Clustering Web People Search Results Using Fuzzy Ant-Based Clustering

Priya Venkateshan
priyaven@gmail.com

ABSTRACT
In this paper, we describe a system to cluster results of people-search which does not require apriori information about the number of clusters the data needs to be clustered into, using Fuzzy Ant-Based Clustering.

General Terms
Algorithms.

Keywords
Web People Search, Fuzzy Ant-Based Clustering

1. INTRODUCTION
In browsing through web search results, often we find a flat ranked list insufficient in giving us a good user experience. This is especially so when the results are all pertaining to different topics. It helps in finding more relevant results quickly if similar results are clustered together.
This is especially true in case of people search, where more often than not, a user is looking for a specific person, and is presented with relevant results interspersed between a mire of irrelevant results, which mostly pertain to others with the same name.
Thus, clustering of people search results such that all the results pertaining to a particular person are all in the same cluster, would aid in quickly browsing through the results and in finding relevant information more quickly.
Various crisp clustering algorithms have been used to cluster web search results. The issue with these algorithms is that they require a predefined number of clusters to cluster the results. A one-size-fits-all approach will not work here, as the number of clusters might vary for the same number of results depending on various parameters like the commonness of the name and the presence of celebrities with that name, among other factors.
Thus there exists a need for a clustering scheme which does not need a predetermined number of clusters.
We attempt to use Fuzzy Ant-Based Clustering [1] for this purpose.

2. SYSTEM DESCRIPTION
2.1 Preprocessing
Before implementing the algorithm on the documents, we preprocess the documents and represent them in the vector-space model, using tf-idf. This involves the following steps:

- Conversion of HTML to Plaintext.
- Removal of the search string and stopwords from the documents. We used a standard list of English stopwords. The search string needs to be removed as all the documents contain it and it would unnecessarily add to the list of terms, while adding little value.
- Apply a stemming algorithm on the documents. We used Porter's Stemmer algorithm.
- Removal of words that occur only once. These are unlikely to help in clustering documents, and hence are removed.
- Extraction of only those words that are the top 10 most frequently occurring words in the document. This is done with an eye on efficiency.
- Representation of document in the vector space model using TF-IDF.

2.2 The Clustering Algorithm
Ant-based clustering algorithms are usually inspired by clustering of dead nestmates, as observed under laboratory conditions. Without negotiating about where to gather the corpses, ants manage to cluster all corpses into one or two piles. The simplicity of the concept behind this phenomenon along with the lack of a centralized control and apriori information are the main motivations for designing clustering algorithms inspired by this behaviour.
Schockaert et. al. have proposed a clustering method where the desired behaviour of artificial ants (and their stimuli for picking up and dropping items) is expressed flexibly by fuzzy IF-THEN rules. It is a suitable algorithm to be used in the context of clustering web people search results (or for that matter, any sort of web search results) as it requires no apriori information on the number of clusters.
The probability that an ant starts performing a task with stimulus s and response threshold value u is given by

$$T_n(s;\theta)=s_n/(s_n+\theta_n) \qquad (1)$$

where n is a positive integer. In fact, this is a slight generalization that was also used in Ref. 2; in Ref. 18 only the case where n 2 is considered. We will assume that $s \in [0.1]$ and $\theta \in]0,1]$

Let us now apply this model to the problem at hand. A loaded ant can only perform one task: dropping its load. Let s_{drop} be the stimulus associated with this task and u_{drop} the response threshold value. The probability of dropping the load is then given by

$$P_{drop} = T_{ni}(s_{drop} ; \theta_{drop})) \qquad (2)$$

where $i \in \{1, 2\}$ and n1 , n2 are positive integers. When the ant is only carrying one item n_1 is used; otherwise n_2 is used. An unloaded ant can perform two tasks: picking up one item and picking up all the items. Let s_{one} and s_{all} be the respective stimuli and u_{one} and u_{all} the respective response threshold values. The probabilities for picking up one item and picking up all the items are given by

$$P_{pickup\_one} = (s_{one}/(s_{one}+ s_{all})). T_{m1} (s_{one} ; \theta_{one})) \qquad (3)$$

$$P_{pickup\_all} = (s_{all} / (s_{one} + s_{all})) . T_{m2} (s_{all} ; \theta_{all})) \quad (4)$$

where m1 and m2 are positive integers.

