

國立台灣大學電機資訊學院資訊工程學研究所
碩士論文



Graduate Institute of Computer Science & Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

以彈性的方式算網頁服務的相似度

An elasticity approach to measuring web services similarity

駱亞倫

Alan Loh

指導教授：李允中 博士

Advisor: Jonathan Lee, Ph.D.

中華民國 106 年 7 月

July 2017



Acknowledgments

I would like to thank Professor Jonathan Lee for his patience and guidance during my two years as his graduate student. I would also like to thank Tony Zhang and Brian Liu for providing the framework and tools needed to complete my research. In addition, I want to give credit to Uiling Hsu for helping establish the foundation for SDEGM from which I was able to build my research off of. And lastly, I would like to thank my family and friends for their support and encouragement throughout the entire two years of my studies.



摘要

爲了符合使用者的需求，網頁服務組成是在服務導向演算法中的一個重要層面。在這當中，服務相配是服務組合中一個不可或缺的部分。可是在做服務間彼此相配時，爲了能夠達到一定的精準度與效率，是否能夠在形容服務的文件裡找出什麼是重要或關鍵的功能是必須的。透過隱含狄利克雷分布(LDA)模型來取出標題導向的做法，我們用一個類似層級分析法(AHP)的架構，定出一定數目的主題來當作條件並以文件的關鍵詞來當替換選項。從一個訓練好的LDA模型我們可以取得主題的機率分佈，並透過模糊隸屬函數來算出每個主題的權重。文件的關鍵詞在每個主題裡的重要性也是透過相似的方法找出來的。透過語言量化(linguistic quantifier)把有序加權平均(OWA)加到AHP的架構裡之後，就可以找出每一個關鍵詞在文章裡的重要值，也因此建立關鍵詞所代表的部件的重要性。

關鍵詞： 層級分析法，序加權平均，隱含狄利克雷分布，服務相配，語言量化，關鍵字重要性



Abstracts

Web service composition is an important aspect of service-oriented computing (SOC) for fulfilling users' requests. As such, service matching plays an indispensable role in the process of composing services. However, in order to accurately and efficiently match services with one another, it is essential to be able to discern the important components of the service to be matched within its documentation. By taking a topic-oriented approach through the use of the Latent Dirichlet Allocation (LDA) model, we follow a framework similar to those of Analytic Hierarchy Process (AHP), with a set number of topics acting as the criteria and the document's keywords as the alternatives. In using a trained LDA model, a probability distribution of topics can be attained, used to calculate the weight of each topic through a membership transformation function. Keyword importance under each topic is also found using a similar process, and by incorporating Ordered Weight Averaging (OWA) into AHP through the use of linguistic quantifiers, an importance value for each keyword within the document can be found, and thus establishing the importance of the component the keyword represents.

Keywords: Analytic Hierarchy Process, Ordered Weight Averaging, Latent

Dirichlet Allocation, Service Matching, Linguistic Quantifier, Keyword Importance





Contents

Acknowledgments	i
Abstracts	iii
List of Figures	vii
List of Tables	viii
Chapter 1 Introduction	1
Chapter 2 Related Work	3
Chapter 3 Analytic Hierarchy Process	4
3.1 AHP Tree	4
3.2 Pairwise Comparison	6
3.3 OWA Extension of AHP	7
Chapter 4 Finding Importance Through Probability	11
4.1 LDA	11



4.2 Importance Transformation Function	13
Chapter 5 Testing and Evaluation	17
5.1 SDEGM	17
5.2 Benchmark	18
5.3 Testing and Results	20
Chapter 6 Conclusion	24
Bibliography	25



List of Figures

3.1	General model of the AHP hierarchy	5
3.2	Document-Topic-Keyword AHP hierarchy	6
4.1	LDA Model Training and Inference	12
4.2	Probability Distribution of LDA Model	13
4.3	Reordering of probability distribution	15
4.4	Different kinds of distribution for α	15
4.5	Importance distribution corresponding with Figure 4.3	16
5.1	SDEGM graph construction process	18
5.2	SDEGM graph matching process	19
5.3	Individual query results for controlled (solid orange) and experimental ("all") (dashed blue)	22



List of Tables

3.1	Scales for pairwise comparisons	7
3.2	Common linguistic quantifiers	9
5.1	Testing Results	21
5.2	Further testing results	23



Chapter 1

Introduction

Web service discovery is a crucial feature in service-oriented computing (SOC) for locating, selecting and binding services to meet users requests. Usually, published services offer similar features but with variations in service interfaces, attributes, and quality-levels. These services often come with documentation describing their functions and features, with one popular format being the Web Services Description Language (WSDL). While many tools and algorithms already exist for extracting keywords from these documents, few, if any, are capable of measuring the importance or relevance of the keyword to the nature of the service itself.

In this work, we propose a topic-based approach in measuring the importance of a keyword within a document. Chapter 3 will introduce the framework of the analytic hierarchy process (AHP), establishing the idea of using topics as criteria and keywords as alternatives in calculating importance of each keyword, as well as the OWA extension of AHP that will be used to calculate the final importance value

of each keyword within the document. Chapter 4 will briefly introduce the LDA model used to calculate a document's topic and keyword probability, and go into detail on how the probability value is transformed into importance weights through membership functions of fuzzy classes. Chapter 5 will detail the implementation of this methodology within the Service Discovery through Elasticity-Based Graph Matching (SDEGM) framework, and discuss the results of testing against an established OWLS-TCv4 benchmark. Chapter 6 will establish the conclusion formed from the discussion of the testing results.



