SRCluster: Web Clustering Engine based on Wikipedia

Yuvarani Meiyappan[1], N. Ch. S. Narayana Iyengar[2] and A. Kannan[3]

[1]*Lead, Infosys Limited, Bangalore, Karnataka, India 560100*

[2]*Senior Professor, School of Computing Science and Engineering, Director, Periar EVR Central Library, VIT University, Vellore, Tamil Nadu 632014, India*

[3]*Professor, Department of Information Science and Technology, Anna University, Chennai, Tamil Nadu 600025, India,*

Yuvarani_meiyappan@infosys.com, nchsniyengar48@gmail.com, kannan@annauniv.edu

Abstract

Web clustering engine greatly simplifies the effort of the user from browsing the large set of search results by reorganizing them into smaller clusters. Current web clustering engines result in additional clusters and misses out few relevant, leading to lack of predictability of clustering outputs. Web clustering engines produces inconsistent results as the content of the cluster do not always correspond to its label. In this paper, a new web clustering engine named SRCluster has been proposed to overcome these deficiencies, in specific for the polysemy unigram search keywords. SRCluster identifies the possible categories and its label for the given polysemy keyword based on Wikipedia. The system determines the improved Lesk score (termed, SRLesk score) for each of the category. The search result is clustered to the category with the maximum SRLesk score. The hypertext of the disambiguation Wikipedia page is utilized for labeling the cluster. The experimental result on AMBIENT dataset shows that the inconsistency and the lack of predictability of clustering outputs is being improved using SRCluster.

Keywords: *Web clustering engine, Search result clustering, Polysemy, Wikipedia, web search result, clustering, disambiguation*

1. Introduction

Search engines are inevitable tool for retrieving information from the Web. While the ranked results from search engines for certain search are definitely good, they are less effective for certain queries especially when they are short, ambiguous, polysemy. The volume of ranked results retrieved by search engines is overwhelming and includes the subtopics, different meanings for the given query. The user has to look for the relevant information among the results retrieved. The search result clustering is one among the approaches that enables the user to locate the relevant information among the search results with less effort. It combines the query-based and category-based search. Web clustering engine, the system that perform clustering of Web search results typically groups the results returned by the search engines based on their meaning into a hierarchy of labeled clusters, also called categories [8].

The interest of this paper is developing a Web clustering engine for the polysemy ambiguous unigram search keyword that outperforms the existing methodologies. The

polysemy keyword typically having different meaning in each context, demands knowledge to identify the appropriate word sense. The different meaning in each context is referred as sense (S) or concept in this paper. For example, the term Jaguar refers to Jaguar cars concept in the context of automobiles, Mac OS concept in the context of technology, large cat concept in the context of animal and so forth.

Web clustering engines technology has matured to have even commercial systems, however the cluster hierarchies built are far from being perfect for a given polysemy keyword. Even in single level, the web clustering engines result in additional irrelevant clusters and misses out few relevant clusters. This leads to lack of predictability of clustering outputs. For example, when the term puma is searched in some of the most popular search engines, they categorize shoes, logo, cricket bats, swede, animal, footwear, store and few others. Though they have listed a number of categories, the search engines are still missing out a few which the user might be looking for. For instance, the user might be interested in information related to puma village or puma language which is not categorized by most of the popular search engines, to the best of our knowledge. The results produced by the web clustering engines are sometimes inconsistent. The results grouped in a cluster are not always relevant to the cluster label. The quality of the cluster label is another concern for the web clustering engines. If the label of the cluster is ambiguous the entire cluster is likely to be ignored by the user group.

The objective of this work is to build a novel web clustering engine (named SRCluster) that overcomes these deficiencies. The experiment conducted based on the description-aware algorithm (STC) and description-centric algorithm (Lingo) has shown that the quality of the clustering can be further improved (refer to Table 1 under Section 7.2 (a)). SRCluster is our attempt to build a clustering engine that considers the semantics of the search result, unlike the description-aware and description-centric algorithms, and produce a better result.

SRCluster identifies the possible concepts and the label for the given polysemy keyword based on the external knowledge resource - Wikipedia. We have considered each Wikipedia article representing a concept. The description given for each of the search result by the search engines is referred as result snippet. The search result is compared with the concepts identified from Wikipedia to determine the most relevant concept. The search result is grouped to the concept with maximum score.

To evaluate the performance of the proposed method, we conducted the experiment on AMBIENT dataset. The experimental results on AMBIENT dataset [4] show that the inconsistency and the lack of predictability of clustering outputs has been improved using SRCluster. The result shows that the label identified for each concept is more meaningful.

The main contributions of the article are as follows.

a. We developed a methodology that clusters the search results snippet better by measuring the content similarity between the result snippet from the search engines (termed as result feature vector) and the concepts identified from Wikipedia (concept feature).

b. We extended Lesk algorithm for measuring the similarity. We measured the $Score_{SRLesk}$ for each of the search result against the concepts (from Wikipedia) to determine the similarity. The search result is clustered to the concept feature with the maximum $Score_{SRLesk}$. The algorithm has outperformed when compared with the competing Web clustering technologies.

c. We have identified that the search result snippet best represents its corresponding Web document for clustering the document to its relevant concept.

d. The experimental result has produced an evidence for the fact that the introductory section alone of the Wikipedia articles can best represent the corresponding concepts.

The rest of the paper is organized as follows: Section 2 describes the related works. Section 3 briefs the overview of web clustering engine. The structure of Wikipedia is explained in Section 4. Section 5 explains our clustering algorithm ScoreSRLesk which is an extension of the traditional Lesk approach. Section 6 explains the SRCluster framework in detail. In section 7 the experiments and their results are discussed. Section 8 has the conclusion of the work.

2. Related Works

2.1 Web Clustering Engines

Web clustering engines group the result set from the search engine based on their meaning. [Claudio et al 2009] has a detailed survey of the various clustering engines algorithms. The algorithms are divided into 3 categories namely data-centric, description-aware and description-centric. We have summarized each of these categories here, as they are detailed in [Claudio et al 2009].

