GROUPER: A DYNAMIC CLUSTERING INTERFACE TO WEB SEARCH RESULTS
Oren Zamir & Oren Etzioni

Acar Erdinç Elif Dal Fatih Çalışır Tolga Çekiç

Old methods were used to bring the results as ranked lists. However, ranked list techniques had some drawbacks like long lists may occur and finding information from these lists might be hard. “Grouper” come up with a solution with clustering the results.

There were two ways to cluster: **pre-retrieval** or **post-retrieval** clustering. In pre-retrieval clustering, documents are clustered before results are obtained. Here, there are two drawbacks. First, pre-retrieval clusters might be based on features that are infrequent in the retrieved set. Second, many non-retrieved documents can influence the clusters. In post-retrieval clustering, documents are clustered after results were obtained. Post-retrieval clustering has better results respect to pre-retrieval clustering so Grouper preferred this way of clustering.

Grouper provides an **independent interface** for clustering which means search engine only searches and obtains results, and clustering interface than clusters these results. In this way, search engines aren’t be imposed with extra clustering works.

This independent interface for search engine has following three key features:
1- **Coherent clusters**: Similar documents should be clustered and overlapping clusters should be generated when appropriate.
2- **Efficiently browsable**: Cluster descriptions should be concise and accurate.
3- **Speed**: The clustering system should be fast.

Suffix tree clustering algorithm is used for Grouper because of several benefits. It is fast, incremental and running in linear time (# of documents). It produces coherent results. It is based on identifying phrases (ordered sequence of words) that are common to group of documents.

Suffix Tree Clustering has these main steps**:**
1- Document cleaning (html tags, numbers, punctuations),
2- Identifying base clusters (set of documents sharing a common phrase) using a suffix tree,
3- Merging base clusters into clusters.

Finally, a clustering interface for a web search engine “Grouper” is described in this paper. Grouper is fast and incremental but has some shortcomings. It does not capture semantic distinctions (after merging base clusters) so, large number of clusters may occur. Grouper II will come up with solutions to these shortcomings. It will allow viewing non merged base clusters and will support hierarchical and interactive interfaces.