al., 2011). $\mathcal{T}$ is the truthfulness propagation matrix built up based on CO-matrix and CC-Matrix. Specifically, $\mathcal{T}[i,j] = max$(CO-matrix[i, j], CC-matrix[i, j]). The ranking scores of all alter-units after the k-th iteration is:

$$\vec{r}^{k} = \gamma * T * \vec{r}^{k-1} + (1-\gamma) * \frac{1}{n} * I_n \quad (4)$$

where n is the total number of alter-units extracted for the doubtful statement, I_n is the unit vector and γ is a weight parameter. Iteration continues until the ranking scores converge. At the end of final iteration, all the alter-units with the final scores are ranked in descending order. Finally, we use the best-performed Top-k algorithm FSG (see Section 2) to select the set of truthful alter-units.

6 EXPERIMENTS

6.1 Dataset

Our dataset consists of 50 doubtful statements. Each doubtful statement has a specified doubt unit and there are two or more truthful alter-units for each doubt unit (see Section 2). Half of the 50 doubtful statements have truthful doubt units and the other half have untruthful doubt units. 25 of the 50 statements are of the CC type (i.e., Type 1) and 25 of the MVA type (i.e., Type 2). These doubtful statements are manually converted from questions with multiple answers in the QA track of TREC-8, TREC-9 and TREC 2001. The conversion is done by re-writing each question to a statement and replacing the WH-word (e.g., who, where, etc.) in the question with one of the provided answers (for the 25 statements with truthful doubt units) or with a term of the same type as

Table 3: Empirical parameters.

Proximity $\cdot \alpha$	0.2	TAE $\cdot \theta_1$	0.9	TP $\cdot \gamma$	0.15
Proximity $\cdot \beta$	0.3	TAE $\cdot \theta_2$	0.12	TG $\cdot T_{TG}$	0.16

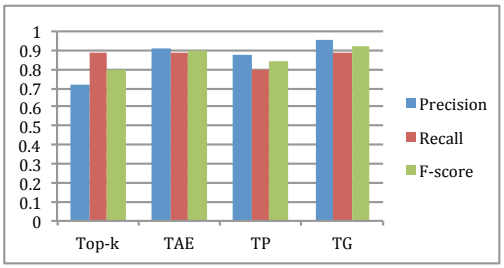

Figure 2: Performance Comparison of Four Algorithms.

the answer(s) that makes the statement untruthful (for the 25 statements with untruthful doubt units). The term that replaces the WH-word will be specified as the doubt unit.

6.2 Performance of Algorithms

We use overall recall (r), precision (p) and F-score as performance measures. For a set of doubtful statements, if the total number of truthful answers is N, the total number of truthful answers recognized by an algorithm is N_r and the total number of answers generated by the algorithm is N_g, then $r = N_r/N$, $p = N_r/N_g$ and F-score $= (2 * r * p)/(r + p)$.

Overall, we proposed four algorithms (i.e. Top-k, TAE, TP and TG) to identify the truthful alternative statements. Each algorithm has several variations, like using different correlation matrices or applying different Top-k selection options. Later, we will analyze how these variations affect the performance. In order to set up the parameters in each algorithm empirically, we randomly select 25 of the 50 statements as training set and the rest as testing set and show the performance over training set and testing set separately in Table 4.

In Fig. 2, we show the best performance results achieved by each of the four main algorithms over all of the 50 doubtful statements in our dataset. From the results, we can see that the TAE, TP and TG algorithms all have significantly improved performance over the Top-k FSG algorithm. Note that the TG algorithm achieves the best F-score, which reaches about 0.92, followed by TAE at 0.90 and TP at 0.84. Algorithm TP performs significantly worse than TAE and TG. The main reason is that this method does not form a truthful group and it still uses a Top-k method to select the final truthful alter-units. Table 3 lists all the parameter values used in our experiments.