Chapter 2

Related Work

In the domain of text-based service matching employing information retrieval techniques to retrieve relevant services based on WSDL, Dong et al. proposed Woogle search engine [4] to calculate the similarities between queries and service operations according to term relationships built by a clustering algorithm. Stroulia and Wang [9] adopted IR techniques such as TF/IDF (term frequency / inverse document frequency) and WordNet ontology to calculate similarity between two WSDLs in order to support query-by-example service discovery. This method exploited the syntactic structure of the WSDL specifications datatypes, messages and operations and the natural-language semantics of their identifiers, comments, and descriptions. Plebani et al. presented an evaluation called URBE (Uddi Registry By Example) [5]. URBE analyzed structures and terms of WSDL documents to measure the similarity between multiple Web service interfaces. URBE also utilized WordNet and domain ontology to realize a semantic-oriented variant for enhancing precision.



Chapter 3

Analytic Hierarchy Process

The analytic hierarchy process [7] is a decision making methodology that breaks down complex problems by analyzing and structuring them into a hierarchial framework. In this section, we will go into detail over how the hierarchial tree structure is incorporated into our methodology for finding keyword importance, as well as how and why we used the ordered weighted averaging (OWA) extension of AHP [11] instead of the original pairwise comparison system.

3.1 AHP Tree

In general, regardless of the decision problem at hand, the first step in the AHP procedure is to decompose the issue into a hierarchy that usually consists of an end goal on top, criteria and subcriteria in the middle, and a list of alternatives at the bottom. The criteria level are generally listed from the most general objective to

the most specific one in descending order, as can be seen Figure 3.1.

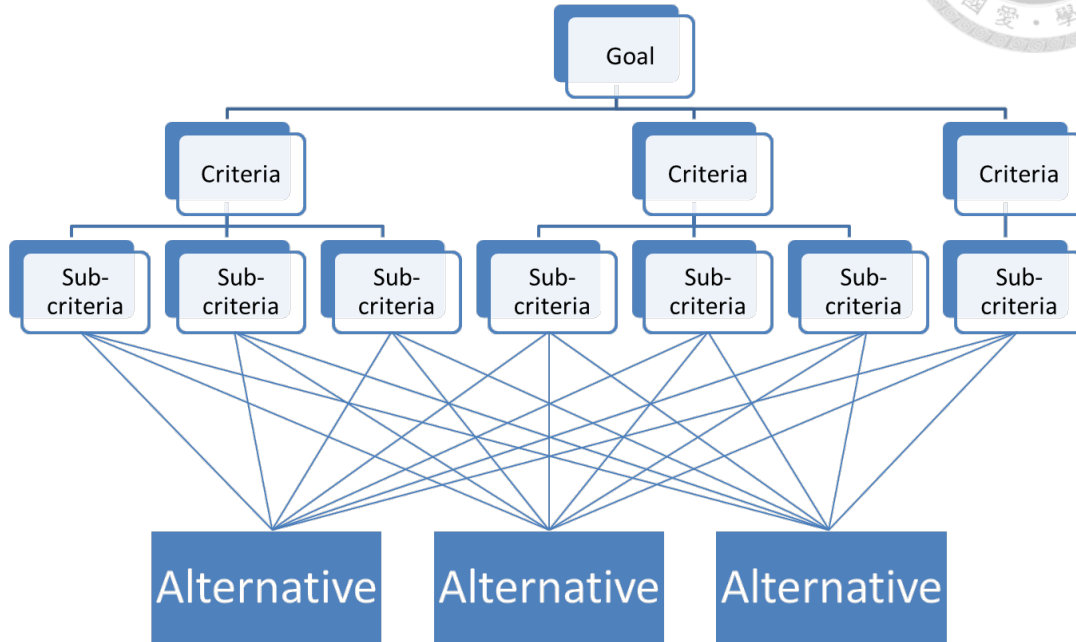


Figure 3.1: General model of the AHP hierarchy

For the purpose of this work, in order to adapt this framework for our topic-based approach in finding keyword importance within a document, a simplified version of the hierarchy was used. As seen in Figure 3.2, much of the structure remains the same, with the only changes being in the terminology used on each level.

A key component of the AHP framework is the calculation of the relative weights of each decision elements within the hierarchy. For our document-topic AHP model, it is only by finding the weight of each topic and the score of each keyword under each topic that we are able to find the final importance value of a keyword within the document.