The data-centric algorithms are found to be proven technique targeted for numeric data. Scatter/Gather, Lassi, WebCat, AIsearch are some of the data-centric algorithms. The algorithms typically derives the label for the cluster from its feature vector (or centroid), which is insufficient from the user perspective, making these algorithms incapable to label and describe the cluster with something sensible for human. The description-aware algorithm concentrates on construction of cluster descriptions which are human interpretable. Suffix Tree Clustering (STC), SnakeT are description-aware algorithms. The setback of the description-aware algorithms is that the clustering precedes and dominates the labeling procedure. The description-centric algorithms are designed to take both the quality of clustering as well description into account. The quality of the description precedes the allocation of the document to the cluster. Lingo, SRC and DisCover are the popular description-centric algorithms. The drawback of these algorithms is that the clusters generated are based on the search results. They may not correspond to the user's interest and hence the label may not be very meaningful for the user, as it is generated based on the contents within the cluster. The algorithm does not consider the semantics of the features in order to cluster as well label them.

2.2 Word Sense Disambiguation

Word sense disambiguation (WSD) is the ability to identify the different meanings of word which is essential to cluster the search results.

[22] has the survey of various approaches of Word Sense Disambiguation systems. The methodologies are classified into supervised, unsupervised and knowledge-based. Supervised WSD methodologies perform better over the other approaches. The setback of the supervised WSD approaches is the non-availability of the large training corpora. Building the large training corpora demands a huge number of person-years of human effort. The unsupervised WSD approaches overcome the knowledge acquisition bottleneck by considering that the same sense of word would have similar neighboring words. The unsupervised WSD approaches, however is more meant for word sense discrimination. The knowledge based approaches exploits external knowledge resources to infer the senses of words in context.

Though the performance is lower than the supervised approaches, they have a wider coverage. The knowledge resources are increasingly enriched.

More the knowledge available, better the performance of the knowledge-based approaches. We believed that more knowledge is available in Wikipedia to perform better. Hence, we adopted the knowledge-based methodologies for the word sense disambiguation for the given polysemy word.

The knowledge-based approaches are classified as selectional preferences, structural approaches and overlap of sense definitions. The selectional preferences are constraints that define the semantic appropriateness of the association provided by a word-to-word relation within the sentence. The approach however has not been found to perform better when compared to the other knowledge-based approaches. The structural approaches are classified further into similarity-based and graph-based. The structural approaches are based on the computational lexicons (like WordNet), which has the structural approaches to analyze and exploit the structure of concept network. The setback of the approach that uses the structural approaches is they are not always current and up-to-date. The overlap of Sense Definition is a simple approach that relies on the number of terms that overlap between the senses definition of the target words.

We here have considered the overlap of sense definition over structural approaches, as the information made available in the computational lexicons with structural approach is not up-to-date. The knowledge resources Wikipedia, on the other hand has the updated information immediately. For instance, the death of Osama Bin Laden has reflected in Wikipedia on May 2, 2011 at 03.08 [10]. We here have adapted the overlap of sense definitions approach with Wikipedia as external knowledge resource to disambiguate the word sense of the given polysemy word. The overlap of sense definition approach is discussed in section 5.

2.3 Clustering Methodologies with External Knowledge Resource

Most of the traditional text based clustering methodologies are based on the “bag of words” representation considering the term frequency within the set of documents, ignoring the topic knowledge and the semantic relationships between key terms [15]. Two documents on the same topic having synonyms or semantically associated terms might be assigned to different clusters. The traditional methodologies do not include the semantic knowledge for identifying the relationship between the terms. This problem was dealt by using ontology to represent the background knowledge. The problem of utilizing the ontology is that it is difficult to find a comprehensive ontology which covers all the concepts for the given term. Some of the research has opted for WordNet as the source for knowledge resource. WordNet, however has limited coverage and has a lack of effective word-sense disambiguation ability. The manual creation of knowledge resource is an expensive and time consuming effort [18]. In the recent years, Wikipedia has become a proven external knowledge resource for many experiments and hence we adapted Wikipedia for the knowledge. The remaining of this section has briefed some of the clustering approaches that considered knowledge resources for clustering.

[16] proposes a hierarchical algorithm to cluster documents using frequent itemsets with Wikipedia for knowledge. They used generalized closed frequent itemsets to build the initial cluster and utilized outlinks and categories listed in Wikipedia article to enhance the initial cluster generated. The approach assumes the occurrence of the same set of itemsets across the documents. In contrary, the search result to be clustered here has limited number of terms in it and might not have the same set of terms occurring in the result snippet. The terms might

include its synonyms and hence the approach of frequent itemsets may not produce a better result for clustering the search results.

[11] utilizes the Open Directory Project (ODP) as the external knowledge for document clustering. The proposed approach uses the textual description of the ODP nodes and the small representative sample of each site within the cataloged URL in order to increase the volume of training data by several orders of magnitude. It also classifies the individual context (a series of non-overlapping segments like windows of words, sentences, paragraphs and others) within the document which performs word sense disambiguation and thus resolves the polysemy of the term. However it is challenging to identify the non-overlapping segments for various polysemy senses within the document. In contrast our approach considers the entire Wikipedia article and the features within it to resolve the disambiguation better.

[15] build the "concept thesaurus" as a phrase index which includes the semantic relations like synonym, hypernym and associative relation for the title of each Wikipedia article. The terms in the text document are mapped to the concept thesaurus, enhancing the traditional content similarity measure for text clustering to consider the semantics. In this methodology "redirect" hyperlink to a specific concept (Wikipedia title) from other Wikipedia pages and the hyperlinks to other concepts are considered as source of synonym. It considers the disambiguation page that presents various possible meanings for a term as source for polysemy. It identifies the hypernym from the "is a" relationship within category links. The approach ignores the terms with semantic relationship other than synonym, hypernym and polysemy. It does not consider the hyponymy, meronymy, holonymy, troponymy and others. [Pu et al, 2008] like [15], considers only the synonym, hypernym, polysemy among the others.