We also tested the proposed algorithms to find out whether these algorithms have different performances on different types of statements. Recall that our dataset has 25 Type 1 (i.e., CC) statements and 25 Type 2 (i.e., MVA) statements. Overall, all the algorithms perform better on Type 1 statements than on Type 2 statements (see Fig. 3). This is probably due to the fact that the truthful alter-units of Type 1 statements have stronger relationships than those of Type 2 statements. Specifically, the relationships via the is_a and part_of hierarchies are usually stronger and more definitive than the correlation relationships. In general, all inference rules introduced in Section 4 are applicable to the alter-units of Type 1 statements while for the alter-units of Type 2 statements only the correlation rules are generally applicable.

It is notable that for Type 1 statements both the precision and recall of algorithms TG and TAE are above 90% and both perform significantly better than algorithm TP. For Type 2 statements, algorithm TG outperforms others with F-score of 0.88, followed by TAE with F-score of 0.85 and TP of 0.84.

In general, algorithm TG performs the best among all the four algorithms. The erroneous cases fall into two categories: (1) Untruthful but relevant alter-units are involved because they are highly correlated with the truthful ones. Like the first example in Table 5, “Jason Lochinvar” is the name of the person “William Conrad” played in the movie, which frequently co-occurs with the truthful alter-unit “William Conrad”. (2) Insufficient correlation results in missed truthful alter-units. For the second example in Table 5, we missed the truthful “Ned Rocknroll” who is the husband with Kate Winslet. Among the three alter-units “Ned Rocknroll”, “Jim Threapleton” and “Sam Mendes” (the latter two are Kate Winslet’s ex-spouse), “Jim Threapleton” and “Sam Mendes” are often mentioned together so they are placed in the same group due to their high correlation. But “Ned Rocknroll” does not have high enough correlation with this group. As a result, ”Ned Rocknroll” is not merged into this group. In the end, this group has a higher overall ranking score than the score of “Ned Rocknroll”.

6.3 Effects of Different Correlations

In Section 4.1, we introduced four different ways to calculate the correlations between a pair of alter-units, including the global correlation, the local correlation using all SRRs, the local correlation that takes the maximum of three correlations computed using three sets of SRRs (i.e., P_1(SRR), P_{21}(SRR) and P_{22}(SRR)), and the combined correlation. We conducted experiments to find out how each of these correlations per-

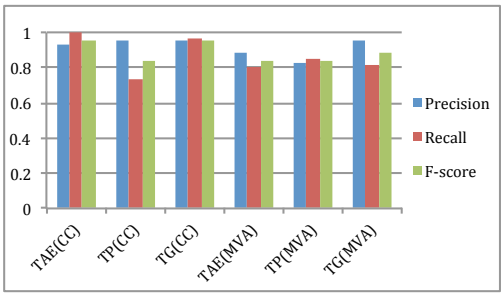

Figure 3: Performance Comparison on CC & MVA Statements.

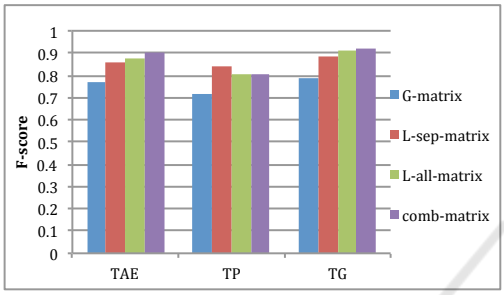

Figure 4: Comparison of Four Different Correlations.

forms when used in different algorithms. From the results in Fig. 4, we can see that the global correlation is the least effective and the other three correlations have similar performance with the combined correlation having a small overall edge over the two local correlations. These results suggest that the global correlation computed based on the numbers of hits of the formed queries is not very reliable. In contrast, the local correlations that take into consideration the proximity of the two alter-units are quite effective.