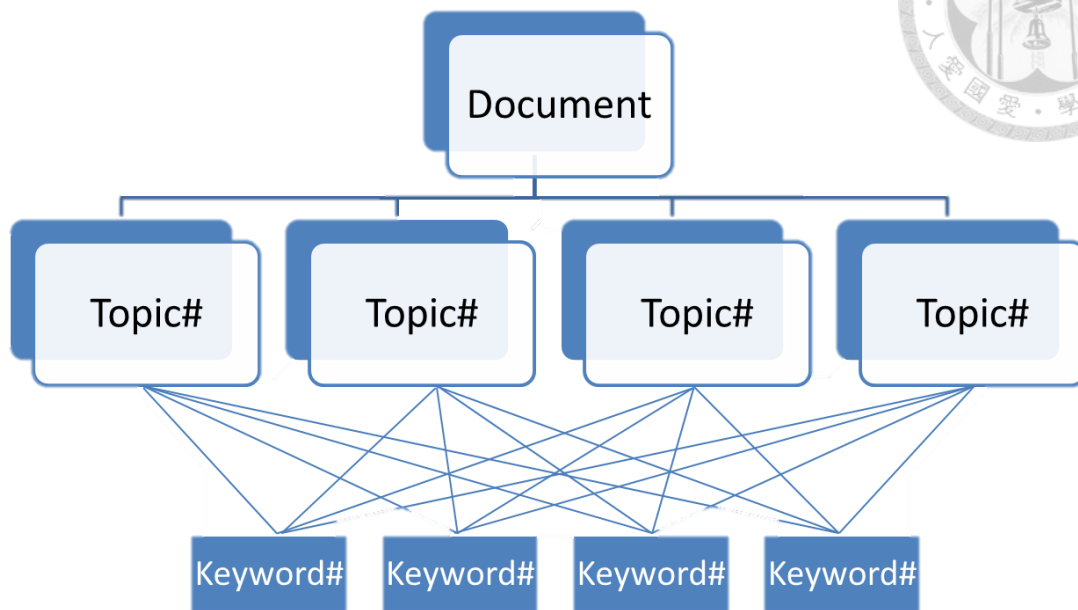


Figure 3.2: Document-Topic-Keyword AHP hierarchy

3.2 Pairwise Comparison

The original AHP methodology uses pairwise comparison to find the relative importance of each element compared with its peers under each criteria. By using an underlying semantical scale with values from 1 to 9 to rate the relative preferences for two elements of the hierarchy, as seen in Table 3.1, a reciprocal matrix is generated and used to calculate the relative importance of each element through the calculation of its normalized eigenvector [7]. As this comparison is largely based on human judgment, some degree of imperfection or inconsistency is to be expected, and can be quantified through the comparison of the consistency index and the associated consistency ratio of the eigenvector against a randomly generated pairwise



Table 3.1: Scales for pairwise comparisons

Intensity of importance	Verbal judgment of preference
1	Equal importance
3	Moderate importance
5	Strong importance
7	Very strong importance
9	Extreme importance
2, 4, 6, 8	Intermediate values between adjacent scale values

comparison matrix [8].

However, it was determined that this methodology was ill-suited for our purposes. The reliance on human judgment issue aside, while it is true that pairwise comparison would allow us to rank a document's keyword in order of importance, the ranking would be limited within the context of the document. In the test environment of matching multiple web services with each other, it is crucial that the importance values calculated for each keyword can be taken in an unrestricted context. As such, an extension of AHP that incorporates OWA was used in conjunction with the LDA model (discussed in Chapter 4) to find the weights of topics and the document's keywords.

3.3 OWA Extension of AHP

Ordered weighted averaging [10] by itself is a class of multicriteria combination operators, involving two vectors of values: the original criterion importance weights



and order weights. The order weights w_i , $i = 1, 2, \dots, n$ are such that

$$w_i \in [0, 1]$$

$$\sum_{i=1}^n w_i = 1$$

and

$$F(a_1, a_2, \dots, a_n) = \sum_{j=1}^n b_j w_j \quad (3.1)$$

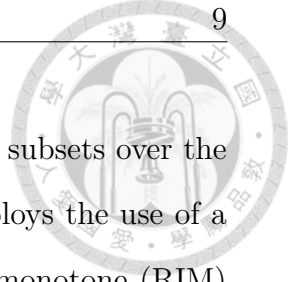
where b_j is the j -th largest element of the criterion importance weights $\{a_1, a_2, \dots, a_n\}$.

In other words, after reordering the original weights a_i from largest to smallest, each weight value is associated with an ordered weight that will ultimately be used to find $F(a_1, a_2, \dots, a_n)$.

While this may be no different than what AHP is already doing without the reordering of values, the inclusion of linguistic quantifiers is what sets the OWA extension apart from the original AHP [11].

If we go back to the original basis of AHP in decision making, the requirements of the decision-maker can vary in a range that goes from “all the criteria have to be satisfied” to “at least one criterion must be satisfied” for each alternative. The same idea can be applied to our topic-based importance finding of keywords in documents, where in a predefined number of topics, we can base the importance of our keyword off of the number of topics it fulfills.

There are two general classes of linguistic quantifiers: absolute and relative quantifiers. Absolute quantifiers can be defined as fuzzy subsets of $[0, +\infty]$, and can also be used to represent linguistic terms such as about 4 or more than 10. On the other hand, relative quantifiers consisting of proportional terms such as “most”, “few”,



“half” are used more for imprecise proportions, representing fuzzy subsets over the unit interval. The OWA extension of AHP that we are using employs the use of a class of relative quantifiers used known as the regular increasing monotone (RIM) quantifiers. It is represented with Q acting as the linguistic quantifier and associated with α value, as can be seen in Table 3.2. Thus, a fuzzy subset over the unit interval $[0, 1]$ can be represented where for each p in the unit interval, the membership grade $Q(p)$ indicates the compatibility of p with the concept denoted by Q .

$$Q(p) = p^\alpha \tag{3.2}$$

Whereas linguistic quantifiers usually denote the requirement “ Q criteria are satisfied by an acceptable solution” in decision making, the statement can be rephrased quite simply into “ Q topics are satisfied by a relevant keyword” for our purposes of finding keyword importance through a topic-based approach.

Table 3.2: Common linguistic quantifiers

Linguistic Quantifier (Q)	At least one	Few	Some	Half	Many	Most	All
α	0.0001	0.1	0.5	1	2	10	1000

To demonstrate how the OWA extension incorporates this back into AHP, let’s assume A_1, \dots, A_n are our criteria and v_i are the importance weight for each criterion. In addition, let $x \in X$ be an alternative being evaluated, with $\{a_1, a_2, \dots, a_n\}$ being the respective scores when evaluating $\{A_1(x), \dots, A_n(x)\}$. Similar to the OWA methodology at its basics in Equation 3.1, we reorder $A_i(x)$ such that b_j is the j -th

largest of a_1, a_2, \dots, a_n , and let u_j denote the importance weight associated with that criterion. Thus, if $A_5(x)$ is the largest score, then $b_1 = a_5$ and $u_1 = v_5$. Under these conditions, the OWA extension is able to construct an OWA operator of dimension n with weighting vectors defined by:

$$w_j(x) = Q\left(\frac{\sum_{k=1}^j u_k}{T}\right) - Q\left(\frac{\sum_{k=1}^{j-1} u_k}{T}\right) \quad (3.3)$$

Where the linguistic quantifier gets applied in the same way as Equation 3.2, and

$$T = \sum_{k=1}^n u_k$$

So when we finally calculate the total score of alternative x , we end up with:

$$D(x) = \sum_{j=1}^n b_j w_j(x) \quad (3.4)$$

By using this equation, with topics acting as our criterion and keywords acting as our alternatives, we will be able to calculate the overall importance of a keyword within a document. However, in order to do so, importance weights need to be decided for each topic and keyword. The proposed methodology to do so will be detailed in the next chapter.



