CHAPTER 6

WEB SEARCH RE-RANKING METHOD

6.1 INTRODUCTION

The need for information retrieval is for the results of the search process to be personalized, context based and closest to what the user wants. In this work, there are two modes, where the results are re-ranked: based on the trust ratings of the content, and based on the ontological processing. This chapter discusses these two methods of information retrieval: Trust based and Ontology based. A new trust based system oriented towards mobile devices is proposed. The trust rating method relies on profile mining and the past experiences of the users. The ontological method models the set of keywords retrieved by the search process as a unified whole, from where the re-ranking of the content can be done using the fuzzy relations between the query term and the ontology.

6.2 TRUST-BASED MODEL

The objective is to calculate the trust value of a website retrieved in the snippet stream. For this a combination of the content in the snippet and the context is taken. The trust-based model is implemented in the personal context mode.

The first step is when the user has not seen the websites already and, hence how to compute the trust values. Is there a need for a new trust-
model or can the probabilistic trust model give the answer? The answer is, there is a need for classifying the unseen content into two categories:

- It is not seen and unrelated to the trusted content in the ontology. Here, the probabilistic model will rely on the type of the word, and user-provided information to characterize the content. The word will be randomly indexed with a trust value. The idea here is that, while the trust rating may not be so accurate at this instance, the indexing of the content ensures that the query will be handled properly on the next possible occasion.

- It is not seen, but related in some way to the content in the ontology: The answer in this case will be to rank the contents, reuse the trust model, and the related trust contents to handle this scenario.

The user may, at some point of time, type a word or phrase that has already been used in the past. There are two outcomes that may be needed then:

- The website selected can be shown to the user, thereby simplifying the entire process of selection. In other words, if the user searches for a particular term, say, ‘Anna University’ and has selected ‘annauniv.edu’ in the past, the same website address can be suggested. An interesting solution here is, to show the website address already seen and selected previously by the user, and also show other relevant websites. This may be because the user wants to explore all choices. The exploration will account for the provision of new content in
the website already seen and not selected, and thus be fair. This is not easy as the user may potentially use hundreds of websites, and so storing and managing all the content is a non-trivial task. This is a word-site addressing issue. The solution needed here is for the users to give trust ratings for the content obtain from websites. These trust ratings can be generic for a website or specific to a single word. In other words, the users can associate a particular website to a particular word and declare that the site is the best for the word, or associate one website as trusted for all the words. For example, the user can trust nokia.com for the term ‘mobile phone’. Also, the user can trust Wikipedia.com for all terms. These two options are implemented in this work.

- The user may have seen a website already for a word, and not selected it. Now, can insights be drawn from this? Also, the content type searched for, needs to be considered. In the case of a noun or noun forms, the users typically may look for specific addresses, and hence, a word-site addressing approach may work. For example, for a search of the ‘BSA University’, the user is typically looking for the content relating to the University’s website. (This conclusion is drawn from an analysis of the data that is collected as a part of this work). For the non-noun form content, the options vary depending on the timing. There is a difference of around 40% in the content, when the same query words are typed. There is a need for a trust based probabilistic model here, that takes into account the type of the query, past history and the trust-relationships in the ontology. This model is proposed in this work.
The overall idea is that the content retrieved consists of two kinds of content sources: Trusted and Un-seen. The trusted content sources fall into the category where the user has seen the content source, and trusts it for a word or for everything. The un-seen content sources are all categories, which do not fall under the trusted sources. The need of the hour is thus a model of trust validation. The algorithm for trust rating works as follows.

6.2.1 Mathematical Modeling

A Query is entered by the user. All queries for which trust values are explicitly specified by the user, are called patterns. All queries which are present in the ontology are called Words.

The user’s queries are denoted as

- Queries \( Q_i \): \( Q_1, Q_2, Q_3, \ldots \)
- Patterns \( P_i \): \( P_1, P_2, P_3, \ldots \)
- Words \( W_i \): \( W_1, W_2, W_3, \ldots \)

The content sources are categorized into three categories:

- Paired sources: \( P_i \): \( P_1, P_2, P_3, \ldots \)
- Trusted sources: \( T_i \): \( T_1, T_2, \ldots \)
- Unseen sources: \( U_i \): \( U_1, U_2, U_3, \ldots \)

Paired sources are those words for which the users specify a website value and call it trusted. The Trusted sources are universally trusted content websites. The Unseen sources are the other websites that occur in the processing.
The Trust values $V_i$: $V_1$, $V_2$, $V_3$... are computed dynamically and are based on the contents of

- Content $C_i$: $C_1$, $C_2$, $C_3$, ...
- Profile: $P_i$: $P_1$, $P_2$, $P_3$...
- Explicit Context $E_i$: $E_1$, $E_2$, $E_3$...

The three trust calculation procedures are

- NTrust: Unseen values
- WTrust: Words
- PTrust: Patterns

are discussed next.

