IMPROVING SEARCH RESULT CLUSTERING BY INTEGRATING SEMANTIC INFORMATION FROM WIKIPEDIA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÇAĞATAY ÇALLI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2010

Approval of the thesis:

IMPROVING SEARCH RESULT CLUSTERING BY INTEGRATING SEMANTIC INFORMATION FROM WIKIPEDIA

submitted by **ÇAĞATAY ÇALLI** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** ________________

Prof. Dr. Göktürk Üçoluk
Supervisor, **Computer Engineering Department, METU** ________________

Dr. Onur Tolga Şehitoğlu
Co-supervisor, **Computer Engineering Department, METU** ________________

Examining Committee Members:

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering Dept., METU ________________

Prof. Dr. Göktürk Üçoluk
Computer Engineering Dept., METU ________________

Dr. Onur Tolga Şehitoğlu
Computer Engineering Dept., METU ________________

Dr. Cevat Şener
Computer Engineering Dept., METU ________________

Dr. Meltem Turhan Yöndem
____

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name:	ÇAĞATAY ÇALLI
Signature :	

ABSTRACT

IMPROVING SEARCH RESULT CLUSTERING BY INTEGRATING SEMANTIC INFORMATION FROM WIKIPEDIA

Çallı, Çağatay
M.S., Department of Computer Engineering
Supervisor : Prof. Dr. Göktürk Üçoluk
Co-Supervisor : Dr. Onur Tolga Şehitoğlu

September 2010, 102 pages

Suffix Tree Clustering (STC) is a search result clustering (SRC) algorithm focused on generating overlapping clusters with meaningful labels in linear time. It showed the feasibility of SRC but in time, subsequent studies introduced description-first algorithms that generate better labels and achieve higher precision. Still, STC remained as the fastest SRC algorithm and there appeared studies concerned with different problems of STC.

In this thesis, semantic relations between cluster labels and documents are exploited to filter out noisy labels and improve merging phase of STC. Wikipedia is used to identify these relations and methods for integrating semantic information to STC are suggested. Semantic features are shown to be effective for SRC task when used together with term frequency vectors.

Furthermore, there were no SRC studies on Turkish up to now. In this thesis, a dataset for Turkish is introduced and a number of the methods are tested on Turkish.

Keywords: Search Result Clustering, Document Clustering, Text Mining

ÖZ

WIKIPEDIA’DAKI ANLAMSAL BİLGİYİ KULLANARAK ARAMA SONUCU
KÜMELEMENİN GELIŞTİRİLMESİ

Çallı, Çağatay
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Göktürk Üçoluk
Ortak Tez Yöneticisi : Dr. Onur Tolga Şehitoğlu

Eylül 2010, 102 sayfa

Sonek Ağacı Kümeleme (SAK), anlamlı isimlere sahip, örtüşebilen kümeleri lineer zamanda üretmeye odaklanan bir arama sonucu kümeleme (ASK) algoritmasıdır. SAK, ASK’nin uygulanabilirliğini göstermiştir. Ancak sonraki çalışmalar daha anlamlı küme isimleri üreten, daha hassas algoritmalar ortaya koymuştur. Buna rağmen, SAK en hızlı sonuç kümeleme algoritması olarak kalmış ve SAK’ın problemleriyle ilgili çalışmalar yapılmıştır.

SAK’ı geliştiren başka çalışmaların aksine, bu tezde hatalı küme isimlerini filtrelemek ve birleştirme fazını geliştirmek amacıyla küme isimleri ve dökümanlar arasındaki anlamsal bağlantılardan faydalanılmıştır. Bu bağlantıları belirlemek için Wikipedia kullanılmış ve anlamsal bilgiyi SAK’a entegre etmek için yöntemler önerilmiştir. Terim frekans vektörleriyle beraber kullanıldığında anlamsal özelliklerin ASK’de etkili olduğu gösterilmiştir.

Ayrıca, şimdiye kadar Türkçe için bir ASK çalışmaşı yapılmamıştır. Bu tezde, Türkçe için bir veri seti oluşturulmuş ve yöntemlerin bazıları test edilmiştir.

Anahtar Kelimeler: Arama Sonucu Kümeleme, Döküman Kümeleme, Metin Madenciliği

To the future

ACKNOWLEDGMENTS

I would like to thank Dr. Meltem Turhan Yöndem for her invaluable contributions, advice and constant support during the course of this thesis.

I would like to thank my supervisors Prof. Dr. Göktürk Üçoluk and Dr. Onur Tolga Şehitoğlu for their mentoring and their eagerness to point me in the right direction. I really enjoyed every moment of brainstorming with them.

I would also like to thank Prof. Dr. İsmail Hakkı Toroslu and Dr. Cevat Şener for their valuable comments and suggestions. I would like to thank Evgeniy Gabrilovich for his explanations and support on Explicit Semantic Analysis.

I would like to thank all staff at the Department of Computer Engineering at METU. This department is one of the best places to work because most times we do not feel like we are actually working. I would like to offer thanks to Atıl İşçen, Umut Eroğul, Çelebi Kocair and Gencay Evirgen. I had a great time while working with them in the same room and I'm grateful for their friendship.

My parents, my brother and his family deserve my deepest gratitude for showing their never-ending love, trust and understanding through the times I need the most. Thanks to all friends near or far, you were always kind enough to accept my apologies. You probably know I did not and I cannot forget you. Even when you were not aware of it, you were a part of this work.

Finally, I would love to express my appreciation to Ezgi Erdoğan. I cannot think of achieving anything in this world without your support. This thesis, like every other thing, would not be completed without your help, patience and caring through hard times. I will always be grateful for your trust and understanding.

I have been partially funded by Turkcell Akademi and TBD (Türkiye Bilişim Derneği) under ”gnçtrkcll Yüksek Lisans Burs Programı” and TUBITAK BIDEB 2210 National Graduate Scholarship Programme for M.Sc. students during my graduate studies.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Since their early days, search engines presented their results with a list of URLs and short descriptions (snippets), ranked by relevance. This approach turned out to be effective for navigational queries. A navigational query is about a website that the user knows about. In this case the user tries to reach a specific website but does not know or remember the location. Broder (2002) [4] defines two more types of queries other than navigational queries - informational and transactional. Recent studies [4][37] show that only about 10-20% of search queries are navigational. In informational queries, the user assumes the information exists on one or more websites and tries to reach that information. These type of queries are indeed dominant with a ratio above 80% .

Current search engines respond this informational need by supplying images, video results, categorical links and other search queries related with the current query. These enhancements compensate for the disadvantages of using a ranked list in discovery tasks. However, as Ferragina and Gulli (2008) [21] emphasized, the users are lazy, compose short, ill-defined queries and look mainly at the top 10 results. For a short, ambiguous user query, the ranked result list is dominated by the popular meaning, and alternative meanings get buried. For example, with the ambiguous query ”amazon”, the results will be dominated by links to an e-commerce company - *Amazon.com*. ”Amazon rainforest”, ”Amazon River” and ”Amazons”, the nation of female warriors, are equally important but get under-represented.

In order to tackle these problems with informational queries, third-generation search engines offer various post-search tools to enhance the query or results. These tools include query suggestion/refinement, ontology mapping and result clustering.[21] Result clustering was introduced with the Scatter/Gather browsing paradigm by Hearst (1996) [33]. With result clustering, users can cope with the immense amount of information from the Web since search results are grouped under meaningful cluster labels. Like Scatter/Gather, there are numerous clusterers which also build a hierarchy between clusters. Vivisimo [72] is probably the best known example of such clustering search engines, currently offering its result clustering service at another website [85].

One of the key advantages of result clustering is the fact that a user can observe all related meanings of the query at once, despite the synonymous or polysemous nature of the phrase. A proper cluster listing also conveys information about the importance of topics. One may think that the same principles for document clustering work for achieving a good result clustering performance. However, the input of search result clustering task is snippets containing no more than one or two sentences. This limited input reduces tolerance of algorithms to noise and traditional document clustering algorithms perform bad.

Early result clustering algorithms used the Bag-Of-Words (BOW) approach, utilizing term frequency vectors, as in traditional document clustering. However, subsequent algorithms proved to be more successful by considering each phrase repeating in multiple snippets as potential clusters. Suffix Tree Clustering (STC) is such an algorithm. Then algorithms such as Lingo utilized latent semantic analysis (LSA) to use abstract concepts hidden in the data as clusters. Ways to integrate explicit information from knowledge bases (e.g. WordNet, Open Directory Project, Wikipedia) into clustering algorithms are also investigated. As a knowledge base (KB), Wikipedia is popular among these studies [63][36] because of its properties such as cross-linked, high quality article content and wide coverage.

The aim of this thesis is to investigate the whether it is possible to improve a popular SRC method, Suffix Tree Clustering (STC), by evaluating a previous study about STC from Wang et al.[73] and by using explicit semantic information from Wikipedia. Figure 1.1 presents the workflow of our research. We identified problems with the method of Wang et al. and investigated several ideas to tackle these problems, including quality controls (Wang-AvgIntra), connectivity checks (Wang-ConnComp, Wang-IndChild) and semantic filtering (Wang-NWD).