Chapter 4

Finding Importance Through Probability

In this chapter, we will briefly discuss the use of the LDA model as our basis in deciding the topics used in our methodology and the importance weight associated with them. As LDA is a generative model that deals only in probability in its output, we will also go into detail on how probabilistic values are transformed into importance through a hypothesis test approach in fuzzy membership functions.

4.1 LDA

Latent Dirichlet allocation [3] is a generative probabilistic model for collections of discrete data such as text corpora. While the inner workings of this model is outside the context of this research, as it can be assumed that any probabilistic model of

text data can be used to achieve our purposes, LDA's use of topic probability is essential in establishing the topic weights used in the OWA extension of AHP as detailed in the previous chapter. By training a LDA model with a corpus and a pre-defined number of topics, we are able to categorize each word in such a way that we can tell what probability each topic has in generating each word from the corpus. Upon doing so, as can be seen in Figure 4.1, we are then able to input documents like WSDLs into the model in order to find the probability distribution of the trained LDA topics for this document. As further detailed in Figure 4.2, while the LDA model may have its own topic probability distribution based on the corpus it was trained with, inputting other documents into the model will generate their own topic probability distribution numbers. However, regardless of the inferred document, the word probability distribution will remain the same.

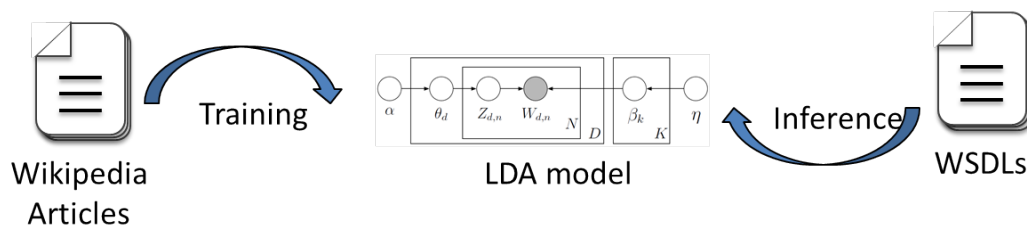


Figure 4.1: LDA Model Training and Inference

By establishing a probability distribution of topics for a document, we are able to infer what the core subject of the document may be about. With this knowledge, we can take the second step in finding how a document's keywords fits within the word probability distribution of each of these core topics. However, finding probability values alone is not enough to establish the importance of either the topics within a

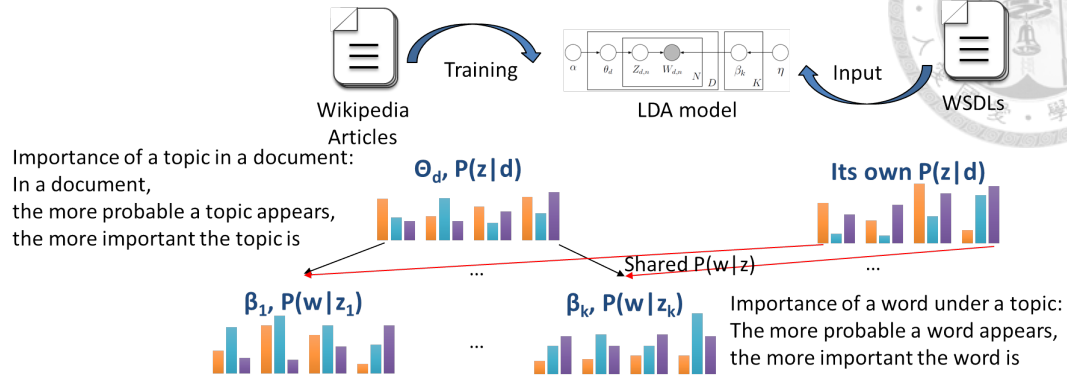


Figure 4.2: Probability Distribution of LDA Model

document or the keywords within a topic, and thus an importance transformation function is needed to achieve our purpose.

4.2 Importance Transformation Function

Pota *et al.* established a method to transform probability distribution into membership functions of fuzzy classes through a hypothesis test approach [6]. By stating a null hypothesis H_0 for testing based off of our topic or word probability distribution, we can establish the importance of that topic or word through the amount of simulated acceptance or rejection of the hypothesis. For example, if we wanted to find the importance of topic k given document d , we could form the null hypothesis:

$$H_0 \equiv \text{“Topic } k \text{ is relevant to Document } d\text{”}$$

As with all hypothesis-based testing, in order to determine whether or not to reject a hypothesis, the *p-value* of the test statistic needs to be compared with the decision rule α . Since we mean to know whether or not something is relevant to



another, we assume that we always want to perform a two-tails test, and thus always use Equation 4.1 to do so:

$$p\text{-value}(x) = \begin{cases} 2 \cdot (1 - CF_D(x)), & \text{if } [x > x^{me}] \\ 1, & \text{if } [x = x^{me}] \\ 2 \cdot CF_D(x), & \text{if } [x < x^{me}] \end{cases} \quad (4.1)$$

Which can be simplified to

$$p\text{-value}(x) = 2 \cdot (0.5 - |0.5 - CF_D(x)|) \quad (4.2)$$

Where x is the keyword or topic in question, and x^{me} is the median of its distribution, and $CF_D(x)$ is the cumulative probability function of its distribution.