**Theorem 1:**

For all the values in $Q_i$, there exists a unique identifier $V_i$ such that the trust value is $V_i$: $NTrust (Q_i) = NTrust ((C_i, P_i, E_i) \times Q_i \times (P_i \mid T_i \mid U_i)) \rightarrow n$ where $n \geq 0$.

The meaning of this theorem is that a unique trust value can be calculated for all queries, irrespective of whether the query has been seen already or not. The only essential and sufficient condition is that the personal mode emphasizes the creation of a link to at least one existing term in the ontological base. The condition here is that the query’s meaning must be known by the system. Based on the meaning, the query value is calculated as a product of the ontology and the existing contents.
There are two possible scenarios here: a) the entire trust value is calculated for the first time in the system, b) the value is calculated for the first time for the query, but the related trust values in the ontology exist. These scenarios are handled by the N Trust Algorithm.

**Theorem 2:**

For all the values in I, there exists a unique identifier $W_i \subset Q_i$ such the trust value is $V_i$: $WTrust (Q_i) = WTrust (W_i \times Ntrust (Q_i)) \rightarrow n$ where $n \geq 0$.

Conversely, the reverse case is not applicable.

The meaning of this theorem is that a website can be linked to more than one word. But, one word cannot have more than one website. This is to maintain the integrity values and keep the system manageable. The implications are in the specification of the trust values by the users. If the user tries to assign more than one website for a single word, then theorem 3 follows.

The trust value calculation is a two step process. First, the relationship of the term with the contents of the ontology, personal profile, etc. is found as before. Then, the trust calculation takes into account the fact that the word is associated by the user with a website and the resultant relationship is framed. Based on the cross product of the two values, the overall trust value is found.
Theorem 3:

For all the values in $P_i$, there exists a unique identifier $Q_i$ such that the trust value is $V_i$:

$$P_{Trust}(Q_i) = P_{Trust}(P_i \times Q_i \times P_{Trust}(V_i)) \Rightarrow n,$$

where $n \geq 0$.

The meaning of this theorem is that for all the trusted contents the initial pairing or declaration must have followed from a single Query term $Q_i$. Many such query terms can declare a single website and such a site is upgraded to the trusted source. The implication of the theorem is that it allows a website trusted more than once by the user to be changed to a universal content provider. Conversely, the user can downgrade any content provider to a lower stage, based on an explicit menu driven operation.

The trust value calculation depends on the contextual relationships as before, and the overall content provider based trust. An additional wrinkle here is that the trust calculation takes into account the past trust values of the content provider too. This is needed as the content providers can burgeon over time, and hence, the past values need to be adjusted.

For all values $V_i$, the order of display of snippets $S_i$ is determined by the trust values of the content.

$$S_i: Rank(N_{Trust}(V_i), W_{Trust}(V_i) \text{ and } P_{Trust}(V_i))$$

6.2.2 Trust Value calculation

Scenario 1: The Unique Trust Value calculation

From theorem 1 it is clear that the unique trust value is dependent on
• Query
• Content
• Personalized content
• Explicit context
• Linkages of the Query with the content of the ontology

The entire trust value is calculated for the first time in the system. For this case, the trust value is calculated as an outcome of the query and relationship of the content, and is a value between 0 and 1. Prioritized Weightage is given for the

• Personalized content (0.25)
• Explicit content (0.25)
• Content
  ▪ Connector terms (0.2)
  ▪ meaning (0.2)
  ▪ relevant (0.15)
  ▪ related terms (0.1)

In case, no Trust values or related contents exist, the relation for a term is initialized with a value 0.05.

In case the value for a website is not seen, but, the related content values exist, the Trust value is calculated as a weighted product of the relations. Now, this trust value of the website is stored in the Ontology. If the snippet website is selected by the user, a multiplication factor of 0.1 is weighted to the website, till the website reaches 0.9. For example if the calculated trust value is 0.45 and it is selected by the user, the weightage is increased by 0.1, now reaching 0.55.
For example, if the query is vegetarian, and the outcome is the stop word removed snippet shown in Figure 6.1.

\[ S_2: \text{fruit food ingredients natural food.} \]

**Figure 6.1 Stopword removed sample snippet**

Now the trust value is calculated in terms of the initialization parameter 0.05, as no related content exists in the Ontology.

Now the trust value is calculated as

\[
(0.1 + 0.1 + 0.05 + 0.2)/4 = 0.11
\]

**Scenario 2: Relative Trust Value calculation**

For the relative trust value calculation and universal trust value calculation, the weightage follows the principle, that despite the fact that the domain is trusted, the contents retrieved in the snippets must also be related to the query on hand. If the contents retrieved from the snippet are not related to the query closely, the weightage is not increased otherwise it is increased and the values get weighted.

**6.2.2.1 Relative Trust**

- First the N’Trust \((Q_i | O_i)\) is calculated.
- \(N’ \text{Trust} (Q_i) = (N’ \text{Trust} (Q_i) + N’ \text{Trust} (Q_i | O_i))\)
  - If this value is >0.15, then the weightage of the website is increased by a value 0.05 till it reaches the universal trust
value 1. Now, the user is asked if the website can be trusted fully. If not, the trust rating is not incremented at all beyond 1.0.