Effectiveness of explicit semantic features is a primary issue that we want to investigate. In Figure 1.1, WLM-Filter, WLM-Merge and WLM-FilterMerge nodes denote the semantic

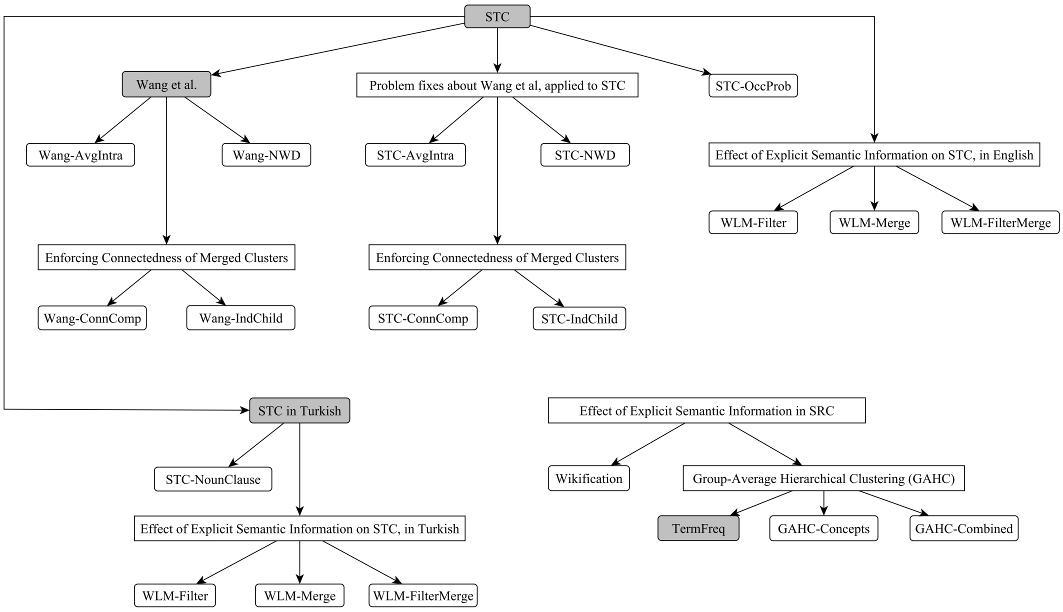

STC
Wang et al.
Problem fixes about Wang et al, applied to STC
STC-OccProb
Wang-AvgIntra
Wang-NWD
STC-AvgIntra
STC-NWD
Effect of Explicit Semantic Information on STC, in English
WLM-Filter
WLM-Merge
WLM-FilterMerge
Enforcing Connectedness of Merged Clusters
Enforcing Connectedness of Merged Clusters
Wang-ConnComp
Wang-IndChild
STC-ConnComp
STC-IndChild
STC in Turkish
Effect of Explicit Semantic Information in SRC
STC-NounClause
Wikification
Group-Average Hierarchical Clustering (GAHC)
Effect of Explicit Semantic Information on STC, in Turkish
TermFreq
GAHC-Concepts
GAHC-Combined
WLM-Filter
WLM-Merge
WLM-FilterMerge

Figure 1.1: Organization of investigated approaches (gray nodes are baseline methods)

controls applied on STC in this regard. To test whether semantic information improves SRC in general, we also implemented term-frequency based clustering (TermFreq) as baseline and compared methods using only concepts (GAHC-Concepts) and both term frequencies and concepts (GAHC-Hybrid). We also compared clustering by wikification (Wikification), an obvious method of using concept information, against other methods.

With the intention of initiating SRC study in Turkish, we introduced a SRC dataset for Turkish. Thinking that meaningful cluster labels could be identified easier in Turkish, we investigated the effectiveness of allowing only noun-clauses (STC-NounClause in Figure 1.1). We also thought that our explicit semantic approach to SRC could also be effective for Turkish, since Wikipedia concept space have similar properties for English and Turkish. In order to analyze the effect of semantic features in Turkish, we repeated the methods we applied for English by using Wikipedia Turkish edition. It should be noted, however, that Wikipedia Turkish edition is still in its infancy and lacks a great deal of Turkish information.

1.2 Outline of the Thesis

The organization of the thesis is as follows:

In Chapter 2, a survey of related work on result clustering is given. Chapter 3 gives detailed background information about the tools and topics related with this thesis. In Chapter 4, the data used for the experiments and the methods used are described in detail.

In Chapter 5, the experimental results and the discussions about these results are provided.

Finally, Chapter 6 summarizes the thesis and gives the conclusions. Possible improvement ideas to the applied methods are given in the future work section.

CHAPTER 2

LITERATURE SURVEY

Search result clustering (SRC) is a subfield of document clustering. One may think that existing full-text classification and grouping methods encapsulates this problem. However there are several differences which turn it to a separate problem requiring new methods or enhancements to current solutions. The Web is constantly growing and it contains documents on almost every subject on Earth, including an ever-increasing noise. The challenging nature of this problem that requires efficient on-the-fly processing and accuracy has attracted a lot of interest in recent years.

Scatter/Gather

Initial studies on SRC dates back to the very beginning of search engines. Archie [15][17] was the first search engine, created in 1990 at McGill University in Montreal. In its first form, Archie was used to locate a file by name and did not index the content of text files. Gopher protocol[1] brought the capability of indexing text content in 1991. In 1992, as the first study investigating document clustering as an access method, Scatter/Gather browsing paradigm[14] was published. Scatter/Gather system presents the user available topics to choose from and the user proceeds by picking a subset of topics according to his/her needs. The corpus get reduced and after reclustering this reduced set, a set of available topics is presented. This browsing method aims to serve as a better discovery tool.

In this study, it is stated that slow performance for large corpora is one of the key problems with document clustering. To tackle this problem, two rectangular time clustering algorithms, Buckshot and Fractionation, are presented.

These algorithms are used to find *k* initial centers in partitional clustering and they apply a cluster subroutine (*group average agglomerative clustering* in this study) to small sets in refinement steps. Among these two algorithms, Buckshot is faster but Fractionation has higher accuracy. Since the corpus is static in the beginning, Fractionation algorithm is used for offline computation of the initial partitioning. However, the result sets change dynamically after user interaction at every Gather step so Buckshot algorithm is used for fast, online clustering of these smaller sets of results.

Using the New York Times News Service articles from August 1990, Scatter/Gather demonstrates that document clustering can be an effective information access method with the availability of fast, online clustering algorithms.

In 1996, another work [33] evaluated Scatter/Gather using TIPSTER [31] corpus and compared its effectiveness to ranked titles. Results of experiments with TREC-4 queries show that clustering with ranking significantly outperforms similarity ranking alone. With both ranking methods applied (ranking by *closeness to the query* and ranking by *closeness to the centroid*), the best cluster stays more relevant than ranked titles. This study is the first to show that clustering significantly improves retrieval results over large text collections and verify the cluster hypothesis[71]. Compared to previous studies, the authors state that Scatter/Gather performs better because of two reasons. First is the use of full-text documents instead of titles and abstracts. Second is the dynamic nature of clustering in Scatter/Gather. Cluster centroids are not pre-calculated, static points and different clusters arise with different result sets.

Suffix Tree Clustering (STC)

After the introduction of document clustering as an efficient browsing method by Scatter/Gather, Zamir et al. [88] came up with two new clustering algorithms, performing better than Scatter/Gather in terms of both performance and clustering quality. In their first algorithm, Word-Intersection Clustering (Word-IC), they tackled slowness problem of Hierarchical Agglomerative Clustering (HAC) algorithms by applying a Global Quality Function (GQF) as a heuristic. Clusters are scored based on the number of words common to all documents in the cluster. This algorithm performs better than Scatter/Gather but still performs in $O(n^2)$ time.

The second algorithm introduces Suffix Tree Clustering (STC) in its raw form as Phrase-Intersection Clustering using Suffix Trees (Phrase-IC). Phrase-IC utilizes suffix trees to find shared phrases and count their occurrences in $O(n)$ time. After sorting cluster candidates according to their score, time complexity of this algorithm becomes $O(nlogn)$. Phrase-IC is incremental and allows overlapping clusters.

Evaluating Group Average Hierarchical Clustering (GAHC), Word-IC and Phrase-IC with snippets returned from MetaCrawler, Zamir et al. concluded that Word-IC with GQF performs best in terms of quality and Phrase-IC performs best in terms of speed. Since Scatter/Gather was only tested on top of an information retrieval (IR) engine, these two algorithms are the first SRC algorithms tested on snippets returned from Web search engines and it is also the first study concerning Web-snippet clustering.

In a following study, Zamir et al. [86] improved the idea of using a suffix tree for document clustering by adding the capability of merging base clusters. A base cluster is scored according to the number of documents and the number of effective words (words with high TF-IDF score) in that cluster. If majority (%50) of their members are overlapping, base clusters are merged and connected components become final clusters in STC. In the study, Single Pass and K-Means are compared to STC algorithm together with Fractionation and Buckshot from Scatter/Gather and GAHC. Single Pass and K-Means algorithms are selected as best candidates for the same purpose since they are linear time clustering algorithms that can produce overlapping clusters. Single Pass is also incremental like STC. However, STC outperformed all the algorithms mentioned above in terms of precision and performance. Further experiments on MetaCrawler data showed that STC highly benefits from allowing overlaps and multi-word phrases. These two features do not improve other algorithms like STC. The study also found that using snippets instead of full text Web documents have a relatively small impact (%15) on quality. Authors state that search engines try to extract meaningful phrases when they are summarizing a Web document with a snippet. Because of this, snippets stay useful despite being a small summary and clustering snippets is a reasonable, faster alternative.

With Grouper [87], Zamir et al. improved STC by eliminating nearly identical phrases and by filtering out general phrases that don't give better information. The latter means that when the system already listed a specific phrase (e.g. ”greenhouse gas emissions forecast”), there is no need to list a phrase that is more general and yet contains no extra information (e.g. ”greenhouse gas”). Zamir et al. also introduced a few performance tweaks to reduce %50 of the workload. These tweaks include elimination of stop-words at the start of phrase and elimination of rare phrases. By analyzing user logs (e.g. time spent traversing results, click distance) of Grouper, Zamir et al. also recognized that merging of base clusters can be confusing when merged clusters do not represent groups that the user expects and clusters should be presented hierarchically since number of clusters increase with larger result sets.