While this may seem difficult to do given our context, since LDA's probability distribution of document, topic, and keywords can be considered to be a categorical distribution, this problem can be overcome by simply reordering the values into a normalized curve, as seen in Figure 4.3, thus allowing us to take the cumulative probability function for any single topic or keyword.

In order to accurately simulate a polling of opinions across a wide variety of perceptions, a distribution of α needs to be decided by deciding on the limiting values $\alpha_{min} (\geq 0)$ and $\alpha_{max} (\geq \alpha_{min} \text{ and } \leq 1)$, as well as type of distribution (constant, linearly increasing, linearly decreasing), as seen in Figure 4.4.

By incorporating the previously discussed $p\text{-value}$ of x keyword or topic with the α distribution, the fuzzy membership function can finally be established for x by finding the cumulative probability value of its $p\text{-value}$ within the α distribution.

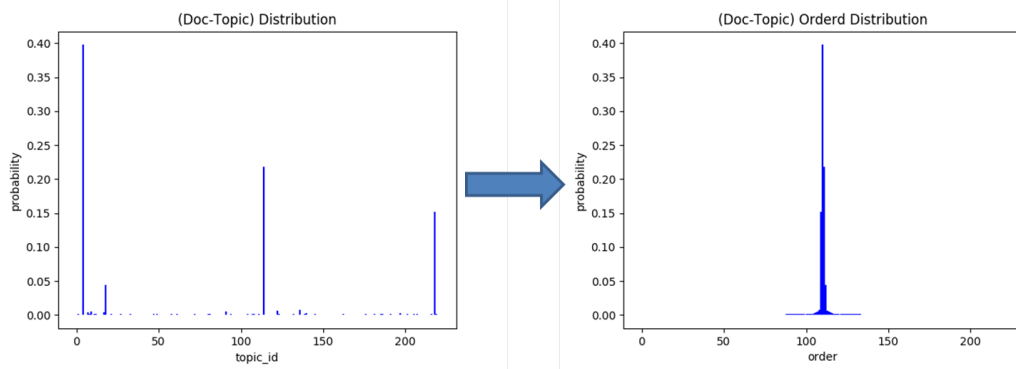


Figure 4.3: Reordering of probability distribution

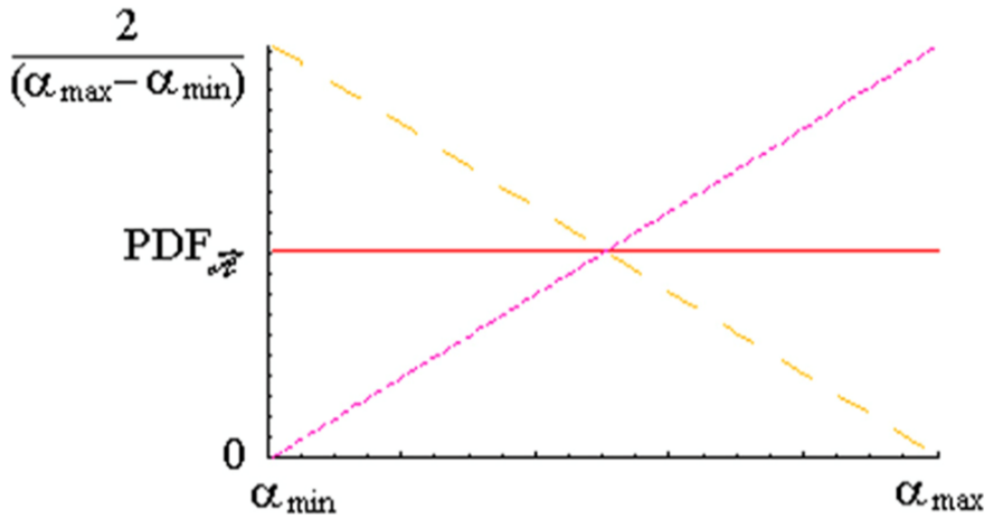


Figure 4.4: Different kinds of distribution for α

Since we're always performing the two-tails test, in incorporating Equation 4.2, the importance of x is found with the following equation:

$$I(x) = CF_A(p - value(x)) = CF_A(2 \cdot (0.5 - |0.5 - CF_D(x)|)) \quad (4.3)$$

By combining this with the probability found with the LDA model, we can finally establish a corresponding importance distribution for Figure 4.3, as seen in Figure 4.5. By simply performing a value look-up for each topic and document keyword, overall importance of a keyword within a document can be found using the OWA extension of AHP model discussed in the previous chapter.