We assume that the objects that have to be clustered belong to some set U, and that E is a binary fuzzy relation in U, which is reflexive (i.e., E(u, u) = 1, for all u in U) and TW -transitive (i.e., TW(E(u, v), E(v, w)) ≤ E(u, w), for all u, v, and w in U). For u and v in U, E(u, v) denotes the degree of similarity between the items u and v.

During the execution of the algorithm, we maintain a list of all heaps. Initially there is a heap, consisting of a single element, for every object in the data set. Picking up an entire heap H corresponds to removing a heap from the list. At each iteration, our ant acts as follows:

- If the ant is unloaded, a heap H from the list is chosen at random.
  - If H consists of a single item, this item is always picked up.
  - If H consists of two items, a and b, both items are picked up with probability $E(a, b)^{k1}$ and one of the two items is picked up with probability $(1 - E(a, b))^{k1}$.
  - If H consists of more than two items, the probabilities for picking up a single element and for picking up all elements are given by formulas (3) and (4)
- If the ant is loaded, a new heap containing the load L is added to the list of heaps with a fixed probability. Otherwise, a heap H from the list is chosen at random.
  - If H consists of a single item a, and L consists of a single item b, L is dropped onto H with probability $E(b, a)^{k2}$.
  - If H consists of a single item and L consists of more than one item, the ant does nothing. The main reason for separating this special case is efficiency. Because the average similarity *avg*(H) will always be 1 in this case, the only situation where it would be desirable to merge H and L is when all the items in L are approximately equal to the single element in H. But in this unlikely case, L and H would be merged at a later iteration of the algorithm.
  - If H consists of more than one item, the probability that L is dropped onto H is given by formula (2). In the above, k1 and k2 are small integer constants.

Experimental results have indicated that c.n is a good estimation of the number of iterations that is required, where n is the size of the data set and c is a parameter that is dependent on the nature of the data set. For this task, we used c=5000.

.

3. RESULTS

The following results were obtained on applying various evaluation measures on the clustering results. The rationale behind the various evaluation measures are given in [3] and [4].

Table 1. Results obtained using various evaluation measures

BCubed Precision	0.61
BCubed Recall	0.38
F-measure of B-Cubed P/R with alpha set to 0.5	0.39
F-measure of B-Cubed P/R with alpha set to 0.2	0.37
Purity	0.71
Inverse Purity	0.48
F-measure of Purity and Inverse Purity with alpha set to 0.5	0.53
F-measure of Purity and Inverse Purity with alpha set to 0.2	0.49

4. FUTURE WORK

This is a rather basic implementation of the Fuzzy Ant-Based clustering algorithm for the purpose of web people search. It does not require a pre-determined number of final clusters.

In the future, we would like to improve upon the model by offering different weights for email IDs and domain names. Also, named entity recognition can be used to great use.

5. REFERENCES

[1] Schockaert, S., De Cock, M., Cornelis, C., and Kerre, E. E. 2007. Clustering web search results using fuzzy ants: Research Articles. Int. J. Intell. Syst. 22, 5 (May. 2007), 455-474. DOI= http://dx.doi.org/10.1002/int.v22:5

[2] Javier Artiles, Julio Gonzalo and Satoshi Sekine. 2009. WePS 2 Evaluation Campaign: overview of the Web People Search Clustering Task
In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, April.

[3] A comparison of extrinsic clustering evaluation metrics based on formal constraints.
E. Amigó, J. Gonzalo, J. Artiles, F. Verdejo.
Information Retrieval Journal. 2008.

[4] The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task.
Javier Artiles, Julio Gonzalo and Satoshi Sekine.
Proc. Int'l Workshop Semantic Evaluations (SemEval 2007).