Hierarchical Clustering

The argument in favor of hierarchical clustering was also supported by Maarek et al. [43]. In hierarchical clustering, resulting clusters can be presented with a tree, containing more specific nodes in deeper levels. The study states that such an interface improves efficiency of user interaction because users can traverse the tree in logarithmic time as opposed to linear-time traversal of flat methods. Aside from reaching a cluster hierarchy, precision is also important. In order to produce tight clusters and increase intra-cluster relevance, Maarek et al. selected complete-link HAC method as the best method. However, this introduced a performance problem compared to $O(n)$ single-link and Ward's methods since the best complete-link algorithm to date performed in $O(n^2 logn)$ time.

Observing that only coarse granularity levels are required for SRC task, Maarek et al. discretized cluster similarity and applied bucket sorting to sorting phase of complete-link HAC method. This decreased sorting complexity from $O(n^2 logn)$ to $O(n^2)$ and produced an $O(n^2)$ complete-link HAC algorithm.

Maarek et al. emphasized that SRC requires high precision. Unlike previous studies using single words or multi-word phrases, pairs of words linked by Lexical Affinity (LA) are used as the indexing unit to improve precision. Preferring precision over recall, documents with low confidence (outliers) are not classified. Evaluations with manually-labeled (full-text) articles from Reuters-21578 dataset show that when outliers are removed, LA improves clustering quality by approximately %30, compared to single words as document features. This result reflects the importance of word proximity information for SRC tasks.

Link-based Clustering

There were other studies utilizing link information from hypertext documents. One of the earliest works to apply *co-citation analysis* on hypertext documents for clustering was [60]. In this algorithm, co-citation pairs above a frequency threshold were iteratively merged if they shared one document. With this algorithm, AB and BC pairs may be merged into ABC when co-cited document sets for AB pair and BC pair are disjoint. This results in clustering errors.

Kitsuregawa et al. [74] came up with another algorithm that represents a Web page with its inlinks and outlinks as two binary vectors, to test *co-citation* and bibliographic *coupling* measures. They applied a modified K-Means algorithm with similarity and merging thresholds. Factors such as maximum cluster size, number of singleton clusters, number of final clusters and cluster entropy were investigated in the study to decide on best values of these thresholds. They conducted their experiments with manually labeled search results from Altavista and authors state that results may be biased.

Since [74] method relies solely on in-link and out-link information, it cannot cluster pages without sufficient in-links and out-links. To tackle this problem, Kitsuregawa et al. [75] incorporated textual information into their Web page representation to combine link and contents analysis in clustering. Snippet, anchor text, meta-content and anchor window of the in-link are used to build a term vector for a Web page. This term vector contains frequency values for corresponding terms. Using manually labeled data as before, Kitsuregawa et al. compared the effect of contents, link and combined (contents+link) information in clustering. Results show that link-based clustering produces medium but tightly related clusters with low entropy. Content-based clustering produces clusters with high recall but link-based clustering has higher precision. Their combination performed best with good precision, lowest entropy and highest recall. Authors state that precision is slightly lower compared to link-based clustering because snippets and anchor windows introduce noise.

[32] also presented a similar approach making use of co-citation, link and contents information but normalized-cut method was used to achieve a better flat clustering than K-Means in this study. He et al. use a metric that combines link and textual information as a product of shared link score and textual similarity score and adds co-citation score to this product, weighted by α. Evaluations with (full text) results returned by *Hotbot* search engine showed that applying normalized-cut with this metric yields tight, high-quality clusters. By picking a suitable normalized-cut threshold, small clusters can be avoided. Time complexity of this algorithm is $O(N_r|E| + |V|log|V|)$. However, the results show that an optimal α value that can minimize the average normalized-cut for all cases does not exist. This means success of the algorithm is closely tied to data characteristics of the case. Additionally, since link score and text similarity score are multiplied, if two Web pages highly resemble each other but do not share links, this method will not be able to put them into the same cluster.

Fuzzy Clustering

Base cluster similarity definition in STC [86] requires conversion of a real value between 0 and 1 to a binary value. Jiang et al.[39] states that this conversion results in arbitrary cluster merge operations since the cut-off value directly affects similarity. Hence they propose that fuzzy clustering is more appropriate since it can use a soft similarity definition. Moreover, they state that SRC requires a robust method to cope with the noise introduced by snippets and irrelevant results from search engines. The study presents Retriever, which uses Robust Fuzzy C-Medoids (RFCMdd) algorithm to cluster search results either with Vector Space based or N-Gram based dissimilarity measure. Jiang et al. argue that phrase commonality, which is the very basis of STC, may not be suitable for snippets because after filtering irrelevant parts (stop words etc.) only a sentence or two remains. N-Gram RFCMdd is compared to both Vector Space RFCMdd and STC to shed light on this argument. Evaluations with search results from both MetaCrawler and Google show that the difference between inter-cluster and intra-cluster distance is highest with N-Gram RFCMdd method. As expected, stop-word elimination and stemming did not play an important role in N-Gram model unlike Vector Space model. This provides additional support for [16]. Jiang et al. conclude that N-Gram RFCMdd produces fewer, more focused clusters. The authors also mention the trade-off between using Vector Space model and N-Gram model in terms of computation time. RFCMdd also eliminates outliers in every medoid update. However, according to presented results, this noise elimination controlled by a parameter removes a number of relevant clusters that the user may be interested. Noisy clusters exist in STC but outlier elimination at this level may decrease precision. Limited experiments of the study did not investigate this effect in detail.

Hierarchical STC

In order to convert the highly efficient flat STC clustering algorithm [87] to a hierarchical one, Maslowska [46] proposed to convert binary base node similarity utilized in [87] to a directed inclusion relation. In [87], base nodes were considered similar if both $|B_m \cap B_n|/B_m$ and $|B_m \cap B_n|/B_n$ are over a specified threshold (%50 in the study). In Maslowska's method called Hierarchical STC (HSTC), if the ratio of common documents (e.g. $|B_m \cap Bn|/B_m$) exceeds this threshold, it means that node n includes node m. In HSTC, Maslowska identifies base clusters as in STC but then merges identical base clusters and proceeds to constructing a directed inclusion graph. Cycles are eliminated by merging the nodes of the cycle as a cluster with a set of phrases from these nodes. Finally, a set of kernel nodes (nodes that cover the whole graph with their out-links and that are not connected internally) are identified and subclusters are added to these top-level nodes by traversing their out-links. HSTC allows overlapping clusters and Maslowska states that HSTC has the same precision as STC, according to empirical evaluations.

A simple method using Term Co-occurence

As part of a Portuguese search engine development project called Tumba!, a simple result set clustering algorithm [65] utilizing term co-occurrence was implemented. If the documents containing term *y* are a subset of documents containing term *x*, *x* subsumes *y*. In the study, this subsumption relation is applied with a tolerance (requiring only %80 inclusion). However, the authors state that this method generates many meaningless clusters and pruning is essential. A set of simple heuristics are presented for pruning but the study does not present any evaluations of the method.

Clustering as Salient Phrase Ranking

Zeng et al.[91] reformulated SRC, which is an unsupervised clustering problem, as a supervised ranking problem. They extract all phrases (n-grams where $n \leq 3$) occurring at least 3 times from the snippets and compute properties such as TFIDF, phrase length and phrase independence for these phrases. Moreover, properties about their contained document set such as intra-cluster similarity and cluster entropy are also computed. Zeng et al. combined these measures linearly and optimized this by applying different regression methods on human labeled training data, consisting of ambiguous queries (e.g. *jaguar*), entity names (*clinton*) and general terms (*games*). In clustering result of this method, a cluster is actually the document set containing selected salient phrase. Evaluations with results from MSN search engine show that this linear-time method converges to a precision of %73.3 for top 5 results with 200 results. However, low precision (around %45) for top 20 results and low coverage are also striking problems with the method.

Semantic Methods

STC[88][86][87][89] is very useful and can be regarded as a break-through in SRC when it's applied to English. Despite its speed and precision with English, it has several drawbacks when it is applied to other languages as stated in [77]. Frequencies in human languages are subject to distortion in many ways. Synonymy, polysemy, varying word order, pronouns, varying character sets, absence of explicit word/sentence separators, form-changing prefix and suffixes etc. are all important factors that can reduce cluster quality.

Zhang et al. [92] introduced Semantic Hierarchical Online Clustering (SHOC) as a language-aware algorithm that can cope with the errors of STC affecting oriental languages. Since the performance of suffix tree data structure is related with alphabet size, it is not appropriate for oriental languages having a much larger set of characters (over 6000 for Chinese). Additionally, absence of explicit word separators (e.g. blanks in English) causes significant noise in phrases generated for multi-lingual search results. As a solution, SHOC applies a key-phrase extraction algorithm based on suffix arrays[44] to generate better cluster labels. *Stability* measure used in key-phrase extraction is similar to *phrase independence* [91] in nature. Completeness and significance ensures that the key-phrase is maximally repeating and has a high frequency and *phrase length*. Representing data as document vs. key-phrase matrix, SHOC applies orthogonal clustering by using singular value decomposition (SVD). Finally, SHOC achieves a hierarchical clustering by iteratively merging and organizing the clusters generated by orthogonal clustering. Unfortunately, Zhang et al. did not present any evaluations of SHOC and the effectiveness of the algorithm is unknown.