- If the value of N'Trust is <0.15, then the weightage of the website is decremented by 0.05. This is done till the value reaches 0.20.

Overall, if the user selects the website, then the user is asked if the website can be trusted for the query too. If the user says yes, the relationships are established. The user can also say no. In that case, the website will have only an incremental operation.

### 6.2.2.2 Universal Trust

For all websites that are universally trusted, the N’TTrust (Q|O_i) is calculated. This is to rank more than one universally trusted content site.

Now, the overall ranking is in terms of the

- Universal trust
- Relative trust
- N’TTrust (remaining)

### 6.3 ONTOLOGY BASED MODEL

In the Web search, it has been observed that the majority of the snippet contents contained the search query. Hence, methods to manipulate the results based on the snippets, must also take into account the linkages of the search term in the context of the snippet, thus needing the ontology. The snippets are assigned a rank based on the inter-term relationships in an
organized set of steps. The Dependency tree approach outlined in Chapter 5 is combined with the Term Frequency approach and used.

Each term processing is considered as a step in the computation of the information gain, and the consolidated information gain $t_{f_{ij}}$ is calculated for the entire snippet contents. Here, the notation $t_{f_{ij}}$ represents a term ‘$t$’ in the snippet ‘$f$’. The term ‘$i$’ stands for the snippet value and the term ‘$j$’ stands for the term in the snippet. Each snippet is randomly chosen from among the search results. The terms visited in each snippet can be written as $t_{f_{i1}}, t_{f_{i2}}, t_{f_{i3}}$...

For each term in the snippet, the distance vector measure is calculated in terms of the term-relationship frequency $\alpha$ where the term relationship frequency is calculated as the measure of the term-relationship value level.

Now the term relationship is calculated for the snippet as to how each term is related to the contents of the ontology in the dependency tree order (Figure 6.2). The position in the parse tree is found. For relationships, the value is 0.9. For meanings it is 1.0. The next position (relevant) is 0.75. The third position (related) is 0.60. The next positions are each assigned a value of 0.55, 0.50, 0.45…etc till the 10 terms are reached. For all other terms 0.05 is assigned. Anything beyond is not assigned any value, and left off.
Figure 6.2 Term relationship - Dependency tree

Now $\alpha$ is calculated as the weighted term frequency value of the ontological content. The value $\beta$ is a measure of the distance vector of the snippet to the query word.

Figure 6.3 Distance vector – Dependency Tree

Now, the first level relationships are assigned a value of 1.0. The second levels are assigned a value of 0.9 (Figure 6.3). The third levels are assigned a value of 0.8 and so on. Thus, the distance measure level between the query and the term in the snippet is found.
The trust value of the Web site from which the content is retrieved, is also given proportional importance $\gamma$.

In this case the query is the phone. The query result is shown below in Figure 6.4.

**Mobile phone - Wikipedia, the free encyclopedia**

en.wikipedia.org/wiki/Mobile_phone

A mobile phone allows calls into the public switched telephone system over a radio link ...

The content, after the removal of stop words is

*mobile phone* – allows – calls – public – switched - telephone - system – radio - link

The Ontology-dependency tree structure is shown in Figure 6.5.

![Ontology-Dependency tree structure for term ‘Mobile Phone’](image-url)
For snippet \(\alpha\) is found as \((1.0 + 0.05 + 0.40 + 0.35 + 0.05 + 1.0 + 0.60 + 0.30 + 0.05)/9\).

For the same snippet \(\beta\) is found as \((1.0 + 0.0 + 0.8 + 0.8 + 0.8 + 0.0 + 1.0 + 1.0 + 0)/9\)

The information gain \(tf_{ij}\) is calculated as follows.

\[
tf_{ij} = (\alpha \sum f_i + \beta \sum d_i + \gamma \sum t_i)/\sum tf_{ij}
\]  

(6.1)

After finding the information gain for each snippet, the similarity measure of the snippets is also found. The similarity measure of the snippets \(Sim(tf_i,tf_{i+1})\), is found by the formula

\[
Sim(tf_i,tf_{i+1}) = |tf_i' \cap tf_{i+1}'|/|tf_i' \cup tf_{i+1}'|
\]

(6.2)

where the set \(tf_i'\) is the set of non-stop words of snippet \(tf_i\). Thus, the snippets which are most similar are clustered together and shown to the user as such.

Overall, the ranking for the entire search results is found, and the results re-ranked. The biggest difference between this approach and the term frequency methods lies in the fact, that the entire search results are processed in one single parse operation. Also, the ontological contents of the user’s guide play a role in the re-ranking operation.

6.4 CONCLUSION

In this chapter, the re-ranking algorithm for content organization has been explained in detail. The re-ranking algorithm is snippet based and takes into account the term frequency and the inter-term, frequency. The
foremost limitation in a Mobile environment is its limited screen size; so there is a need to adapt the content retrieved according to the display size. The next chapter focuses on the Uniform Interface model to overcome the limitation of the screen size.