The approach by Rada Mihalcea [21] uses Wikipedia as a source of sense annotations for word sense disambiguation and to train accurate sense classifier for the given term. The methodology builds the sense tagged corpus by extracting the paragraph that has the ambiguous term (including the Wikipedia disambiguation page), collect the possible label for the term and manually mapping the label to its WordNet sense. The approach involves manual processing. In our approach, we propose to utilize the sense defined within the Wikipedia disambiguation page and to consider the anchor text of the in-links to the Wikipedia article about a particular sense, avoiding the manual process of mapping the label to its WordNet sense.

2.4 Labeling the Cluster

The label of the cluster is directly presented to the users which should be meaningful, readable and relevant. Only little work has focused on labeling the resulting cluster of documents, in spite of its importance.

Clustering algorithms uses the term frequency to identify the label of the cluster. [7] labels the cluster with the most frequent words in the cluster. [20] proposed statistical method for identifying the label. They used $\chi 2$ tests of independence to remove non-descriptive terms and the collection frequency of the terms in order to identify the label. [13] uses the information gain for labeling the cluster. The candidate labels are extracted from the content of the web page, anchor text and extended anchor text. The approach found that labels extracted from anchor text and extended anchor text provides better description, however with an overhead of extracting the link structure of the WWW resulting with higher cost. [19], [26] considers the frequent phrases in the document for labeling the cluster. All these approaches assume that the appropriate label would occur in all the documents that are available within the

cluster, which may not be the reality. The documents often do not contain the terms that describes their category. The approaches also ignored the semantic relationship between the terms.

Some of the approaches attempt to exploit the external knowledge resources like WordNet, Wikipedia for labeling the cluster. [9] investigates the usage of Wikipedia for labeling the clusters. The approach extracts the candidate label as the set of important terms that best represent the concept of the documents within the cluster, and the category and title of the related Wikipedia pages for each of the important terms extracted. The best candidate label is chosen by aggregating the various Judges used. Labeling the cluster with Wikipedia has resulted in favor, especially in collections of documents whose topics are covered well by Wikipedia concepts. The approach however may not produce the expected result for domain specific collections due to their irrelevance to the document's topic.

Few other existing methodologies have demonstrated that categories of Wikipedia articles can describe the document's concept better. The mechanism explained in [12] builds the semantic interpreter that interprets the meaning of a text fragment as a weighted vector of Wikipedia concepts. The semantic relatedness between two texts is determined by comparing their weighted vector of Wikipedia concepts, using cosine metric. In [25], Wikipedia and spreading activation on the Wikipedia's category links graph are used to determine the common concept related to the given set of documents. They have shown that it is possible to predict concepts common to the documents, by utilizing Wikipedia as ontology.

We believe that the categories, hypertext from the disambiguation page and the title of Wikipedia articles can describe the concept of the document better. Hence, we utilized the category and the hypertext within the disambiguation page along with the title of Wikipedia article to label the cluster as explained in the concept identifier & labeller component of SRCluster in Section 6.

3. Overview of Web Clustering Engine

Web Clustering Engine is a post-retrieval clustering that clusters the search results retrieved for the broad topic. The majority of Web queries are informational type, expecting relevant information for the given broad topics [3], [23]. The output of the Web clustering engines ensures fast subtopic retrieval, quick topic exploration within unknown topics, and easy identification of relevant search results within the broad topic.

The components of the Web Clustering Engines are explained in [8]. Figure 1 has the generalized components of the Web search clustering engines.

Search Keyword: The system provides an interface to accept the search keyword from the user. In SRCluster, the input will be a 1-gram polysemy word.

Search Result Acquisition: The component accepts the search keyword as input. It allows us to configure the number of search results to be extracted from various search engines and the list of search engines from which the results should be extracted. The component extracts and stores the search results which include the URL pointing to the document, title, and snippet. The component preprocesses the search results. Typically the search results would be converted into a sequence of features through language identification, tokenization, removal of stop words, stemming, but is not restricted to these steps.

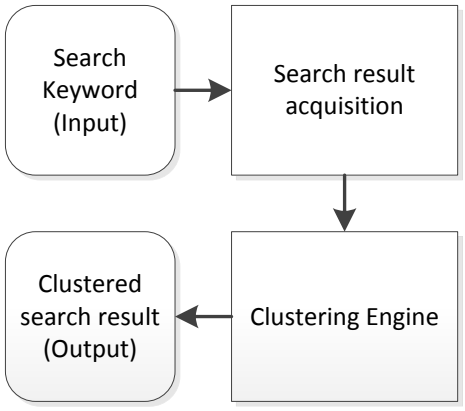

Figure 1. Generalized Components of Web Clustering Engine

Clustering Engine: The clustering engine component converts the preprocessed search results to a format suitable for the clustering algorithm. It extracts the features and provides them as input to the clustering algorithm within. The clustering algorithm would build the cluster and identifies the label that best describes each cluster.

Clustered Search Result: The component presents the result to the user. The results are presented in rich visual interface with the category listed and, when expanded the corresponding search results are displayed to the user.

SRCluster focuses on the clustering engine component of the Web clustering engine.

4. Knowledge Resource of SRCluster

SRCluster considers Wikipedia as its knowledge resource. Wikipedia is considered to be the common reference for any topic and search. Wikipedia has articles for each topic. The topic which has more meaning (referred as concept or sense) has a separate disambiguation page. The disambiguation page has the list of categories to which the topic belongs. Each category has the list of links referring to the Wikipedia article along with the brief about the article in few words, it is referring to. http://en.wikipedia.org/wiki/Jaguar_(disambiguation) is the disambiguation page, a non-article page that has the list of various categories and a list of links to the corresponding Wikipedia article that covers each of them. Figure 2 has the list of possible categories for the term 'Jaguar'.

