

Query Similarity Computing Based on System Similarity Measurement

Chengzhi Zhang, Xiaoqin Xu, and Xinning Su

Department of Information Management of Nanjing University,
Nanjing 210093, China
zcz51@citiz.net

Abstract. Query similarity computation is one of important factors in the process of query clustering. It has been used widely in the field of information processing. In this paper, a unified model for query similarity computation is presented based on system similarity. The novel approach of similarity computation uses the literal, semantic and statistical relative features of query. The method can take advantage of the normal approaches to improve the computation accuracy. Experiments show that the proposed method is an effective solution to the query similarity computation problem, and it can be generalized to measure the similarity of other components of text, such as sentences, paragraphs etc.

Keywords: query similarity, query clustering, similarity unit, system similarity measuring, literal similarity, semantic similarity.

1 Introduction

In the field of information processing, the similarity computation between strings or queries, such as words, phrases, etc, plays an important part in dictionary compilation, machine translation based on examples, information retrieval, automatic question answering, information filtering and so on. Strings or queries similarity computation is one of important factors in the process of query clustering. It has been used widely in the field of information processing.

This paper builds a unified model to compute the similarity between queries by integrated using the advantages of the three methods, e.g. literal similarity measurement, statistical relevant similarity measurement, semantic similarity measurement, and overcoming their shortcomings. Namely, a unified model of similarity computation is built, which is based on similarity system theory[1] and the measurement of multiple features. It takes similar cell as the queries basic processing unit and considers the similar cell's literal, semantic and statistical relevant features synthetically. At the same time, the model amends the position information missing problem in the processing of sorting similar unit.

2 Related Work

According to the different features, the existing methods of queries similarity computation could be classified into three types: methods based on literal similarity, methods based on statistical relevant similarity, methods based on semantic similarity. Of which, the computation methods based on literal similarity are mainly computed based on edit distance[2] and based on common words or phrases[3]. Methods based on statistical relevant similarity mainly compute words co-occurrence [4], vector space model[5], grammatical analysis[6] and so on. The improved methods based on large-scale corpus such as PMI-IR[7] and various smoothing algorithms[8] is used to resolve the problem of data sparseness in corpus. Methods based on semantic similarity would mainly make use of paraphrase dictionary[9] or some large-scale Ontology[10][11] to do semantic similarity computation.

Method based on literal similarity is simple, and be easy to achieve. But it is not flexible enough and doesn't consider the synonym substitution. Methods based on statistical relevant similarity could get much efficient relevancy between the strings, which could not be observed by people only. But this method depends on the training corpus, and is largely affected by the problem of data sparseness and data noise. Sometimes, methods based on semantic similarity may compute similarity between the strings, which are visual to be literal dissimilarity and statistical to be weak relevancy. But the Ontology are usually built by hand, which need to spend a lot of time.

3 Unified Modeling of Queries Similarity Computation

3.1 Mathematical Description of Queries Similarity Computation

Traditional approaches mostly compute similarity from a certain feature of queries. The similarity computation methods, which combine the literal, semantic and statistical relevant features of queries, have not yet been reflected in any report. Before unified modeling to the similarity computation of queries, we will give several related notes and definitions.

Ω : a set of Chinese strings or queries ;

S : Ω 's subset, that is $S \subseteq \Omega$;

Ψ : semantic dictionary; the authors use it to segment the Chinese text and each listing has a corresponding semantic codes; $\Psi \subset S$;

S_1 、 S_2 : two given queries, including:

$S_1 = \{a_1, a_2, \dots, a_i, \dots, a_M\}$, $i \in [1, M]$, the element quantity of S_1 is M ;

$S_2 = \{b_1, b_2, \dots, b_j, \dots, b_N\}$, $j \in [1, N]$, the element quantity of S_2 is N ;

The element of S_1 、 S_2 could be single character, semantic words segmented by semantic dictionary (or Ontology) or its corresponding semantic codes[11]. Take query ‘计算机控制’ for example, when the element is single character, the query may be expressed as follows: { 计, 算, 机, 控, 制}; if it is segmented by semantic dictionary

(including the words without semantic codes, this paper takes *Tongyici Cilin*[12] as semantic system), the query may be expressed as {Bo010127, Je090101}.