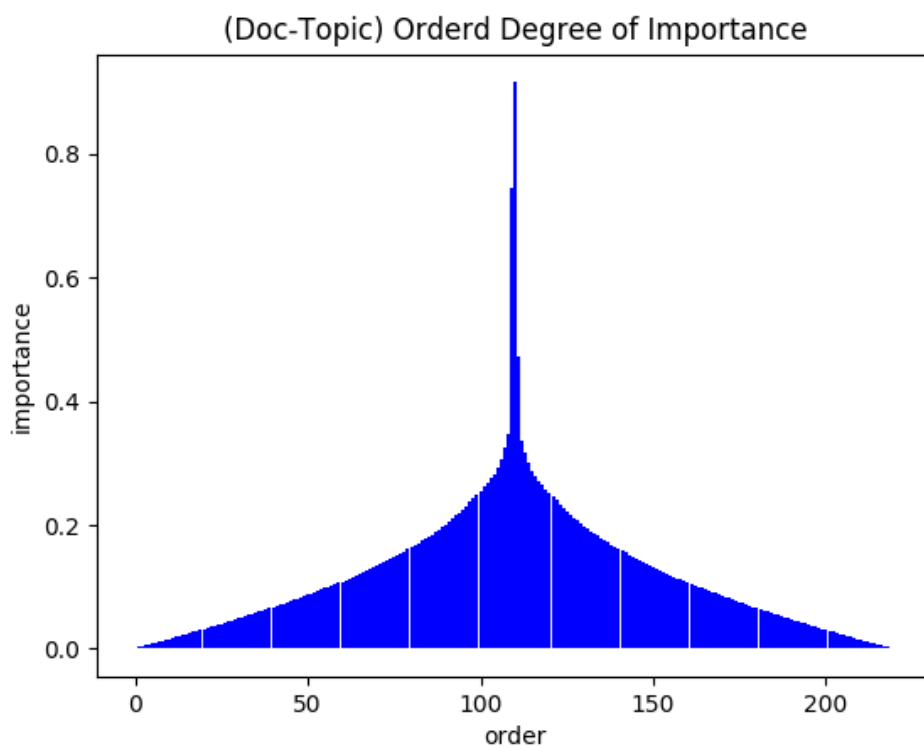


Figure 4.5: Importance distribution corresponding with Figure 4.3



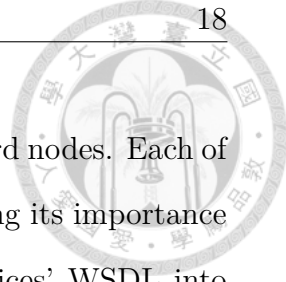
Chapter 5

Testing and Evaluation

In this chapter, we will briefly describe the Service Discovery through Elasticity-Based Graph Matching (SDEGM) framework used to test our keyword importance finding methodology. We will also talk about the benchmark that will serve as the basis of our results, the controlled test we are comparing our results with, and the validation of our results using t-test statistics.

5.1 SDEGM

SDEGM is a web service matching algorithm that uses Web Services Description Language (WSDL) to locate web services based on similarity to an initial request service. It does this by extracting keywords from a WSDL and construct a three part graph based on the WSDL's structure, consisting of an operation part, an input part, and an output part, as can be seen Figure 5.1. The graph is made up



entirely from three kinds of nodes: connector, datatype, and keyword nodes. Each of these node type has an elasticity value associated with it, indicating its importance within within the graph. By transforming all inputted web services' WSDL into graph form, it will attempt to find an optimal pairing of services by turning each service pairing into a bipartite graph matching problem, as seen in Figure 5.2, with each node's elasticity serving as the basis for calculating the distance between the matched nodes. By integrating our keyword importance finding methodology into SDEGM for determining the elasticity of keyword nodes, we will be able to test its effectiveness in matching web services together against a set benchmark.

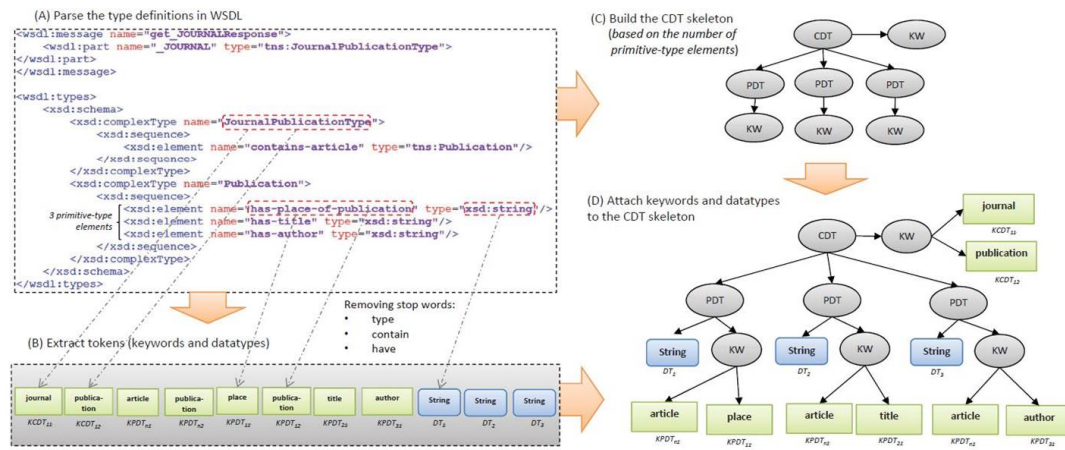


Figure 5.1: SDEGM graph construction process

5.2 Benchmark

Benchmark test data for experimental evaluation are obtained from OWL-S retrieval test collection v4 (OWLS-TC v4) [1], which consists of 1080 web services