Figure 2. Categories for 'Jaguar'

5. Overlap of Sense Definition

5.1 Traditional Lesk Approaches

The overlap of sense definitions is a simple knowledge-based approach proposed by Michael Lesk [17]. The approach automatically identifies the sense of the word which it is intended for (in written English) using machine readable dictionaries. The approach determines the intended sense of the given word by counting overlaps between dictionary definitions of various senses (gloss(S)) and the words in the definition of nearby words (context(w)). Formally, for a given word w, its intended sense is identified by measuring $score_{LeskVar}$.

$$score_{LeskVar}(S) = \left| context(w) \cap gloss(S) \right|$$

where S being each possible sense of the word w, context(w) being the bag of content words in the context window around the target w.

Banerjee and Pedersen [1] extended the Lesk algorithm to include the concepts that are related through explicit relations in the dictionary (hypernymy, meronymy and others) of each sense. Formally, for each sense S of the target word w, the scoreExtLesk is calculated as follows.

$$score_{ExtLesk}(S) = \sum_{s':s \xrightarrow{rel} s' or s \equiv s'} \left| context(w) \cap gloss(S') \right|$$

where S being each possible sense of the word w, context(w) being the bag of content words in the context window around the target w, gloss(S') being the bag of words of a sense S' where S' is either S or related to S through a relation *rel*, which is configurable. Banerjee and Pedersen referred to WordNet as external knowledge source for the sense.

5.2 SRLesk : Extended Lesk Approach for SRCluster

The clustering algorithm of the SRCluster being proposed here extends the Lesk algorithm further to identify the appropriate concept of the search result. Each Wikipedia article represents a concept. The number of unique terms and their frequencies are more in Wikipedia articles.

The extended Lesk algorithm for SRCluster is termed as SRLesk. The context here in SRLesk is the result snippet returned by the search engines. The gloss is derived from the result snippet. The SRLesk considers the overlapping terms between the search result and the concept as well the weight of the overlapping terms among the concepts. The approach determines the $score_{SRLesk}$ for the search result with the possible concept of the given search keyword.

$$score_{SRLesk}(concept_j) = \frac{1}{\left|N_i\right|} \sum_{\{terms_k \mid \exists terms_k \in result_i \wedge terms_k \in concept_j)\}} \left(\frac{tf_{k,j}}{tf_k} \right)$$

where N_i being the number of terms in the $result_i$, $terms_k$ being the terms that overlap in the $result_i$ and $concept_j$, $tf_{k,j}$ being the term frequency of the each $term_k$ within the $concept_j$, tf_k being the term frequency of the term k across the Wikipedia concepts. The Wikipedia concept with the highest SRLesk score is considered to be the intended sense of the $result_i$.

6. SRCluster

6.1 Overview

SRCluster Web clustering engine accepts unigram polysemy search keyword and presents the search result as clusters. SRCluster categorizes the search results for the given polysemy keyword into different concepts based on Wikipedia articles. It identifies the label for each of the concept. It first determines the semantically related candidate terms for each concept from its corresponding Wikipedia article. The search result acquisition component extracts the search results from the popular search engines simultaneously. The clustering engine component preprocesses the search results and generates the feature vector, termed *result feature*. The clustering algorithm determines the overlap of the result feature with the related terms for identifying the semantically related concepts. The search result is clustered to the concept with which the overlap is higher.

6.2 Architecture

Figure 3 provides an illustration of the overview of the SRCluster. The user enters the unigram polysemy keyword in search keyword component. The two components namely result extractor and concept identifier & labeller are invoked simultaneously by the engine. The concept identifier and labeller component identifies the possible concepts for the given keyword and identifies a label for each of the concepts identified. The concept feature builder builds the concept feature. The concept feature has the list of terms that occur across the Wikipedia articles with its corresponding concept and the term frequency within the concept. The result extractor component extracts the search result from the popular search engines. The result feature builder extracts the features vector from each of the search result and stores as the result feature in the repository. The SRLesk clustering algorithm clusters the result feature and its Web document of the search result to the relevant Wikipedia concepts available in the concept feature.

The functionality of each of the component is explained below.

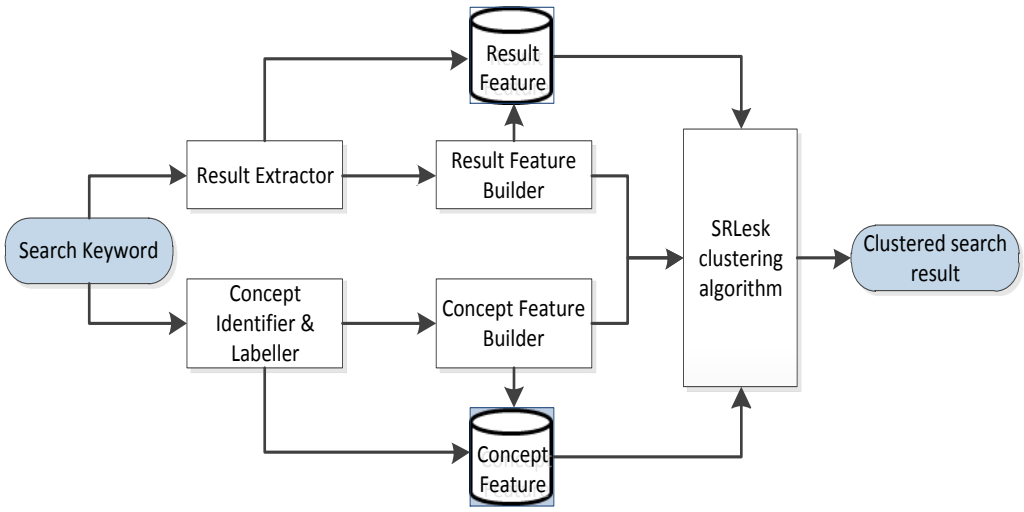

Figure 3. Overview of SRCluster

6.3 Result Extractor

The result extractor component of the SRCluster submits the given keyword to the popular search engines – Yahoo! [28] and Bing [2] search engines. The result from Yahoo! and Bing is fetched by using application programming interfaces (API) provided by these engines. The results extracted from the search engine would be comprised of URI, title and a snippet that summarizes the document it is referring to.