s_i : similar unit; to identify the similar features between S_1 and S_2 , the elements having similar features are known as similar unit. Similar elements which become similar units between S_1 and S_2 , are called similar units, notated as $s(a_i, b_j)$, abridged notated as s_{ij} . Element a_i of S_1 is similar to element b_j of S_2 . Element a_i and b_j are similar elements, which constitute the element cell $s(a_i, b_j)$. According to the similarity priority, we order the character string S_1 and S_2 . And we could get:

$$\begin{aligned} S_1' &= \{a_1, a_2, \dots, a_i, \dots, a_M\}, \\ S_2' &= \{b_1, b_2, \dots, b_j, \dots, b_N\}, \end{aligned}$$

At this point, the similar elements between the strings are a_i and b_j . The similar unit is $s(a_i, b_j)$, abridged notates as s_i .

Definition 1: Quantity of the similar units is the similar degree between element a_i of S_1 and the corresponding element b_j of S_2 , which is notated as $q(s_i)$.

Definition 2: Similarity of the strings is the similar degree between queries S_1 and S_2 , which is notated as $\mathbf{Sim}(S_1, S_2)$.

The common mathematical description of the queries similarity is as follows:

$$\mathbf{Sim}(S_1, S_2) = f(M, N, K, q(s_i)), \quad (i \in [1, K]) \quad (1)$$

Namely, similarity $\mathbf{Sim}(S_1, S_2)$ is a multiple function, whose variables are the quantity of element M in S_1 , the quantity of element N in S_2 , the quantity of similar units K between S_1 and S_2 and $q(s_i)$ which reflected the similar degree between each similar elements.

According to the primary method of similarity measurement between the similar systems in the similarity system theory[1], we should consider two aspects when we do the similarity computation between the queries. That is, the quantity of the similar units and the similar units' numerical value of the similar units. The formula is as follows:

$$\mathbf{Sim}(S_1, S_2) = Q_n \cdot Q_s = \frac{K}{M + N - K} \sum_{i=1}^K \lambda_i q(s_i) \quad (i \in [1, K]) \quad (2)$$

And, λ_i is the weight which reflected the influence degree of similar cell s_i makes to the

strings' similarity, $\lambda_i \in [0, 1]$, $\sum_{i=1}^K \lambda_i = 1$.

Considering the similarity degree by the quantity of the similar units, i.e. Q_n , and the the similarity degree by the similar units' numerical value of the similar units, i.e. Q_s , has a co-complementary function to compute the whole similarity of the similar units, we could assign different weights to Q_n and Q_s , respectively α and β . And that $\alpha, \beta \in [0, 1]$, $\alpha + \beta = 1$. So there is:

$$\mathbf{Sim}(S_1, S_2) = \alpha \cdot Q_n + \beta \cdot Q_s = \alpha \cdot \frac{K}{M + N - K} + \beta \cdot \sum_{i=1}^K \lambda_i q(s_i) \quad (i \in [1, K]) \quad (3)$$

3.2 Improvement of Literal Similarity Computation

If we just consider the literal feature of the queries, namely element a_i of S_1 and element b_j of S_2 are completely literal matching, a_i and b_j could be regarded as similar elements. And the influence degree of similar unit makes to the queries is equal, namely $q(s_i)=1$ and $\lambda_i=1/K$. According to formula (2), we could get:

$$\mathbf{Sim}(S_1, S_2) = \frac{K}{M+N-K} \quad (4)$$

Formula (4) is common and is a simple similarity computation method based on literalness. For example, according to formula (4), the similarity between ‘计算机’ and ‘微机’ is \mathbf{Sim} (‘计算机’, ‘微机’) =0.25. Because formula (4) computes the queries similarity excluding the similar elements’ position information, the computation result would not be reliable. For instance, \mathbf{Sim} (‘计算机’, ‘机计算’)=1.