Inspired by SHOC[92], Osinski et al. [54] developed a new, description-oriented, flat SRC algorithm called *Lingo*. Lingo tries to solve language related problems mentioned in [77] similar to SHOC. However, it separates cluster label discovery phase from clustering completely and puts higher priority to cluster description. Lingo first applies language identification, stop-word removal and stemming to input snippets. After this step it identifies candidate phrases using suffix arrays like SHOC but then applies dimensionality reduction on term-document matrix using SVD. The number of candidate phrases, hence the number of final clusters, depends on a *candidate cluster threshold*. Lingo assigns phrases to abstract concepts by computing cosine distance and selecting the closest phrase. This distance between an abstract concept and a phrase becomes the score of the cluster candidate. After pruning, cluster labels serve as indexing terms in Vector Space model and snippets are assigned to a cluster if they exceed a *snippet assignment threshold*. Evaluations on Open Directory Project (ODP, also known as Directory Mozilla, DMOZ) data [56] show that Lingo is superior to STC in cluster labeling and separating mixed documents to their topics. In the experiments, STC chose common, frequent and meaningless labels (e.g. *include*, *used*) and mixed documents based on these meaningless phrases. The results also suggest that both Lingo and STC have a tendency to over classify DMOZ categories.

SRC methods relied on a handful of parameters to be determined by experiments with limited data. Thresholds like *candidate cluster threshold* in Lingo directly affect quality of the results. To eliminate the need for parameters and always achieve an optimal clustering, Mecca et al. [48] proposed *dynamic SVD clustering*, which incrementally projects documents to k-dimensional spaces and removes $k-1$ longest edges from their minimum spanning tree (MST). Dynamic SVD tries to maximize the difference between the length of $k-1$ longest edges in MST and the average with a quality function. The number of clusters k is decided when a local maximum is found. Evaluations with full-text documents, both manually labeled results returned by Google and document from DMOZ categories, point to a significant improvement over STC and Lingo. Dynamic SVD achieves an average Grouper quality of %90.3 (STC scores a maximum of %80) and cluster contamination values below %5 (Lingo has an average of %25). Despite these good results, the limited number of input documents (97 documents in maximum case) for clustering should be noted. Analysis also reflect that using snippets instead of full-text documents severely degrades performance (more than %40 in average F-measure). Since SVD computations are costly (9 seconds for 97 documents in the study), STC is still the fastest algorithm with decent results.

Using DMOZ categories as Knowledge Base

Grouper [87][89] used only continous phrases to measure the similarity between documents. As discussed in [66], this causes problems when the algorithm is applied to human languages where the positional order of parts of speech is subject to change. SnakeT [20][21] was proposed to overcome this problem by extending the use of lexical affinity (LA). SnakeT used two knowledge bases: an anchor to URL mapping created with 50 million URLs crawled using Nutch and a semantic knowledge base using a DMOZ index to rank an *approximate sentence* according to its frequency within DMOZ categories. It first generates 2-gapped sentences with the method used in [43] and then merges sentences in a snippet and a certain proximity window to generate k-gapped, longer sentences. Snippets that share the same approximate sentences are clustered together. Finally, k-approximate sentences that have a good rank and shared by certain majority (%80) of clusters are used to generate parent labels and achieve a hierarchy.

Formal Concept Analysis

There are studies approaching SRC problem as a formal concept analysis (FCA) task. FCA methods try to identify partial orders between a set of objects and combine them in a concept lattice. An early application of FCA to SRC problem is CREDO [7], which yields a 2-level hierarchy in two phases. CREDO first applies stop-word removal and stemming on input data and generates most general concepts by analyzing titles. Then concepts in lower levels are recursively generated with FCA method using both title and snippets. CREDO only uses single words as indexing terms but it can produce multi-word cluster labels with its use of FCA method. This method can be considered as rectangular-time since its complexity is $O(nmC + m)$ where C is the number of clusters, n is the number of documents and m is the number of terms. A notable difference of CREDO from other methods is that it produces a lattice instead of a tree which provides navigational flexibility.

Revisiting *K*-center approach

In 2006, Geraci et al. [28][27] approached SRC as a classical k-center problem again, with a data-centric method. They used a modified *furthest-point-first* (FPF) algorithm [29] that has better performance than existing fast variants of K-Means. Using weighted term frequencies, Geraci et al. clustered snippets with this approximate, rectangular-time algorithm and generated cluster labels with Information Gain criterion. According to their evaluations with ODP data, their system (called *Armil*) performed better (around %10) than Vivisimo when it was run with a similar target of 40 clusters. Geraci et al. used *normalized mutual information* (NMI) and *normalized complementary entropy* (NCE) from [67] in this comparison but a later study [93] explains that these measures are biased.

Improvements on Suffix Tree Clustering

Wang et al.[73] observed that similarity measure defined in the original STC algorithm [86] by Zamir and Etzioni has two disadvantages. When ratio of the number of documents in cluster m and n is unbalanced, it prevents merging of clusters even when one is a subset of the other. Moreover, when there are similar documents in separate, non-overlapping clusters, original STC cannot merge these clusters because the similarity formula only takes overlapping clusters into account. Wang et al. solved these problems by introducing a new similarity formula for merging. This new formula is the combination of two formulas measuring the overlap (similar to original STC but it fixes the problem with unbalanced clusters) and textual similarity of non-overlapping documents. Evaluations with manually labeled search results from Google showed that this new merging algorithm improves STC.

In another study [11], Chim et al. investigated the use of suffix tree as a document model. Using STC algorithm [86], Chim et al. represents documents as a feature vector of base nodes, weighted by TF-IDF frequencies of corresponding phrases. They introduce the notion of *stopnodes* to Suffix Tree Model (STM) which is similar to *stopwords* in Vector Space Model (VSM). Chim et al. evaluated the effectiveness of STM by clustering OHSUMED [34] medical abstracts and RCV1 [42] corpus using GAHC algorithm. Their new similarity measure (NSTC) brings a performance improvement of %51 over TF-IDF cosine similarity (TDC) and %22 over STC with an average F-measure score of 0.83.

Improving Formal Concept Analysis

Observing that concept lattice generated with FCA may contain irrelevant concepts in its raw form, [93] introduced an improved version of CREDO, called Conceptual and Hierarchical Clustering (CHC). CHC filters and organizes the raw concept lattice by measuring concept importance, concept similarity and concept coverage. The study compares CHC with STC, Lingo and Vivisimo with ODP data by using Average NMI (ANMI) and Average NCE (ANCE) metrics, which are improved, unbiased versions of normalized mutual information (NMI) and normalized complementary entropy (NCE), employed in another comparative SRC study[28]. Results show that CHC performs best in ANMI@K but slightly worse than Lingo in ANCE@K. Yet, the authors state that Vivisimo still generates the best cluster labels among these algorithms. Zhang et al. also mention the idea of using external knowledge sources as a possible improvement.

Using Wordnet as Knowledge Base

By extending document representation with Wordnet synsets [35], Hotho et al. found that Wordnet improved text clustering. They tested different strategies to utilize Wordnet concepts: adding concepts to the term vector, replacing a number of terms with their concepts in the term vector or only using the concept vector. Querying Wordnet returned all related concepts about a term so they also tested 3 different disambiguation strategies: using all available concepts, only the first (common) concept or using the concepts that maximize TF scores of its subconcepts and superconcepts in its context. Finally, they considered hypernyms of a concept to improve the representation with a parameter controlling the depth. They achieved a purity improvement of %8.4 on Reuters-21578 news corpus by using background knowledge with 5 levels of hypernyms, using *disambiguation by context* and term vectors extended by concept frequencies.

Using Wikipedia as Knowledge Base

[63] extended CREDO [7] by incorporating redirection, disambiguation and strong link information from Wikipedia into the concept lattice and CREDO result page. Redirections are used to merge different forms of a concept. As an example, concepts like ”president kennedy” and ”j.f.k.” can be combined as ”john f. kennedy” using redirections. Disambiguations are used to enrich the concept lattice since the number of results, hence the number of concepts, that a search engine can provide in first few pages is limited. This way, ”ruby” query can be extended with *gemstone*, *pistol* or *elephant* senses of the word, which are usually buried under *programming language* results. The authors state that integrating information gained from strong links into an existing formal context is challenging so they are used to present related concepts with the results.

Gabrilovich et al. [23][25] used their state-of-the-art Explicit Semantic Analysis (ESA) [26][22][24] method to classify documents from Reuters-21578, RCV1, OHSUMED, 20 Newsgroups (20NG) [41] and movie reviews from [58] using SVM [38] with a linear kernel. Measured by precision-recall break-even point (BEP), their results showed that ESA significantly improved classification performance with improvements up to %30.4 for RCV1 and %18 for OHSUMED.

In a follow-up study, Hu et al. [36] reiterated that WordNet has limited coverage and lacks effective word-sense disambiguation and as [23], they proposed to use Wikipedia to enrich text representation. However, Hu et al. stated that Gabrilovich et al. did not utilize hierarchical information in Wikipedia and treated synonym, hypernym, associative concepts and terms equally. As an improvement, they used category links to decide on hypernymy and out-linked categories with content similarity to decide on synonmy and associative relations. Using K-Means to cluster Reuters-21578 and OHSUMED data, they compared their method with Hotho et al. [35] and Gabrilovich et al. [23]. Results showed that their method significantly improves clustering, yielding the best purity and inverse purity scores.

CHAPTER 3

BACKGROUND

3.1 Search Result Clustering

Data clustering is a statistical data analysis technique that is used to partition a data set into a set of meaningful groups according to given criteria. Different clustering algorithms have implicit weaknesses and strengths against different definitions of cluster meaningfulness. However, the main goal of clustering algorithms remains the same: sum of the distances between data points in the cluster (intra-cluster distance) should be minimal and sum of the distances between clusters (inter-cluster distance) should be maximal.

Document clustering is often a harder task than clustering numerical data because one only has a numerical representation of textual data, defined mainly by word frequency. For human languages, most statistical correlations are noise according to context and this makes document clustering even harder.

Over the years, mapping documents to vectors defined by term frequency weights to compute distances in Vector Space Model (VSM) remained as an effective strategy. With reliable frequency information and proper noise removal, good results are achieved with many clustering algorithms.