6.4 Result Feature Builder

The result feature builder component is responsible for preprocessing each of the search results and building the feature vector for the search result. The component tokenizes the search result by considering the space as the delimiter. The texts from the WWW are often plagued with noise. The noises are removed by considering the special characters as delimiter for tokenization. The common, unimportant terms within the text are termed as stop words. The component removes the stop words from the search result. Stemming is performed on the text to remove the inflectional suffixes and to retain the base of each term. The feature vector is created for the search result by the component with the cleaned and stemmed terms. The feature vector is termed as *Result feature*.

6.5 Concept Identifier & Labeller

The search keyword is provided as input to the concept identifier & labeller component. The component identifies the possible concept for the given keyword based on the disambiguation page in Wikipedia and a suitable label for each concept.

SRCluster engages the disambiguation page of Wikipedia to identify the possible concepts for the given keyword. Each hyperlink pointing to a Wikipedia article about the keyword in the disambiguation page represents a specific concept. An extract of the disambiguation page from Wikipedia for "Jaguar" is shown in figure 4. Jaguar Cars hyperlink within the disambiguation page that refers to the URL http://en.wikipedia.org/wiki/Jaguar_Cars, is the Wikipedia article about Jaguar pertaining to car concept under the category companies.

Figure 4. An Extract of Jaguar_(disambiguation) from Wikipedia

SRCluster identifies each hyperlink under the various categories within the disambiguation page as a concept for the given polysemy term. SRCluster considers the category under which the link occurs, the hypertext and its surrounding text for labeling the concept. The title of

the article is considered if it does not match the hypertext and its surrounding text. The list of concepts along with the label is stored. For instance, the label for Jaguar Cars is defined as Companies : Jaguar Cars, a British manufacturer.

6.5 Concept Feature Builder

The objective of the concept feature builder is to enable concept selection per attribute (term) in contrary to the existing attribute selection per concept.

Each hyperlink within the disambiguation page of Wikipedia points to the article which is considered to be the general reference for the specific topic. SRCluster considers these articles to have the terms that are semantically related to the specific concept. Once the system identifies the URL of the hyperlink within disambiguation page, the system extracts the content of the article and preprocesses the text. The preprocessing of the text includes tokenization, removal of stop words, stemming. The remaining tokens are identified to be the terms that are semantically related to the concept. A feature vector termed as *concept feature* for the terms that are semantically related to each of the identified concepts from Wikipedia, is generated based on the processed tokens.

The concept feature is generated to improve the performance of the system. It has the concept that represents the Wikipedia article (WikiDocId) and the frequency of the term that occurs within the article, for each term.

$$Term \rightarrow \{(Concept, Weight)\}$$

The component adds the processed term from the Wikipedia article to the concept feature with the corresponding concept. While adding the term to the concept feature, it includes the concept (which is the Wikipedia article id) and the term frequency within Wikipedia article. If the term is already available in the concept feature, the mapping details (concept, frequency) are added to the existing list.

6.6 SRLesk Clustering Algorithm

The component accepts the result feature from result feature builder and concept feature from concept feature builder as inputs. SRLesk algorithm explained in section 5 is implemented to cluster the result feature.

The SRLesk scores for a specific result with all its possible concepts are measured. The higher the SRLesk score for a result feature with a concept, the more the similarity between the search result and the concept. The specific search result is thus clustered to the concept which has the maximum SRLesk score.

$$score(result_i) = \max_j \left(score_{SRLesk}(concept_j) \right)$$

where $Score(result_i)$ represents the score for the search result, $score_{SRLesk}(concept_j)$ represents the SRLesk score for the search result ($result_i$) with respect to $concept_j$. The concept ($concept_j$) with the maximum SRLesk score is considered to be the cluster to which the search result ($result_i$) belongs and grouped.

6.7 Clustered Search Result

SRCluster presents the clustered search result to the user. The result includes the category along with the brief in the format of <CATEGORY> : <BRIEF ABOUT THE CONCEPT>. When it is expanded, the search results clustered under the concept are listed for the reader.

7. Experiments

7.1 Background Information

i. **Wikipedia Dataset:** Wikipedia releases its dumps periodically. The latest release of the dumps can be downloaded from http://download.wikipedia.org. In our experiment, the static HTML dumps for English language has been utilized.

ii. **Clustering Dataset:** The experiment has been conducted with the AMBIENT [Carpineto et al, 2008] dataset. It is a dataset designed for evaluating the subtopic information retrieval. It consists of 44 topics which are selected from Wikipedia disambiguation page. Each topic has a set of subtopics. Each subtopic has a set of documents that comprises of URL, title and snippet, retrieved from a Web search engine as of January 2008. They are annotated with subtopic relevance judgments.

iii. **Evaluation Criteria:** Cluster quality is evaluated by two metrics, purity [30], Entropy. Purity measures the coherence of a cluster, the extent to which each cluster contains documents from primarily one class. The purity of the cluster C (Purity(C)) is measured as a weighted average of the purity of cluster Ci, (P(Ci)).

$$Purity(C) = \sum_{i=1..k} \left(\frac{n_i}{n} P(C_i) \right)$$

$$P(C_i) = \frac{1}{n_i} \max_h (n_i^h)$$

Here n is the number of members in cluster C, n_i is the number of members in cluster C_i, n_i^h denotes the number of members in C_i belonging to the h^{th} class, $max_h(n_i^h)$ being the number of members from the most frequent category in cluster C_i.

Entropy measures the disorder within the clusters. The entropy of a cluster C (Entropy(C)) is measured as weighted sum of entropy of each individual clusters ($E(C_i)$).

$$Entropy(C) = \sum_{i=1..k} \left(\frac{n_i}{n} E(C_i) \right)$$

$$E(C_i) = \sum_{h=1..k} -\frac{n_i^h}{n_i} \log\left(\frac{n_i^h}{n_i}\right)$$

Here n is the number of members in cluster C, n_i is the number of members in cluster C_i, n_i^h denotes the number of members in C_i belonging to the h^{th} class.