According to formula (3), set $\alpha=0.6$, $\beta=0.4$. Because Chinese character string has the feature that the topic kernel lies often back of it, we define λ_i as formula (5).

$$\lambda_i = \left[i / \sum_{k=1}^K k + j / \sum_{k=1}^K k \right] / 2 \quad (5)$$

Where, i and j respectively expresses that element unit s_{ij} is the number i element of S_1 and the number j of S_2 . So formula (3) could be transformed as follows.

$$\mathbf{Sim}(S_1, S_2) = 0.6 * \frac{K}{M+N-K} + 0.4 * \sum_{i=1}^K \left[i / \sum_{k=1}^M k + j / \sum_{k=1}^M k \right] / 2 \quad (6)$$

We could see that, when computing the similarity of queries, formula(6) haven’t considered the similar units’ position information in the queries completely. For example, when computing the similarity between string ‘机微’ and ‘微机’ by formula (6), we could get \mathbf{Sim} (‘机微’, ‘微机’)= 1. The reason for this result is that, the assumption of $q(s_i)$ equal to ‘1’ is improper. Since, in addition to completely matching with the similar element’s literalness, the computation of similar units’ quantity is also correlated to the different positions of similar elements in different queries.

This paper introduces the moving distance ($\text{Distance}(s_{ij})=|i-j|$) of similar elements to optimize the value of $q(s_i)$. $|i-j|$ expresses the absolute value of the distance between the similar element s_{ij} ’s position in S_1 and in S_2 . Taking the moving cost factor into account, $q(s_i)$ could be computed by formula (7):

$$q(s_i) = \frac{1}{1+|i-j|} \quad (7)$$

And, formula (3) would be transformed into:

$$\mathbf{Sim}(S_1, S_2) = 0.6 * \frac{K}{M+N-K} + 0.4 * \sum_{i=1}^K \left[\frac{i / \sum_{k=1}^M k + j / \sum_{k=1}^M k}{2} \cdot \frac{1}{1+|i-j|} \right] \quad (8)$$

According to formula (8), the similarity between query ‘机微’ and ‘微机’ is \mathbf{Sim} (‘机微’, ‘微机’)= 0.8.

Furthermore, the difference between literal similarity and word's similarity is that the similar elements' granularity is different. Considering the methods of similarity computation, they are both based on the literal feature. Therefore, in essence they are the same.

3.3 Quantity Computation of Multi-feature Similar Units

A key step of seeking the queries similarity is the computation of similar units' quantity $q(s_i)$. Obtaining similar units is the necessarily previous step of computing similar units. But in fact, because the similarity of reviewed object is very complex, the similar units are difficult to obtain. This paper takes a simple strategy, which takes a certain feature of the elements as the foundation to judge whether they are similar units. That is, giving a threshold quantity δ , taking a certain feature as object, and without taking into account the position difference of elements in different queries. If the similarity of this feature between element a_i and b_j , i.e. $q(s_i)$, is exceed δ , a_i and b_j would be similar elements. When judging whether two elements are similar elements or not, this paper is based on the literal feature, semantic feature and statistic relevant feature.

For semantic feature, we could set $\delta_1=0.25$. The queries are segmented by semantic dictionary, i.e. Ψ , and the segmented results are represented as semantic codes: Code[i], Code[j]. Without taking into account the position difference of elements in different queries, if the similarity between Code[i] and Code[j] is greater than 1/4, the two elements could be viewed as similar. $q(s_i)_1$ could be computed as follows.

$$q(s_i) = \begin{cases} 1 & \text{if } strcomp(\text{Code}[i], \text{Code}[j]) = 0 \\ 1/[2 * (6 - n)] \cdot \frac{1}{1 + |i - j|} & \text{elsewise} \end{cases} \quad (9)$$

Where, n stands for the first different layer number in the processing of comparing the two semantic codes from root nod, $n \in [1, 5]$.

For literal feature, without taking into account the position difference of elements in different strings, we set $\delta_2=1$. That is, $q(s_i)_2$ could be got by formula (7). If we just consider the literal feature, the similarity computation of strings would be degenerated to the literal similarity computation, which is the situation of formula (6).

For statistical relevant features, without taking into account the position difference of elements in different queries, we set $\delta_3=0.5$. Statistic relevant degree is measuring the similarity between elements with a view to statistic distribution. Through the training corpus resources, we compute the mutual information between queries. For words in queries, i.e. a_i and b_j , if their mutual information $MI(a_i, b_j)$ is greater than the threshold quantity δ_3 , we could consider that they are similar and could save the computation result into the statistical relevant table. It could be noted as Tab_Relation. When computing the statistical relevant similarity between elements a_i and b_j , we could find it in Tab_Relation directly. If elements a_i and b_j belong to Tab_Relation, it would mean that the two elements are similar. $q(s_i)_3$ could be computed as follows.

$$q(s_i)_3 = \frac{MI(a_i - b_j)}{Max(MI)} \quad (10)$$

And, Max(MI) is the maximal value of the words' mutual information in Tab_Relation.