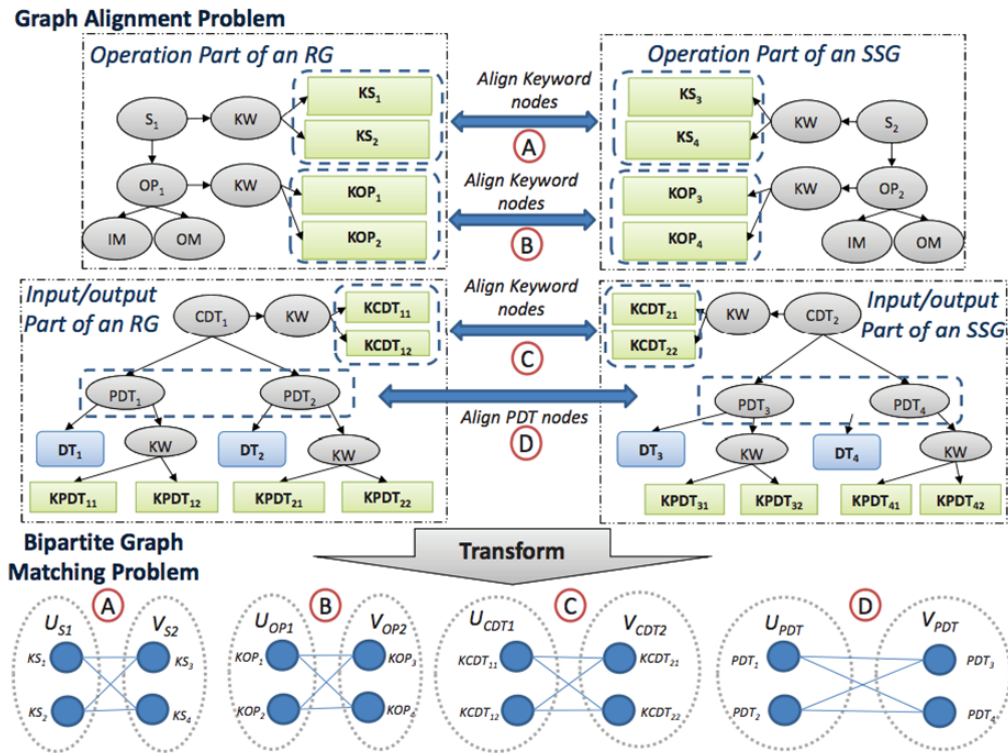
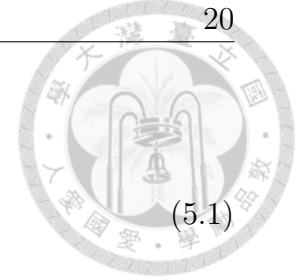


Figure 5.2: SDEGM graph matching process

described by OWL-S as well as WSDL, covering nine application domains: education, medical care, food, travel, communication, economy, weapon, geography, and simulation. This benchmark also includes 42 test queries described by WSDLs in OWLS-TC v4, each of which is associated with a set of relevant services defined by the benchmark.

The three main indicators of reference that will be used for the purpose of evaluating our service matching test results are Top-K Precision ($P^K(q_i)$), Average Precision ($AvgP(q_i)$), and Mean Average Precision ($MAP(Q)$), and are defined as follows



$$P^K(q_i) = \frac{|Rel(q_i) \cap Rank^K(q_i)|}{|Rank^K(q_i)|} \quad (5.1)$$

Where $Rel(q_i)$ is the set of relevant services for a given query q_i , and $Rank^K(q_i)$ stands for the set of top K web services result related to the query q_i .

$$AvgP(q_i) = \frac{\sum_j^{n(q_i)} P^j(q_i) \cdot Rel(i, j)}{Rel(q_i)} \quad (5.2)$$

Where $n(q_i)$ is the number of retrieved services for queries q_i and $Rel(i, j)$ is an indicator function for query q_i , equaling 1 if the item at rank j is a relevant service, zero otherwise.

$$MAP(Q) = \frac{\sum_i^n AvgP(q_i)}{n} \quad (5.3)$$

Where n stands for the number of queries being tested, which in our case would be 42.

5.3 Testing and Results

In regards to the SDEGM framework, we will only be focusing on service matching based off of keyword nodes alone, thus ignoring the datatype and connector nodes. In addition, the operation, input, and output parts of the service graph will all carry equal weight. Our control test will be set up such that all keyword nodes carry an elasticity value of 1, while our experimental test will be using our document-keyword importance finding methodology to determine elasticity value of

each individual keyword node. For the purpose of comparison, all linguistic quantifiers listed in Table 3.2 of Chapter 3 were included in the trials. The LDA model used for this test was trained with a corpus based off of a database dump of English Wikipedia articles [2] containing 750,906 terms. 221 topics were used in the training.

As can be seen in Table 5.1, compared to the controlled test of using a single constant value of 1, within the SDEGM framework, only a small improvement can be seen in the mean average precision for some of the quantifiers used across all 42 service queries. If we compare the individual query results between the controlled test and the experimental test using the quantifier “All”, as can be seen in Figure 5.3, we can see that while there were noticeable improvements in precision for some of the queries, the majority of them resulted in mostly the same value.

Table 5.1: Testing Results

Elasticity Method	Constants (KW, DT, C)	Graph Part Weights (OP, In, Out)	Quantifier	Mean Average Precision
Constant	1, 0, 0	1, 1, 1	-	0.799028
LDA_OWA	-, 0, 0	1, 1, 1	At least one	0.799077
LDA_OWA	-, 0, 0	1, 1, 1	Few	0.798609
LDA_OWA	-, 0, 0	1, 1, 1	Some	0.797513
LDA_OWA	-, 0, 0	1, 1, 1	Half	0.797719
LDA_OWA	-, 0, 0	1, 1, 1	Many	0.798101
LDA_OWA	-, 0, 0	1, 1, 1	Most	0.801592
LDA_OWA	-, 0, 0	1, 1, 1	All	0.809243

However, while the overall improvement in precision may be small between the controlled test and the experimental test using the quantifier “All”, the results cannot be considered statistically insignificant. A null hypothesis H_0 can be stated that “There is no difference between using constant values and the AHP OWA