7.2 Experiments and Results

The AMBIENT dataset has 44 topics with an average of 17 subtopics under each topic. The topics and its subtopics which do not have any appropriate terms within the search result are considered to be noise and are removed from the dataset. The search result whose snippet is null are removed from the dataset. After the noise removal, there were 20 search results for 350 subtopics under 44 topics. We considered 18 topics from AMBIENT for the evaluation.

In this section, however we have mentioned the results of 9 topics among the 18, as the result pattern was similar.

The open source web clustering engine Carrot2 [5] has been as competing technique for evaluation. Carrot2 has been chosen as the competing technique as it has both the description-aware as well description-centric implementation. STC of Carrot2 is description-aware implementation and Lingo of Carrot2 is description-centric implementation. The AMBIENT data set is provided to Carrot2 for evaluating the purity and entropy of Lingo and STC. The purity of both STC and Lingo has been calculated and the results are shown in Table 1 and Figure 5. The entropy of both STC and Lingo are calculated and the results are shown in Table 2 and Figure 6. From the results, we observe that Lingo clustering algorithm performs better than STC.

Table 1. Purity measure for Lingo, SRC, SRCluster

Topic	Lingo	STC	SRCluster
Aida	0.780	0.030	0.655
Camel	0.550	0.040	0.919
Indigo	0.400	0.000	0.867
Jaguar	0.290	0.290	0.888
Excalibur	0.090	0.000	0.774
Minotaur	0.300	0.220	0.760
Urania	0.348	0.140	0.782
Zenith	0.500	0.000	0.871
Zodiac	0.389	0.000	1.000

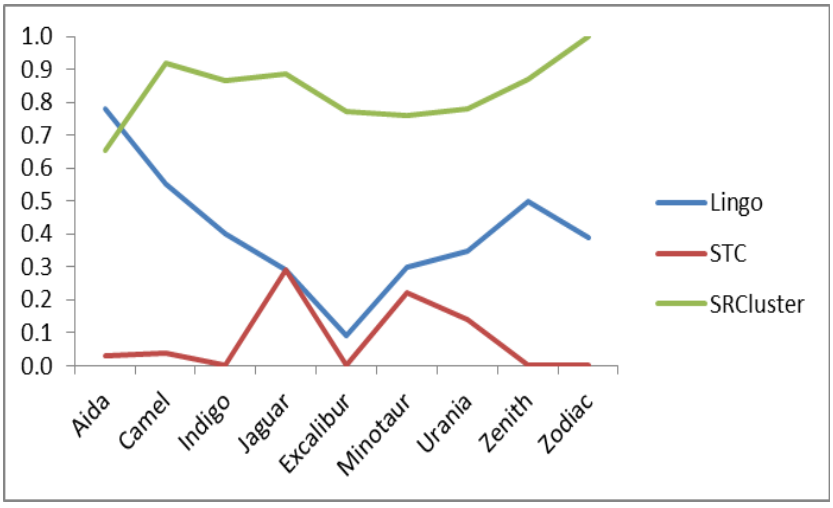

Figure 5. Purity Measure for Lingo, SRC, SRCluster

Table 2. Entropy measure for Lingo, SRC, SRCluster

Topics	Lingo	STC	SRCluster
Aida	0.210	0.449	0.103
Camel	0.142	0.650	0.032
Indigo	0.390	1.000	0.071
Jaguar	0.190	0.495	0.041
Excalibur	0.880	1.000	0.080
Minotaur	0.150	0.400	0.068
Urania	0.229	0.700	0.299
Zenith	0.245	1.000	0.047
Zodiac	0.520	1.000	0.000

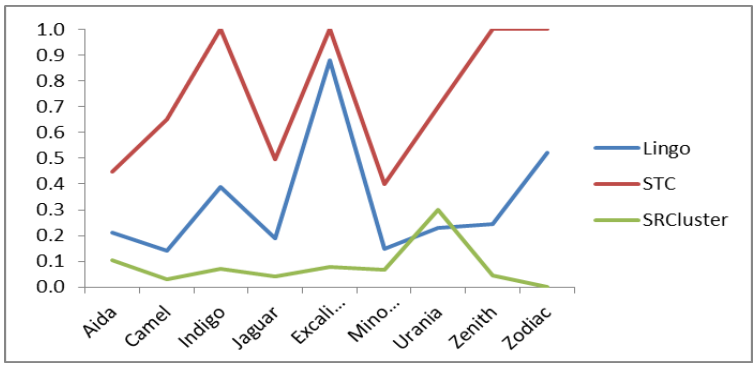

Figure 6. Entropy Measure for Lingo, STC and SRCluster

We then measured the purity and entropy for SRCluster with the same set of search results. The purity and entropy for SRCluster are given in Table 1 and 2. The results are compared with STC and Lingo in Figure 5 and 6. The experimental result shows that the SRCluster produces a better result when compared to STC and Lingo.

Table 1 and 2 shows the purity and entropy measure of SRCluster which considers the whole content of Wikipedia article to determine the candidate terms based on the semantic similarity. Though the approach has produced better results, the result of topic 'Aida' however is not better than Lingo. When we looked into the results snippets and the Wikipedia article for 'Aida', we realized that two of the Wikipedia articles about 'Aida' (namely Aida, the opera by Giuseppe Verdi and Elton John and Tim Rice's Aida) are highly related to each other. The system produced a purity of 0.899 and Entropy of 0.0714 when the result snippets related to these two topics are eliminated for evaluation.

As the results were promising, we implemented variations in building the concept index to determine the better approach that identifies the appropriate set of candidate terms from Wikipedia. The objective of first variation is to evaluate whether to consider the entire content of Wikipedia or only the introductory section of Wikipedia to determine the semantically related candidate terms for building up the concept index. The second variation is to evaluate whether to include the tokens comprised of many words within Wikipedia in order to determine the candidate terms for the concept index. This has been achieved by considering the anchor text (the clickable phrase of the hyperlink) within the Wikipedia article as a single token comprising of many words.