After considering the multiple features, it's hard to estimate the influence weight λ_i of each similar unit s_i gives to queries' similarity. This paper takes it as equal weight, that is $\lambda_i = 1/K$. And K is the quantity of similar unit.

3.4 Description of Similarity Computation Algorithm of the Queries

For the given queries S_1 and S_2 , the similarity between them could be computed by formula (3). If query $S_1 = \{a_1, a_2, \dots, a_i, \dots, a_m\}$ and $S_2 = \{b_1, b_2, \dots, b_j, \dots, b_n\}$ are completely different, we could use Ψ to segment S_1 and S_2 by the maximal matching segment method. The result would be $S_1' = \{A_1, A_2, \dots, A_i, \dots, A_M\}$, $S_2' = \{B_1, B_2, \dots, B_j, \dots, B_N\}$. Clearly for A_i (or B_j), if A_i is not belong to Ψ , A_i would be single character. If A_i and B_j are belong to Ψ , the similar unit's quantity $q(s_i)$ could be computed according to the priority of 'semantic > statistical relevance > literalness'. The detailed algorithm of the queries' similarity computation could be described as follows.

Algorithm: Similarity_Query compute the similarity between character query S_1 and S_2

Input: character string S_1, S_2

Output: **Sim**, the similarity of the queries: S_1, S_2

Process:

- Initialize: **Sim**=**Q_s**=0.0, $M = N = K = \text{Num} = 0$, $\alpha = 0.6$, $\beta = 0.4$, $\delta_1 = 0.25$, $\delta_2 = 1$, $\delta_3 = 0.5$
- Segment S_1 and S_2 by Ψ , get $A[i]$, $[j]$, and create the Corresponding semantic codes
- For each $A[i]$
 - For each $B[j]$
 - If $A[i], B[j] \in \Psi$ then
 - If $q(s_1)_1 \geq \delta_1$ then
 - $K = K + 1$, $M = M + 1$, $N = N + 1$
 - Compute $q(s_1)_1$
 - Else if $q(s_1)_3 \geq \delta_3$ then
 - $K = K + 1$, $M = M + 1$, $N = N + 1$
 - Compute $q(s_1)_3$
- Segment the element $A[i]$ or $B[j]$ which has no corresponding similar element by single character
- $\text{Num} =$ 'the number of the same single character between S_1 and S_2 ', $K = K + \text{Num}$, $M = M +$ 'number of the single characters of which couldn't match on literalness of S_1 ', $N = N +$ 'number of the single characters which couldn't match on literalness of S_2 ', compute $q(s_1)_2$
- $\lambda_i = 1/K$, $Q_s = \sum_{i=1}^K \lambda_i q(s_i)$
- **Sim** = $\alpha \cdot Q_n + \beta \cdot Q_s = \alpha \cdot K / (M + N - K) + \beta \cdot Q_s$
- Return **Sim**, the similarity of the queries.

4 Experiments and Results

This paper has done two experiments, and each experiment computes the queries' similarity based on literal, semantic and multiple features. The first group experiment is a close testing, whose objects are Chinese queries. It tests the synonym search of Chinese words and phrases. The test processing is: first, draw out 100 pairs of economy and military queries from the search log database, which are viewed as highly similar to each other. Then, make them into disorder and generate nearly 40,000 pairs of synonym automatically. Then compute the similarity by the methods based on literal similarity, semantic similarity and multiple features respectively. After that, select those words which similarity is greater than 0.66 to compare with the first 100 pairs of words. The testing result shows as table 1.

The second group experiment is an open testing. Testing set is made up of unordered queries. And the search testing of synonyms are doing based on the open set. The test processing is: select 891 queries of politics and 200 queries of economy, and make each pair of these words out of order, then compute their similarity and choose those words whose threshold quantity is greater than 0.66, then identify these words by hand. According to the similarity, we could divide them into synonym, quasi-synonym, hypogynous word, relevant words and irrelevant words. The first three kinds could be viewed as synonym, while the other two kinds are no-synonym. The testing result shows as table 2.