6.4 Variations of TG Algorithm

In Section 5.2, we presented three different methods to compute the correlation between two groups: *alter-unit based*, *group based* and *synonym based*. From the results in Fig. 5, the synonym based method improves the precision from 0.93 to 0.95, resulting in an increased F-score from 0.91 to 0.92. In comparison, the group based method reduces the F-score to 0.88

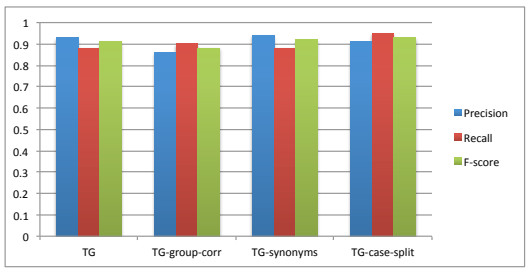

Figure 5: Comparison of Variations of TG Algorithm.

Table 4: Performance on training and testing sets.

	Training set			*Testing set*		
	Prec	Recall	$F1$	Prec	Recall	$F1$
TAE	0.92	0.89	0.91	0.89	0.86	0.87
TP	0.90	0.82	0.86	0.86	0.77	0.81
TG	0.95	0.91	0.93	0.90	0.88	0.89

Table 5: Erroneous examples.

Doubtful Statement	Answers Expected	Answers Found
[William Conrad] starred in "Jake and the Fatman?"	"William Conrad" "Joe Penny", "Alan Campbell"	"William Conrad", "Joe Penny", "Jason Lochinvar", "Alan Campbell"
Kate Winslet married to [Tom Cruise]	"Ned Rocknroll", "Jim Threapleton", "Sam Mendes"	"Jim Threapleton", "Sam Mendes"

in spite of a slight increase on the recall.

Besides the above three variations of the TG algorithm, we do an extra experiment to evaluate how we shall benefit if we know the type of a statement in advance. We select one agglomerative clustering threshold for all the Type 1 statements (0.19 in our experiment) and one for the Type 2 statements (0.09 in our experiment) and apply the TG algorithm with the synonym based method. From the results (see the last group of results in Fig. 5) we can see that using separate thresholds boosts the F-score from 0.92 to 0.94 by significantly increasing the recall with a small sacrifice on precision.

7 RELATED WORK

Two lines of research are related to our work.

- *Verification of Fact Statements*

Honto?Search (Yamamoto et al., 2007; Yamamoto et al., 2008) and T-verifier (Li et al., 2011) are works focusing specifically on the verification of the truthfulness of fact statements. They aim to find just one correct answer for each doubtful statement. In contrast, our work focuses on doubtful statements that have multiple correct answers and aims to find all of these answers. None of the algorithms introduced here was discussed in the above papers.

- *Question-answering Systems*

Question-Answering systems have been an active research area in information retrieval and NLP communities for many years. The goal is to develop techniques that can answer natural language questions from a text corpus (see (Prager, 2006) for survey).The TREC conference series ran a QA Track from 1999 to 2007. In TREC 2001, a list task was started. A question in a list track has multiple truthful

answers (Voorhees, 2001). IBM's QA system Watson (Ferrucci et al., 2010) is a state-of-the-art QA system with many advanced features. But Watson does not deal with questions with multiple answers. Furthermore, most QA systems, including Watson, use pre-collected text corpus, not the open Web as in our approach.

Techniques for answering list questions in QA systems are relevant to our work. In (Wang et al., 2008), the authors proposed a method to expand a set of answers from selected answer seeds. The idea of this method is similar to our TAE algorithm except their expansion only depends on the global correlation of two candidate answers. According to our experimental results in Section 6.3, global correlation turns out to be the least effective among several types of correlations we evaluated. Specifically, local correlation and combined correlation are significantly better for performing truthful alter-units (answers) expansion. In (Jijkoun et al., 2007), answers are clustered based on their similarity and all answers in the same cluster are treated as one unit in the answer's ranking process. Similar idea is also found in (Ko et al., 2007) except they extend the similarity computation from string distance metrics to exploring semantics similarity based on WordNet, Wikipedia, etc. Essentially, their solution accepts multiple answers being "synonyms" to each other, which is one of our inference rules. The work in (Razmara, 2008) is most relevant to our TG algorithm. Both methods perform clustering on candidate answers (alter-units) based on correlations among them. But they also have several significant differences. First, different correlations are used. We use a combined correlation and the method in (Razmara, 2008) uses correlation based on sentences extracted from some documents (no global correlation, no proximity information and no SRRs are used). Second, the clustering process is also different. Our method has three sub-steps and the best option for correlation computation (i.e., synonym-based) is not used in the method (Razmara, 2008). Finally and very important, we would like to emphasize that the fact statements we consider and the questions QA systems consider are very different concepts. The main difference is the information about the doubt unit. Each fact statement we consider has an instance of the doubt unit while questions in QA systems have only type information about the doubt unit (e.g., from a question starting with "Where", it can be easily inferred that the type of the doubt unit is Location). An instance has significantly more information than a type. We can usually infer a more precise type from the instance. For example, from "New York City" we can infer a type City which is more specific than Location. Furthermore, the instance itself provides valuable information as it may be used to find clues (via different relationships such as correlation relationships) for truthful alter-units. Our approach takes advantage of this difference.