Search Result Clustering (SRC) is sub-topic of document clustering in which the clustering methods are expected to produce clusters with intuitive descriptions with limited input. The input to SRC algorithms are snippets which are typically one or two sentences long.

Common requirements for SRC methods can be defined as the following[77]:

- Algorithms should be fast, preferably linear-time and incremental.
- Algorithms should produce good results even with short text fragments.
- Algorithms should allow overlapping clusters. Search results often have multiple topics.
- Cluster labels should be intuitive. They should allow the users to understand the reason behind a clustering.

Unlike other document clustering tasks, the use of short snippets as input is a pressing factor in SRC problem since the information gained from term frequencies tend to become noisy and meaningless for smaller sample size.

3.1.1 Suffix Tree Clustering

Suffix Tree Clustering (STC) is a search result clustering algorithm which runs in linear time. Built around the principles of snippet-tolerance, speed, incrementality and ability to produce overlapping clusters, it remains as the fastest algorithm showing the feasibility of SRC. We discuss it in further detail because STC algorithm is the main subject of the improvements in this study.

3.1.1.1 Original STC - Zamir et al.

STC algorithm first appeared in a study [88] from 1997, in its simplest form called *Phrase-intersection clustering (Phrase-IC)*. As in all subsequent STC implementations, input documents are first cleaned by marking sentence boundaries, removing non-word tokens (e.g. numbers, punctuation), stop word removal and stemming. Then these cleaned strings from input documents are inserted into a data structure called *generalised suffix tree*.

Suffix tree data structure is at the heart of STC algorithm. It can be constructed in linear time, incrementally[70] and once constructed, it allows fast string operations such as retrieving the frequency of a phrase in constant time.

As Zamir et al.[86] defines, a suffix tree of a string *S* is a *compact trie* containing all suffixes of *S*. This means that a suffix tree has the following properties:

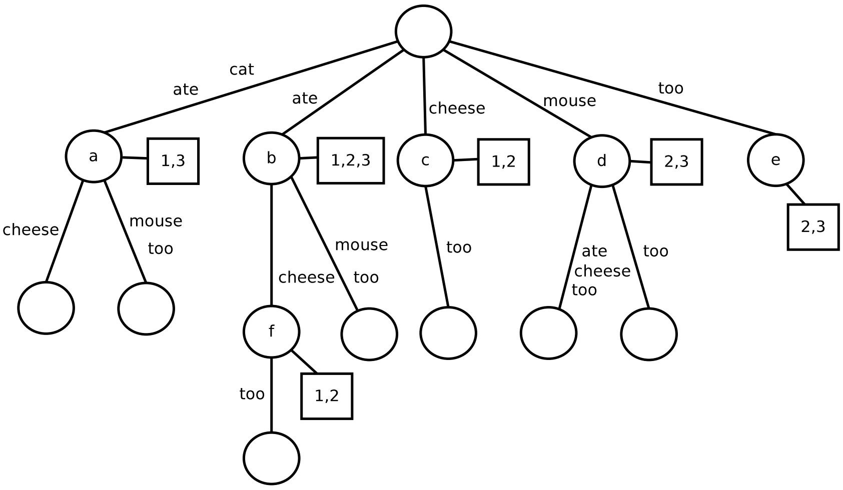


Figure 3.1: The suffix tree of documents ”cat ate cheese” (doc 1), ”mouse ate cheese too” (doc 2) and ”cat ate mouse too” (doc 3)

- Except the root, each internal node has at least 2 children.
- Each edge is labeled with a non-empty substring of S.
- There can be no edges from the same parent with the same label.
- A suffix corresponds to only one path in the tree.

A *generalised suffix tree* is a suffix tree constructed from a set of strings. In STC algorithm, sentences from documents are inserted to the suffix tree as words, not characters. For each internal node, the document where that suffix occurs is marked as shown in Figure 3.1. Internal nodes of the suffix tree represent phrases shared by groups of documents so each internal node can be viewed as a base cluster encapsulating 2 or more documents. Traversing all internal nodes to read this information costs $O(n)$ time.

During this traversal, STC algorithm assigns a score to every base cluster. In *Phrase-IC*, the score $s(B)$ of base cluster B is computed as:

$$s(B) = |B| \cdot |P|$$

where $|B|$ is the number of documents in base cluster and $|P|$ is the phrase length.

Zamir et al. [88] states that after computing base cluster scores, *Phrase-IC* sorts these potential clusters and determine the clusters using a simple selection algorithm which ensures that selected clusters are not identical or highly overlapping. In this form, *Phrase-IC* runs in $O(nlogn)$ time since sorting and selecting from $O(n)$ potential clusters takes $O(nlogn)$ time.

As a follow-up study, Zamir et al.[86] modified *Phrase-IC* in three major ways:

- They modified their base cluster scoring function to reduce the effect of irrelevant words.
- They added a merging step to the algorithm as a better strategy to deal with highly overlapping clusters.
- Unlike *Phrase-IC*, STC algorithm is incremental. Each document is added to the suffix tree as it arrives. After each addition, similarities of relevant base clusters are recalculated and final clusters are updated if necessary. Only the k highest scoring base clusters ($k = 500$ in this study) are considered for similarity computations to keep the cost constant.

This is the first version of the actual *Suffix Tree Clustering (STC)* algorithm. In *STC*, the score $s(B)$ of base cluster B is computed as:

$$s(B) = |B| \cdot f(|P|)$$

where $|B|$ is the number of documents in base cluster and $|P|$ is the *effective phrase length*. Words appearing in the stoplist, rare words (appearing in 3 or less documents) or too frequent words (appearing in more than 40% of the collection) do not contribute to the effective phrase length. f is a function which penalizes single word phrases, is linear until 6-word phrases and constant for longer ones.

For the merging phase, STC utilizes a binary similarity measure. Any two base clusters B_m and B_n are only similar iff:

$$\frac{|B_m \cap B_n|}{|B_m|} > 0.5 \quad and \quad \frac{|B_m \cap B_n|}{|B_n|} > 0.5 \tag{3.1}$$

where 0.5 is the *similarity threshold*.

As in Figure 3.2, similar base clusters form an undirected graph, with typically small connected components. Base clusters from each connected component are merged and treated as a single final cluster. The sum of base cluster scores becomes final cluster score. Zamir et

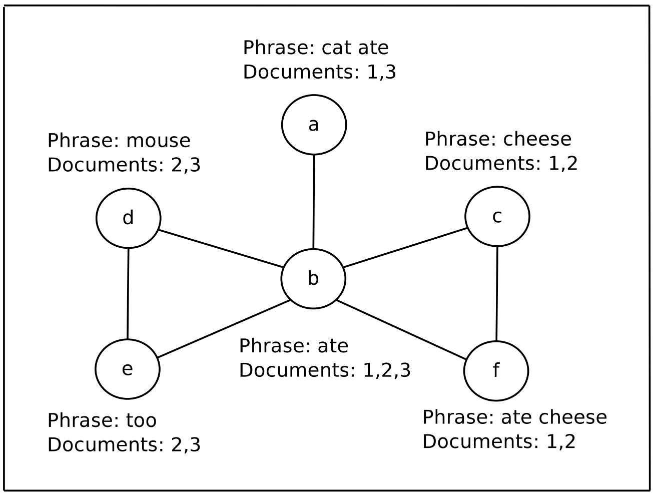


Figure 3.2: Base cluster graph of the example given in Figure 3.1

al. state that this step is indeed a single-link clustering algorithm running on base clusters. However it does not suffer from chaining effect because the connected components are small. The number of final clusters can vary. Thus, after sorting final clusters, they only report top 10 clusters that are typically of interest.

In their analysis, Zamir et al. found that the use of phrases instead of single words and the nature of STC allowing overlapping clusters are the main reasons behind the success of STC. Moreover, they also found that STC is not as sensitive to the similarity threshold, unlike agglomerative hierarchical cluster algorithms high sensitivity to the number of clusters required.

Then, in Grouper system [87], Zamir et al. presented improvements to STC for selecting better cluster labels in the merge step. Previously, all merged phrases in a connected component were concatenated and displayed as cluster label. However, this does not result in concise cluster labels. As a better strategy for selecting cluster labels, they proposed the following:

- If a phrase *P* shares most (more than 60% in the study) of its effective words with a phrase having a higher coverage, *P* should not be displayed.
- Phrases other than most general and most specific phrases should not be displayed.
- Only display a most-general phrase if it significantly (20% in the study) improves the coverage of its most-specific phrase.

For each cluster, a maximum of 5 phrases are displayed, selected according to previous rules. Moreover, observing that stop words in the beginning and end of phrases do not change phrase semantics, these stop words are stripped before inserting strings into the suffix tree. Rare words are also stripped off since rare words can only create rare phrases. With these changes, input size is reduced and all operations including suffix tree construction take less time (reported as a reduction of 50% in the study).

3.1.1.2 Improvements over STC

Crabtree et al.

There have been studies trying to improve STC after the original studies from Zamir et al. One such study deals with the coverage problem in STC, based on the idea that the best clustering is achieved when there is minimal overlap and maximal coverage. In 2005, Crabtree et al.[13] observed that the scoring from the merging phase of STC over-emphasized overlapping clusters and this decreased the coverage of final clusters when STC was applied on full-text documents, reducing clustering quality. The problem arised in full-text documents because STC generates too many overlapping base clusters with increased text.

To solve this problem, Crabtree et al. distributed the score of each base cluster to their constituent documents, instead of simply summing up the scores of merged clusters to compute final cluster score. For overlapping documents, each such document is assigned the average of document scores from different clusters. The score for the final cluster is computed by summing up the scores of member documents.