From the statistic data of table 1, we could see that through the experiment of searching for synonymic compound word, the recall of the pairs of synonym could

Table 1. The search result of synonymic compound word corresponding to the Chinese compound word

Domain	Testing words	word pairs	Recall (%)		
			A	B	C
economy	196	38,213	40	87	93
military affairs	200	39,800	49	87	92
Total	396	78,013	44.5	87	92.5

Notes: A,B,C stands for the, literal similarity measurement, semantic similarity measurement and multi-feature-based similarity measurement.

Table 2. The synonym extraction result on open testing set

Domain	Word pairs (Sim \geq 0.66)	Precision (%)		
		A	B	C
politics	347	11.24	24.78	27.64
economy	4,730	10.34	23.70	29.34
Total	5,077	10.40	25.52	29.23

achieve 92.5 % when the computation is based on multiple features, and 87% when based on semantic similarity, and just 44.5% when based on literal similarity. All this showed that, judging from the angle of the recall, the method based on multiple features is much better than the method just based on semantic or literal similarity.

From the data in table 2, we could see that, the precision is 29.23 % when using the method based on multiple features to recognize the synonym, and 25.52 % when based on semantic similarity, and just 10.40 % when based on the literal similarity. It shows that, on the aspect of synonym identification, the method based on multiple features is better than the method based on just literal or semantic similarity. This indicates that using the method based on multiple features is much more effective than the method based on just literal or semantic similarity. Besides, by the method based on multiple features, the rate of the searched relevant words whose similarity is greater threshold quantity is 80.08%, which is higher than the searching result by the other two methods, whose rate is respective 72.15% and 60.78%. It indicates that, the method has obvious advantage when it is used on the queries clustering.

5 Conclusion and Future Work

This paper builds a unified model to compute the queries' similarity. It takes similar unit as the queries' basic processing unit and considers the similar unit's literal, semantic and statistical relevant features synthetically. At the same time, the model amends the position information missing problem in the processing of sorting similar unit. The result of the experiments shows that, the method based on multiple features is efficient and it also has heuristic significance in the similarity computation between sentences and paragraphs. For the research of queries' similarity is also related to the knowledge of semantics, system theory and so on. For there are some questions existed in the present semantic system, the setting of similar unit's weight still needs farther research, i.e. estimate the combination coefficients from the data instead of using predefined value. Furthermore, the future work includes also testifying whether the method is applicable to other oriental languages, especially the languages in which Chinese characters are not used. It will be interesting to see the application of the proposed algorithm in English queries, running in a larger text corpus.

Acknowledgments. We thank the reviewers for the excellent and professional revision of our manuscript.

References

1. Zhou ML. Some concepts and mathematical consideration of similarity system theory. *Journal of System Science and System Engineering* 1(1)(1992)84-92.
2. Monge AE, Elkan CP. The field-matching problem: algorithm and applications. *Proceedings of the Second Internet Conference on Knowledge Discovery and Data Mining, Oregon, Portland(1996)267-270.*

3. Nirenburg S, Domashnev C, Grannes DJ. Two approaches to matching in example-based machine translation. Proceedings of TMI-93, Kyoto, Japan(1993)47-57.
4. <http://metadata.sims.berkeley.edu/index.html>, accessed: 2003.Dec.1.
5. Crouch CJ. An approach to the automatic construction of global thesauri. Information Processing and Management 26(5)(1990)629-640.
6. Lin DK. Automatic retrieval and clustering of similar words. Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, Montreal(1998)768-774.
7. Peter D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. Proceedings of the 12th European Conference on Machine Learning, Freiburg(2001) 491-502.
8. Weeds J. The Reliability of a similarity measure. Proceedings of the Fifth UK Special Interest Group for Computational Linguistics, Leeds(2002)33-42.
9. Pierre P. Senellart. Extraction of information in large graphs: Automaitc search for synonyms. Masters Intership Reports. University catholique de Louvam, Louvain-la-Neuve, Belgium(2001)1-17.
10. Resnik P. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language, Journal of Artificial Intelligence research 11(1999)95-130.
11. Li SJ, Zhang J, Huang X, Bai S. Semantic computation in Chinese question-answering system, Journal of Computer Science and Technology 17(6)(2002)933-939.
12. Mei Jiaju. *Tongyici Cilin*. Shanghai Lexicographical Publishing House(1983).