8 CONCLUSION

In this paper, we investigated the very challenging problem of processing doubtful fact statements that have multiple alternative answers for a specified doubt unit. The goal is to find all truthful answers for such doubtful statements. We first evaluated a Top-k solution and showed that none of the variations of this solution is sufficiently accurate. We presented solutions for two types of MTA statements (compatible concepts and multi-valued attributes). Our solutions explored some fundamental relationships among truthful alter-units such as synonym, is_a, part_of and co-occurrence correlation relationships. Based on different ways in which the above relationships are utilized, we proposed three algorithms (TAE, TP and TG) for selecting the truthful alter-units. We carefully evaluated the effectiveness of different algorithms and different types of correlations on different types (CC and MVA) of MTA statements. Our experimental results indicate that the TG algorithm is the most effective overall with F-score around 90%.

ACKNOWLEDGEMENT

This work was supported in part by the following NSF grants: IIS-1546441 and CNS-0958501. This work was partially done when the first two authors visited SA Center for Big Data Research hosted in Renmin University of China. This Center is partially funded by a Chinese National "111" Project "Attracting International Talents in Data Engineering and Knowledge Engineering Research".

REFERENCES

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Ferrucci, D. A., Brown, E. W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J. M., Schlaefer, N., and Welty, C. A. (2010). Building watson: An overview of the deepqa project. volume 31, pages 59–79.

Jijkoun, V., Hofmann, K., Ahn, D., Khalid, M. A., van Rantwijk, J., de Rijke, M., and Sang, E. F. T. K. (2007). The university of amsterdam's question answering system at qa@clef 2007. In *CLEF*, pages 344–351.

Ko, J., Si, L., and Nyberg, E. (2007). A probabilistic framework for answer selection in question answering. In *Proceedings of HLT-NAACL*, pages 524–531.

Li, X., Meng, W., and Yu, C. (2011). T-verifier: Verifying truthfulness of fact statements. In *Proc. of ICDE*, pages 63–74.

Magnini, B., Negri, M., Prevete, R., and Tanev, H. (2002). Is it the right answer? exploiting web redundancy for answer validation. In *Proc. of ACL*, pages 425–432.

Prager, J. M. (2006). Open-domain question-answering. *Foundations and Trends in Information Retrieval*, 1(2):91–231.

Razmara, M. (2008). Answering list and other questions.

Voorhees, E. M. (2001). Overview of the TREC 2001 question answering track. In *Proceedings of The Tenth Text REtrieval Conference*.

Wang, R. C., Schlaefer, N., Cohen, W. W., and Nyberg, E. (2008). Automatic set expansion for list question answering. In *Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 947–954.

Yamamoto, Y., Tezuka, T., Jatowt, A., and Tanaka, K. (2007). Honto? search: Estimating trustworthiness of web information by search results aggregation and temporal analysis. In *APWeb & WAIM*, pages 253–264.

Yamamoto, Y., Tezuka, T., Jatowt, A., and Tanaka, K. (2008). Supporting judgment of fact trustworthiness considering temporal and sentimental aspects. In *WISE*, pages 206–220.