Although this new scoring method solved overlapping problem for base clusters, coverage of final clusters was still not optimal. As a heuristic, Crabtree et al. considered adding clusters to the final cluster set only if more than 50% of their documents were distinct. However, as in Figure 3.3, they explained that better solutions might be missed without look-ahead.

In Figure 3.3, the clusters are ordered as D,A,B,C according to the number of distinct documents and when the algorithm selects D without looking ahead for combinations of A,B and C, it stops after adding {D}. However, with 1-step look-ahead, all cluster combinations of 1 or 2 clusters are considered and finally, {A,B,C} is selected. At each step, there can be many combinations that need to be considered but according to the order and overlap between

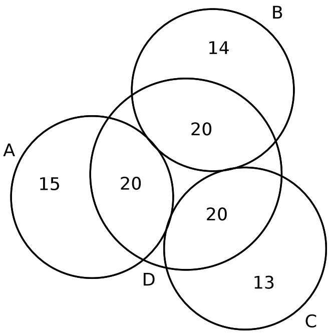


Figure 3.3: Cluster Selection Example from Crabtree et al.

clusters, pruning is applied to increase performance.

Wang et al.

In another study, Wang et al.[73] investigated the scoring formula used in STC. They observed that there are two disadvantages of using Equation (3.1):

- Size of the clusters affect similarity. This means that even though cluster *A* is a subset of cluster *B*, *A* and *B* cannot be merged when *B* is too big.
- It only deals with the similarity of overlapping parts. Even if cluster *A* and cluster *B* have no overlap, their non-overlapping parts might have very similar documents.

As solutions to these two issues, they first modified the formula for computing overlap:

$$Overlap(B_m, B_n) = \frac{|B_m| \cap |B_n|}{Min(|B_m|, |B_n|)} \tag{3.2}$$

Then they applied cosine similarity formula for computing textual similarity of non-overlapping parts:

$$Sim(B'_m, B'_n) = \frac{\overline{B'_m} \cdot \overline{B'_n}}{|\overline{B'_m}| \times |\overline{B'_n}|} \tag{3.3}$$

They finally combined these two formulas by weighing with α (0.6 in the study) :

$$S_{m,n} = \alpha * Overlap(B_m, B_n) + (1 - \alpha) * Sim(B'_m, B'_n) \tag{3.4}$$

and specified $S_{m,n} > k$ as merging condition, where k is the *similarity threshold* (0.5 in the study).

3.1.2 Carrot2 Search Results Clustering Engine

Carrot2[57] is an open source Web Information Retrieval and Web Mining framework, focused in search results clustering. It serves as the main tool for our study, by enabling us to implement and evaluate different clustering algorithms in a standard environment. Driven by both academic and commercial purposes, Carrot2 minimizes the effort required to create a usable SRC system.

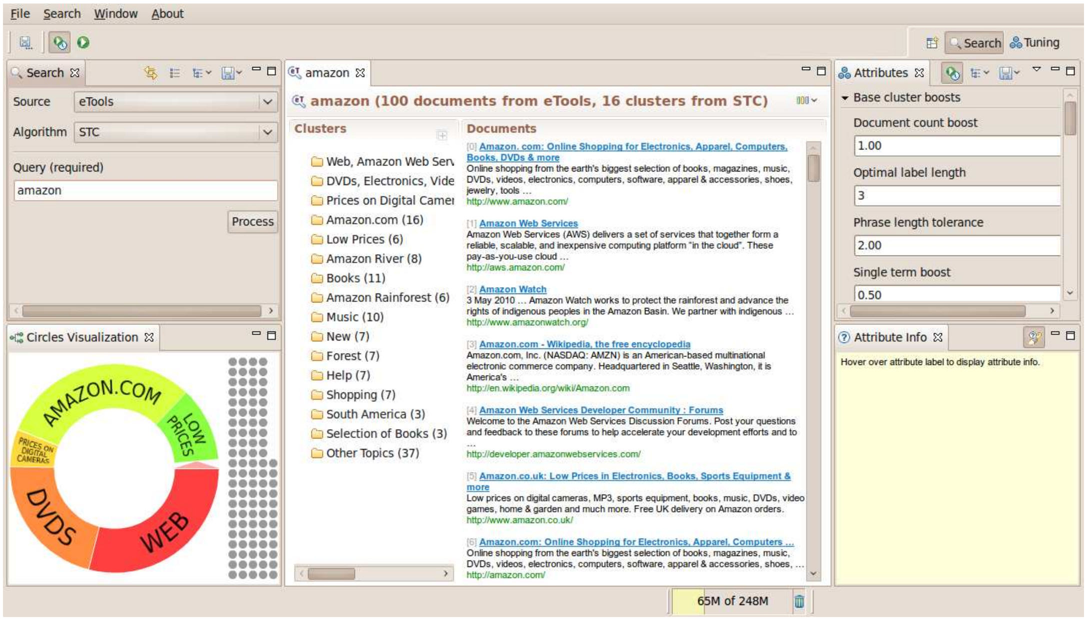
Carrot, the initial version of Carrot2, was created by Weiss et al.[78] in 2003. Carrot was first used to analyze the original STC algorithm with Polish data. It was also used to implement a hierarchical variation of STC algorithm[47]. Continuous improvements on Carrot led to Carrot2, which included implementations of classic agglomerative techniques (AHC), K-means, fuzzy clustering[40], biology-inspired clustering[64] and Lingo[55], together with STC[89]. However, Carrot2 only includes STC and Lingo implementations as of version 3.0.

Carrot2 provides the following facilities[57]:

- Search engine interfaces, including meta-search engines (e.g. eTools.ch, MSN, Yahoo) specific search engines (e.g. Pubmed)
- Efficient tokenization
- Trigram based language identification
- Stopword filtering and stemming (for languages supported by Snowball)
- Test datasets (AMBIENT and ODP239) and quality measurements
- Search result ranked list and cluster presentation
- Runtime performance measurements (result download time, algorithm running time)

Although Carrot2 includes language identification and stemming, they are only active in Lingo by default, not in STC. Implementations are responsible for utilizing these features.

Carrot2 provides both a workbench and a web application. Effects of algorithm parameters can be investigated using the workbench as in Figure 3.4 and resulting clusters can be visualized in detail as shown in Figure 3.5. The web application however, can be used to serve a clustering search engine from a web page, in a simpler form as in Figure 3.6.



Source: eTools

Algorithm: STC

Query (required): amazon

Process

amazon (100 documents from eTools, 16 clusters from STC)

Clusters
- Web, Amazon Web Serv
- DVDs, Electronics, Vide
- Prices on Digital Camer
- Amazon.com (16)
- Low Prices (6)
- Amazon River (8)
- Books (11)
- Amazon Rainforest (6)
- Music (10)
- New (7)
- Forest (7)
- Help (7)
- Shopping (7)
- South America (3)
- Selection of Books (3)
- Other Topics (37)

Documents

[0] Amazon. com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more
Online shopping from the earth's biggest selection of books, magazines, music, DVDs, videos, electronics, computers, software, apparel & accessories, shoes, jewelry, tools ...
http://www.amazon.com/

[1] Amazon Web Services
Amazon Web Services (AWS) delivers a set of services that together form a reliable, scalable, and inexpensive computing platform "in the cloud". These pay-as-you-use cloud ...
http://aws.amazon.com/

[2] Amazon Watch
3 May 2010 ... Amazon Watch works to protect the rainforest and advance the rights of indigenous peoples in the Amazon Basin. We partner with indigenous ...
http://www.amazonwatch.org/

[3] Amazon.com - Wikipedia, the free encyclopedia
Amazon.com, Inc. (NASDAQ: AMZN) is an American-based multinational electronic commerce company. Headquartered in Seattle, Washington, it is America's ...
http://en.wikipedia.org/wiki/Amazon.com

[4] Amazon Web Services Developer Community : Forums
Welcome to the Amazon Web Services Discussion Forums. Post your questions and feedback to these forums to help accelerate your development efforts and to ...
http://developer.amazonwebservices.com/

[5] Amazon.co.uk: Low Prices in Electronics, Books, Sports Equipment & more
Low prices on digital cameras, MP3, sports equipment, books, music, DVDs, video games, home & garden and much more. Free UK delivery on Amazon orders.
http://www.amazon.co.uk/

[6] Amazon.com: Online Shopping for Electronics, Apparel, Computers ...
Online shopping from the earth's biggest selection of books, magazines, music, DVDs, videos, electronics, computers, software, apparel & accessories, shoes, ...
http://amazon.com/

Attributes

Base cluster boosts

Document count boost: 1.00

Optimal label length: 3

Phrase length tolerance: 2.00

Single term boost: 0.50

Hover over attribute label to display attribute info.