Table 3 and figure 7 shows the purity measure of all the 3 variations of SRCluster being compared.

Table 3. Purity Measure of SRCluster Variations

Topics	Entire Conent	Introductory Section	Tokens with many words
Aida	0.655	0.636	0.630
Camel	0.919	0.903	0.924
Indigo	0.867	0.813	0.831
Jaguar	0.888	0.863	0.852
Excalibur	0.774	0.710	0.714
Minotaur	0.760	0.835	0.792
Urania	0.962	0.971	0.951
Zenith	0.871	0.742	0.814
Zodiac	1.000	1.000	1.000

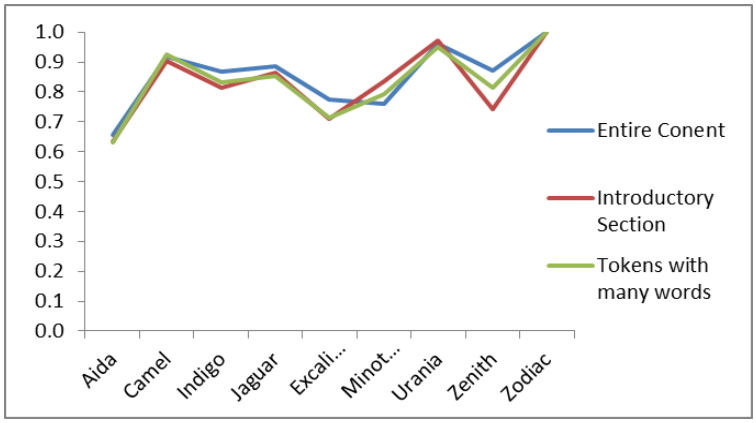

Figure 7. Purity Measure of SRCluster Variations (Entire content, introductory section, tokens with many words)

Table 4. Entropy Measure of SRCluster Variations

Topics	Entire Conent	Introductory Section	Tokens with many words
Aida	0.103340138	0.110866081	0.1192
Camel	0.032	0.038	0.031
Indigo	0.071	0.078	0.073
Jaguar	0.041	0.053	0.050
Excalibur	0.080	0.188	0.157
Minotaur	0.068	0.063	0.065
Urania	0.069	0.009	0.095
Zenith	0.047	0.090	0.059
Zodiac	0.000	0.000	0.000

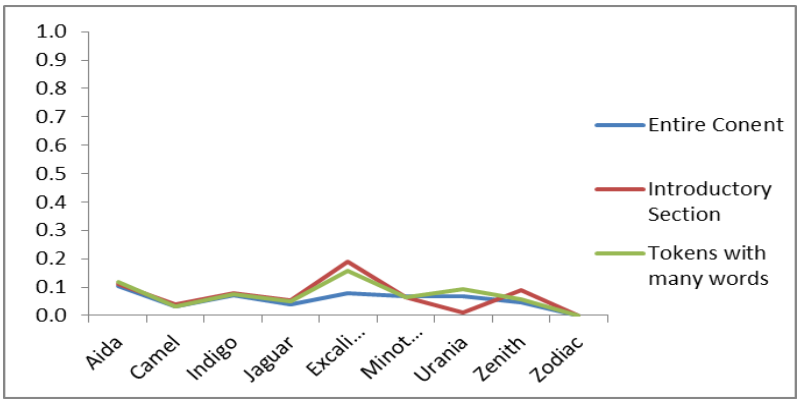

Figure 8. Entropy measure of SRCluster variations (Entire content, introductory section, tokens with many words)

Of the 3 variations, the results in table 3, 4 and figure 7, 8 shows that all the three are close. We then estimated the processing time for each of the variations. The comparison of the processing time of all the 3 variations is shown in table 5 and figure 9.

Table 5. Processing Time of SRCluster Variations

Topics	Entire Conent	Introductory Section	Tokens with many words
Aida	1.09	0.82	1.39
Camel	0.609	0.491	0.716
Excalibur	0.835	0.613	0.923
Indigo	1.329	0.923	1.494
Jaguar	0.773	0.535	0.875
Minotaur	0.819	0.601	0.891
Urania	0.856	0.684	0.911
Zenith	0.601	0.451	0.684
Zodiac	0.799	0.512	0.907

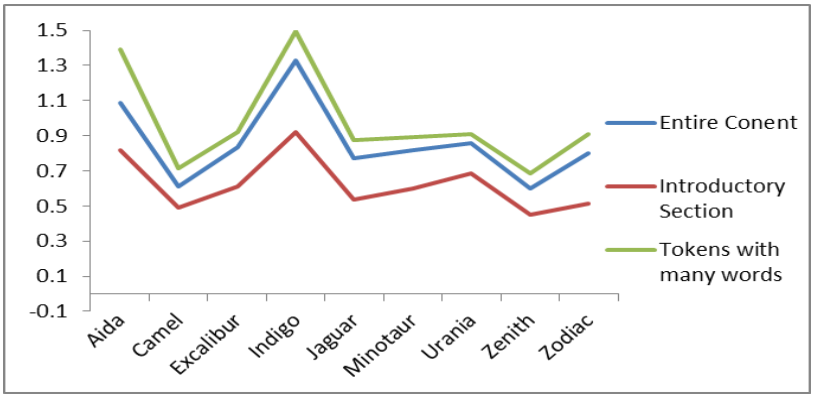

Figure 9. Processing time of SRCluster variations (Entire content, introductory section, tokens with many words)

The result shows that considering the introductory section of Wikipedia to build the candidate terms for the cluster takes lesser time when compared to considering the entire content and identifying the tokens with many words. The purity and entropy of considering only the introductory section of Wikipedia is not producing the best result among the three always. When the processing time is considered the approach is better than the other two.