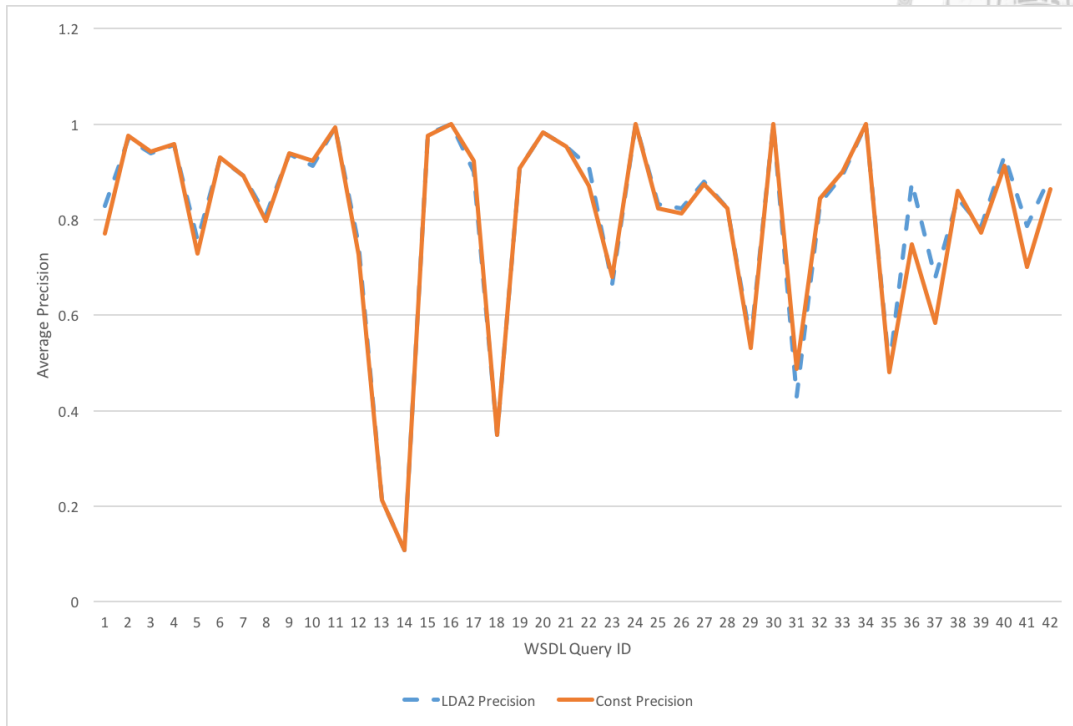


Figure 5.3: Individual query results for controlled (solid orange) and experimental (“all”) (dashed blue)

method”, while using an experimental hypothesis H_A stating “Using the AHP OWA method will result in better mean average precision than using constant values.” By finding the t-test statistic for 2 dependent means on a one-tailed hypothesis against the commonly accepted significant value $\alpha = 0.05$, a t -value of 2.07987 was obtained, with a resulting p -value of 0.02033. Since $0.02033 < 0.05$, our results can be considered statistically significant.

It is worth noting, however, that upon further testing under a different configuration of SDEGM where we only focus on the operation part of the service graph, the quantifier “At least one” performs noticeably better than the rest of the quantifiers,



while the quantifier “All” gives the worst mean average precision result, as seen in Table 5.2.

Table 5.2: Further testing results

Elasticity Method	Constants (KW, DT, C)	Part Weights (OP, In, Out)	Quantifier	Precision
Constant	1, 0, 0	1, 0, 0	-	0.891053
LDA_OWA	-, 0, 0	1, 0, 0	At least one	0.892334
LDA_OWA	-, 0, 0	1, 0, 0	Half	0.881510
LDA_OWA	-, 0, 0	1, 0, 0	Many	0.876903
LDA_OWA	-, 0, 0	1, 0, 0	All	0.862055



Chapter 6


Conclusion

In this paper, we introduced a new methodology of finding keyword importance within a document by a combination of using the OWA extension of AHP, linguistic quantifiers, LDA topical probabilistic model, and importance transformation through fuzzy membership function. As can be seen in the results of testing with the SDEGM framework, while there is potential for this methodology to be considered an improvement over using a single constant value for all keywords, the experimental results does not give a clear indicator that this is always the case. However, it must also be taken into consideration that the SDEGM framework is still a piece of research work in progress, and given the conflicting results seen in Table 5.1 and Table 5.2, it may be possible that the inconclusive results are caused by parameters outside the scope of this paper. Therefore, further testing is recommended under different service matching contexts and benchmark in order to thoroughly evaluate the effectiveness in this method of importance finding.



Bibliography

- [1] Owls-tc. <http://projects.semwebcentral.org/projects/owls-tc/>.
- [2] Wikimedia downloads. <https://dumps.wikimedia.org/>.
- [3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
- [4] X. Dong, J. Madhavan, and A. Halevy. Mining structures for semantics. *SIGKDD Explor. Newsl.*, 6(2):53–60, Dec. 2004.
- [5] P. Plebani and B. Pernici. Urbe: Web service retrieval based on similarity evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 21(11):1629–1642, Nov 2009.
- [6] M. Pota, M. Esposito, and G. D. Pietro. Transforming probability distributions into membership functions of fuzzy classes: A hypothesis test approach. *Fuzzy Sets and Systems*, 233:52 – 73, 2013. Theme: Fuzzy numbers and statistics.
- [7] T. L. Saaty. Tl. “decision making with the analytic hierarchy process. *International Journal of [Services Sciences (IJSSCI)*, pages 83–98, 2008.
- [8] T. L. Saaty and L. T. Tran. On the invalidity of fuzzifying numerical judgments in the analytic hierarchy process. *Mathematical and Computer Modelling*, 46(7):962 – 975, 2007. Decision Making with the Analytic Hierarchy Process and the Analytic Network Process.

- 
- [9] Y. Wang and E. Stroulia. *Semantic Structure Matching for Assessing Web-Service Similarity*, pages 194–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [10] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, Jan 1988.
- [11] R. R. Yager and A. Kelman. An extension of the analytical hierarchy process using owa operators. *J. Intell. Fuzzy Syst.*, 7(4):401–417, Dec. 1999.