Figure 3.4: Carrot2 Workbench showing results for *amazon* query

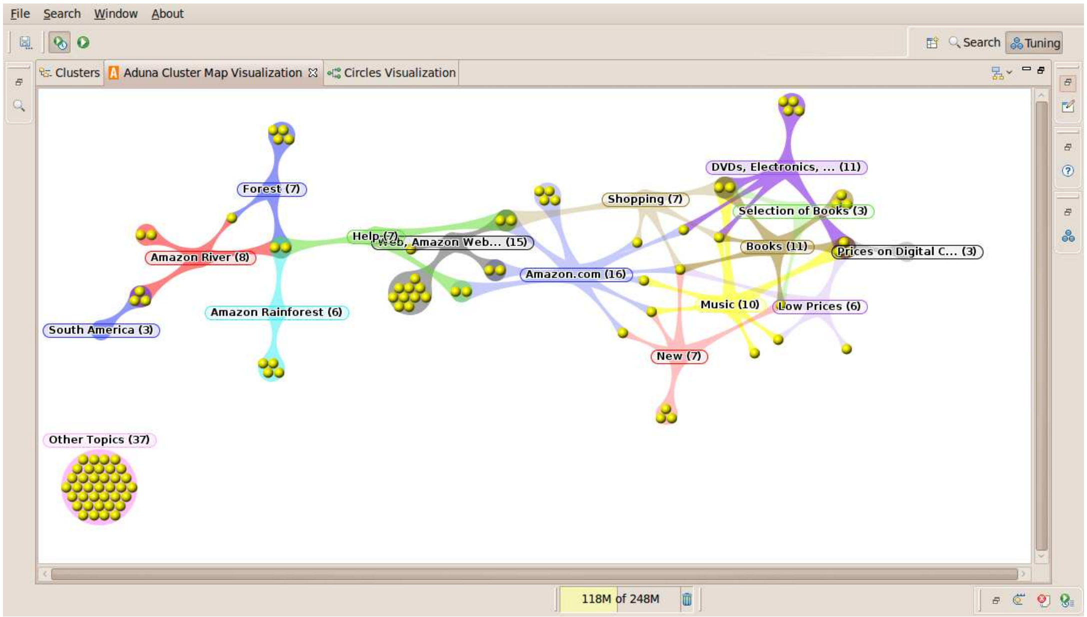
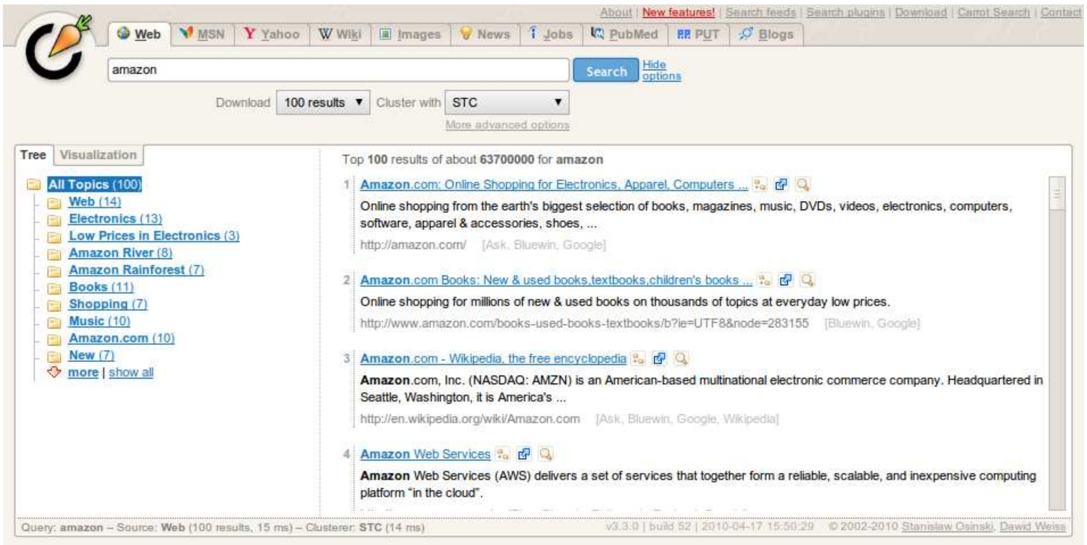


Figure 3.5: Aduna Cluster Map visualization from Carrot2 Workbench, showing results for *amazon* query



amazon

Download: 100 results
Cluster with: STC

All Topics (100)
- Web (14)
- Electronics (13)
- Low Prices in Electronics (3)
- Amazon River (8)
- Amazon Rainforest (7)
- Books (11)
- Shopping (7)
- Music (10)
- Amazon.com (10)
- New (7)

Top **100** results of about **63700000** for **amazon**

1 Amazon.com: Online Shopping for Electronics, Apparel, Computers ...
Online shopping from the earth's biggest selection of books, magazines, music, DVDs, videos, electronics, computers, software, apparel & accessories, shoes, ...
http://amazon.com/ [Ask, Bluewin, Google]

2 Amazon.com Books: New & used books,textbooks,children's books ...
Online shopping for millions of new & used books on thousands of topics at everyday low prices.
http://www.amazon.com/books-used-books-textbooks/b?ie=UTF8&node=283155 [Bluewin, Google]

3 Amazon.com - Wikipedia, the free encyclopedia
Amazon.com, Inc. (NASDAQ: AMZN) is an American-based multinational electronic commerce company. Headquartered in Seattle, Washington, it is America's ...
http://en.wikipedia.org/wiki/Amazon.com [Ask, Bluewin, Google, Wikipedia]

4 Amazon Web Services
Amazon Web Services (AWS) delivers a set of services that together form a reliable, scalable, and inexpensive computing platform "in the cloud".

Query: **amazon** – Source: **Web** (100 results, 15 ms) – Clusterer: **STC** (14 ms) v3.3.0 | build 52 | 2010-04-17 15:50:29 © 2002-2010 Stanisław Osiński, Dawid Weiss

Figure 3.6: Carrot2 Web Application, showing results for *amazon* query

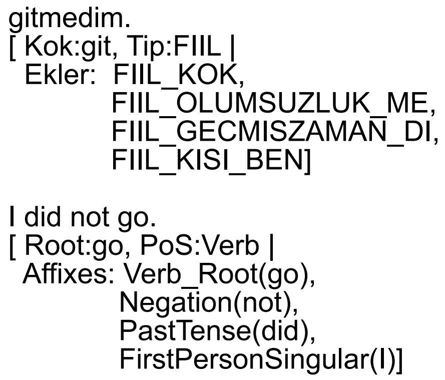
3.2 Natural Language Processing

Natural Language Processing (NLP) is an intersecting area of computer science and linguistics which involves the study of underlying mechanisms used in human languages in order to design computer systems that are able to analyze, understand, and generate these languages.

Like other research areas working on textual data, NLP techniques are almost always used in document clustering, at least for preprocessing. When text input is converted to statistical data, cleaning steps such as stopword removal and stemming are essential to increase accuracy.

3.2.1 Part of Speech

Part-of-Speech (POS) or *lexical category* is a linguistic category that groups different lexical items (words) according to their linguistic function or behaviour. As an example, *nouns*, *verbs*, *adjectives*, *adverbs*, *pronouns*, *prepositions*, *conjunctions* and *articles* are parts of speech describing English. Categorization of a word is affected by its morphological, syntactic and semantic properties. Part of speech tagging, which is the process of identifying the category for a word, is generally seen as a sentence-level task because the contribution of



```
gitmedim.
[ Kok:git, Tip:FIIL |
  Ekler:  FIIL_KOK,
          FIIL_OLUMSUZLUK_ME,
          FIIL_GECMISZAMAN_DI,
          FIIL_KISI_BEN]

I did not go.
[ Root:go, PoS:Verb |
  Affixes: Verb_Root(go),
           Negation(not),
           PastTense(did),
           FirstPersonSingular(I)]
```

Figure 3.7: An example POS tagging by Zemberek

these linguistic properties vary according to language. Most of the time, semantic properties of a word depend on other words from the sentence. An example for POS tagging is given in Figure 3.7.

In this study, we have used POS tagging to increase the performance of clustering by boosting or filtering out clusters based on their labels.

POS tagging task in our case is significantly different because unlike traditional POS tagging scenarios, cluster labels are much shorter than a sentence. Moreover, for POS tagging in Turkish, NLP tools limit us to using word-level morphological analysis. We used Zemberek, an NLP library for Turkish, for this task in this study.

Fortunately, our case does not require a complex POS tagging strategy. We performed experiments filtering out phrases not ending with nouns, leaving only noun clauses. Effects of this heuristic are discussed in the relevant section.

3.2.2 Natural Language Processing Tools

3.2.2.1 Snowball Stemmer

Snowball is small programming language created by Martin Porter to define stemmers.[61] Originally motivated by the need for standard stemmer definitions, Snowball project has come to include stemmers for many different languages. Turkish stemmer in Snowball is implemented by Evren (Kapusuz) Çilden in 2007, based on [19]. Snowball is part of our work because *Carrot2* framework uses it for stemming different languages including English and Turkish by default.

3.2.2.2 Zemberek

Zemberek[90] is an open-source NLP framework for Turkic languages, which provides functions such as morphological analysis, error-tolerated parsing, word suggestion and spellchecking. We have integrated the Turkish stemmer from Zemberek to *Carrot2* framework and we also use it to analyze Turkish words morphologically for identifying nouns.

3.3 Knowledge Bases

3.3.1 Open Directory Project

Open Directory Project (ODP)[52], also known as *Dmoz* which is an acronym for *Directory Mozilla*, is a project dedicated to build an open, community-edited hierarchical ontology of web pages. Every branch in this hierarchy correspond to a topic, including links to web pages and associated short (25-30 words) summaries. As of May 2010, ODP has 4,528,597 web sites classified in over 590,000 categories. Multilingual content is included under *World* category and we made use of the data from ”World/Türkçe” category in this study, for generating Turkish data based on ODP. As in this work, ODP has been used in various SRC studies (e.g. [56][29]) to derive a gold standard clustering from human-labeled data.

3.3.2 Wikipedia

Wikipedia[79] is a collaborative encyclopedia project aiming to create a free, web-based, multilingual encyclopedia that can be edited by anyone accessing the site. *Nupedia*, the predecessor of Wikipedia project, was created by Jimmy Wales and Larry Sanger on March 2000. Unlike Wikipedia, Nupedia only allowed peer-reviewed articles written by experts.

Upon observing the slow progress of Nupedia, Wales and Sanger created a *wiki* for Nupedia project on January 10, 2001. A *wiki* is a web site that allows collaboration of multiple users by enabling them to create, edit, and hyperlink pages easily. Wales and Sanger initially introduced this wiki to collect user contributions for subsequent editorial review by experts. However, users did not support this process and ultimately, Wikipedia was created to allow edits by anyone and launched on January 15, 2001.