8. Conclusion

In this work, we present a Web clustering engine that employs the Wikipedia disambiguation page and the articles to produce the result of the web clustering engines. SRCluster, clusters the search results to the corresponding concept. The experimental results shows that the SRLesk has outperformed when compared to the competing methodologies. The SRLesk which is based on concept feature per attribute approach outperforms the description-aware and description-centric methodologies for clustering the search results.

With the results of the SRCluster variation, we conclude that the introductory section of Wikipedia article alone can be considered to identify the semantically related terms for the given keyword. This approach utilizes less memory and, processing time when compared to the other two variations. The system shows that the hypertext and its surroundings can be considered for labeling the clusters with a better quality.

There are two directions in which the work can be extended further. We believe that leveraging Wikipedia for web clustering engine can further be extended for hierarchical conceptual clustering. The scoreSRLesk can be further improved to suit the domain-driven disambiguation.

References

[1] Satanjeev Banerjee and Td Pedersen, "Extended Gloss Overlaps as a Measure of Semantic Relatedness", In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, **(2003)**, pages 805-810

[2] "Bing API", http://msdn.microsoft.com/en-us/library/dd900818.aspx, last accessed on **(2010)** December 18.

[3] Broder A, "A taxonomy of Web Search", ACM SIGIR Forum 36, 2, **(2002)**, pages 3-10.

[4] Carpineto C., Romano G. "Ambient dataset", http://credo.fub.it/ambient/, **(2008)**.

[5] "Carrot2 Clustering Engine", http://search.carrot2.org/stable/search, last accessed **(2011)** March.

[6] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, Introduction to Information Retrieval, Cambridge Press, **(2008)**.

[7] Chuang S., and Chien L., "A practical web-based approach to generating topic hierarchy for text segments". In proceedings of the 20th International Conference on Information and Knowledge Management, **(2004)**, pages 127-136.

[8] Claudio Carpineto, Stanisiaw Osinski, Giovanni Romano and Dawid Weiss, "A Survey of Web Clustering Engines", ACM Computing Surveys, Volume 41, No. 3, Article 17 **(2009)** July, pages 17:1 – 17:38.

[9] David Carmel, Haggai Roitman and Naama Zwerdling, "Enhancing Cluster Labeling using Wikipedia", Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, **(2009)** July 19-23, pages 139 – 146.

[10] Wikipedia, "Death of Osama Bin Laden", http://en.wikipedia.org/w/index.php?title=Death_of_ Osama_bin_Laden&dir=prev&action=history, accessed on **(2011)** May 6.

[11] Evgeniy Gabrilovich and Shaul Markovitch, "Feature Generation for Text Categorization Using World Knowledge", In Proceedings of the Nineteenth International Joint Conference for Artificial Intelligence, Pages 1048-1053, Edinburgh, Scotland, **(2005)**, pages 1048--1053.

[12] E. Gabrilovich, S. Markovitch, "Computing Semantic relatedness using Wikipedia-based explicit semantic analysis", In IJCAI '07, Hyderabad, India, **(2007)**, pages 1606 – 1611.

[13] Glover Eric J, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake, "Using web structure for classifiying and describing web pages", In Proceedings of WWW, ACM Press, **(2002)**, pp 562-569.

[14] Xiaohua Hu, Xiaodan Zhang, Cimei Lu, E.K. Park, Xiaohua Zhou, "Exploiting Wikipedia as External Knowledge for Document Clustering", KDD '09, **(2009)** June 28 – July 1, pages389 – 396.

[15] Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, Zheng Chen, "Enhancing Text Clustering by Leveraging Wikipedia Semantics", SIGIR '08, **(2008)**.

[16] G.V.R. Kiran, K.Ravi Shankar and Vikram Pudi, "Frequent Itemset based Hierarchical Document Clustering using Wikipedia as External Knowledge", International Conference on Knowledge-Based and Intelligent Information Engineering Systems, Wales, UK, **(2010)** September.

[17] Michael Lesk, "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone", In Proceedings of the 5th SIGDOC, **(1986)**, Pages 24-26.

[18] NG. T.H., "Getting serious about word sense disambiguation", In Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: why, What and How? Pages 1 -7.

[19] S. Osinski and D. Weiss. "A concept-driven algorithms for clustering search results". IEEE Intelligent Systems, 20 (3), **(2005)**, pages 48-54.

[20] Alexandrin Popescul, Lyle H. Ungar, "Automatic Labeling of Document Clusters", Unpublished manuscript, available at http://citeseer.nj.nec.com/popescu100automatic.html, **(2000)**, pages 1 – 16.

[21] Mihalcea R, "Using Wikipedia for Automatic Word Sense Disambiguation", in Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL 2007), Rochester, **(2007)** April, pages 196 – 203.

[22] Navigi R., "Word Sense Disambiguation: A Survey", ACM Computing Surveys, vol. 41, no. 2, Article 10, **(2009)** February, pages 10:1 – 10:69.

[23] Rose D.E., Levinson. D, "Understanding user goals in Web search", In Proceedings of the 13th International Conference on Word Wide Web, ACM Press, **(2004)**, page 13-19.

[24] Stein, Benno, and Sven Meyer zu Eissen, "Topic identification: Framework and Application". In Proceedings of International Conference on Knowledge Management, **(2004)**, pages 353-360.

[25] Syed Z. S., Finin T., and Joshi A.,"Wikipedia as an Ontology for Describing Documents', in ICWSM '08, **(2008)**, pages 136 – 144.

[26] P. Treeratpituk and J. Callan, "Automatically labeling hierarchical clusters", In DG.O '06, New York, USA, **(2006)**, ACM, pages 167 – 176.

[27] Wang X., Zhai C., "Learn from Web Search Logs to Organize Search Results", In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, **(2007)**, pages 87 – 94.

[28] "Yahoo! Search BOSS Web Services", http://developer.yahoo.com/search/, last accessed on **(2010)** December 24.

[29] "Zemanta Developer network", http://developer.zemanta.com/, last accessed **(2011)** March.

[30] Zhao Y and Karypis G, "Criterion functions for document clustering: experiments and Analysis", Technical Report, Department of Computer Science, University of Minnesota, **(2010)**.