Fundamental facilities of Wikipedia include redirects, disambiguation pages, templates and categories. Aside from handling typographical errors, redirects allow Wikipedia to handle synonyms. ”President Kennedy” and ”JFK” are both mapped to the same concept, ”John F. Kennedy”. Disambiguation pages list all possible concepts for a given word. As an example, the disambiguation page for ”apple” includes links to ”Apple Inc.” (company), ”Apple” (fruit) and ”Apple Corps” (record company). Templates allow contributors to reuse existing pages by including a modified version of them inside other pages. Every page in Wikipedia belongs to at least one category and category pages list their sub-categories or pages. Categories in Wikipedia form a network rather than a tree.

3.3.2.1 Wikipedia English

English Wikipedia is the pioneering language edition of Wikipedia project. It was the only language edition until the introduction of German Wikipedia in March 2001. As of May 19, 2010, there are 3,296,597 articles in English Wikipedia.

There are studies[26][2][63][59][36][62] using English Wikipedia for enriching textual data with conceptual features. Aside from direct concept mapping, its category network and link structure are also exploited. In this thesis, we have used English Wikipedia to support existing textual features with conceptual features since our inputs are short and sparse text fragments.

3.3.2.2 Wikipedia Turkish

Turkish Wikipedia, spelled *Vikipedi*, was created in December 2002. As of May 19, 2010, there are 144,555 articles in Turkish Wikipedia. In this thesis, we have used Turkish Wikipedia for generating additional features as with English Wikipedia.

3.4 Semantic Relatedness

Semantic relatedness is a measure of the relation between two or more concepts. By itself, textual data does not always include necessary information about the context and different measures of semantic relatedness, utilizing external knowledge bases, help to increase the accuracy of feature representation.

3.4.1 Normalized Google Distance

World Wide Web enabled us to collect every piece of data about everything from physical items to abstract concepts in one place. With the rise of modern search engines (e.g. Google), the number of documents matching a query can be retrieved in less than quarter of a second, data transfer time included. The number of web pages indexed by Google is approaching 10^{10}.

Observing this potential, Cilibrasi et al.[12] created a semantic distance metric using a co-occurrence formula based on probabilities of Google events. In this framework, for a search query *x*, *probability p(x)* of event **x** is defined as $p(x) = |\mathbf{x}| \,/\, M$, where $|\mathbf{x}|$ is the number of web pages returned for the query, containing term *x* and *M* is the number of web pages indexed by Google. The joint event of term *x* and term *y* appearing in the same set of web pages is defined as $p(x, y) = |\{w : x, y \in w\}| \,/\, M$, where $\{w : x, y \in w\}$ denotes the set of web pages *w* containing both *x* and *y*.

Assuming that Google events capture all information about search terms, conditional probabilities can be used to derive a similarity between the terms, as in the following formula:

$$D(x, y) = min\{\, p(x|y)\,,\ p(y|x)\,\} \tag{3.5}$$

Equation (3.5) gives a similarity value between 0 and 1, where $D(x, y) = 1$ if the terms *x* and *y* have the same meaning.

If we want to compute the similarity between *horse* and *rider*, we use the values $p(horse) \approx 0.0058$, $p(rider) \approx 0.0015$, $p(horse, rider) \approx 0.0003$ obtained by querying Google and compute $p(horse|rider) \approx 0.02$ and $p(rider|horse) \approx 0.0517$ with the definition $p(x|y) =$

$p(x,y)\,/\,p(y)$. The similarity of *horse* and *rider* becomes $D(horse, rider) \approx 0.02$.

However, this formula is not sufficient because of two problems: First, smaller probability differences have a higher impact on similarity difference. Second, we cannot get higher similarity values for larger probabilities because of using absolute probabilities.

Normalized Google Distance (NGD) formula solves these problems by taking negative logarithm of conditional probabilities and normalizing the distance with the maximum log probability of x and y:

$$D_{ngd}(x,y) = \frac{max\{\, log 1/p(x|y)\,,\; log 1/p(y|x)\,\}}{max\{\, log 1/p(x)\,,\; log 1/p(y)\,\}}$$

for $p(x|y) > 0$, and $D_{ngd}(x,y) = \infty$ for $p(x|y) = 0$.

Substituting previous probability definitions for Google events, this formula is simplified to the formula below:

$$NGD(x,y) = \frac{max\{\, log f(x)\,,\; log f(y)\,\} - log f(x,y)}{log M - min\{\, log f(x)\,,\; log f(y)\,\}} \tag{3.6}$$

If $f(x)$, $f(y) > 0$ and $f(x,y) = 0$, then $NGD(x,y) = \infty$. If $f(x) = f(y) = 0$, NGD(x,y) is undefined. With this formula, previous distance between *horse* and *rider* becomes $NGD(horse, rider) \approx 0.443$.

Unlike Equation (3.5), Equation (3.6) yields a distance ranging between 0 and ∞. If $NGD(x,y) = 0$, term x and y have the same meaning in Google sense and also $NGD(x,x) = 0$, as expected. NGD is a *scale-invariant* measure, insensitive to the value of M, the indexed pages in Google.

In this thesis, we integrated NGD to cluster selection and merging operations, in an effort to integrate semantic features of cluster labels and documents to the clustering process.

3.4.2 Explicit Semantic Analysis

Without background knowledge, much of the information extracted from textual data lose its value. Upon seeing a word such as ”amazon”, human beings can generate all alternative meanings (company, female warrior, river, rainforest etc.) and resolve it to ”Amazon rainforest” upon observing contextual evidence such as ”jungle” or ”tropical”. To understand text as human beings, computers need to perform a similar task by using background knowledge.

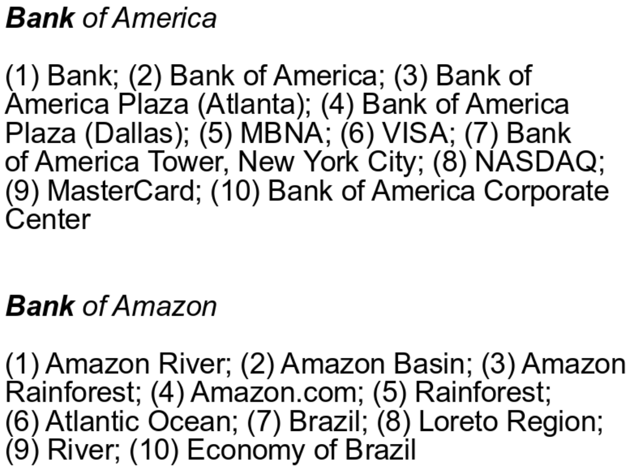
Wordnet is an effort to collect and serve such background knowledge for English. Short definitions, conceptual and lexical relations between items (nouns, verbs, adjectives, adverbs) are provided. These lexical items are grouped into cognitive synonyms (synsets), corresponding to different concepts. CYC is another huge effort in order to build a rule database of common sense knowledge. These projects are enormously helpful in certain domains but they cannot be directly used in a general text processing task.

Explicit Semantic Analysis (ESA) method, introduced by Gabrilovich and Markovitch [24], tries to tackle this problem utilizing a continously growing knowledge base, Wikipedia. Wikipedia is both a general and structured knowledge repository since it is an encyclopedia evolving at the speed of life (this is especially for popular topics).

Gabrilovich et al. proposed that every page in Wikipedia can be treated as concepts and these concepts can then be used to represent any given text. To achieve this, they first preprocess Wikipedia to remove overly specific articles (with less than 5 inlinks and outlinks), disambiguation and category pages, lists of Wikipedia, pages for specific dates, years or eras, together with very short articles with fewer than 100 non stop words. They also use a stop category list to remove articles belonging to a number of categories. The stop category list used in [25] is provided in Appendix A. Preprocessing also includes removing stop words, rare words (appearing in fewer than 3 articles) and stemming.

Templates (common page fragments shared between articles) and page redirects are resolved. After this anchor texts pointing to pages are collected and inserted into corresponding articles. This helps to better resolve synonyms, enabling an article to match a query even when the original article text does not contain query terms.

When preprocessing is complete, an inverted index is built. To avoid weak, spurious associations between concepts, the index is pruned with a sliding window technique. With a sliding window of 100 concepts and with a threshold of %5 acceptable decrease between the beginning and end of the sliding window, this technique limits the term vector at a point where the score decreases fast. According to statistics[25], 24% of concepts are retained on an index



Bank *of America*

(1) Bank; (2) Bank of America; (3) Bank of America Plaza (Atlanta); (4) Bank of America Plaza (Dallas); (5) MBNA; (6) VISA; (7) Bank of America Tower, New York City; (8) NASDAQ; (9) MasterCard; (10) Bank of America Corporate Center

Bank *of Amazon*

(1) Amazon River; (2) Amazon Basin; (3) Amazon Rainforest; (4) Amazon.com; (5) Rainforest; (6) Atlantic Ocean; (7) Brazil; (8) Loreto Region; (9) River; (10) Economy of Brazil

Figure 3.8: Example ESA queries ”bank of america” and ”bank of amazon”

of Wikipedia from November 11, 2005 and this pruning rate is similar for March 23, 2006 dump.

For a given string, query terms are extracted and relevant vectors are read from the inverted index. Then these vectors are averaged and this practically corresponds to disambiguation of terms. Top 10 concepts generated for ”bank of america” and ”bank of amazon” queries are shown in Figure 3.8.

Aside from regular feature generation, Gabrilovich et al. also generate features from a *second-order* interpretation of Wikipedia data. Since a link does not necessarily imply a strong relation, filtering linked concepts is essential. In order to do this, Gabrilovich et al. takes a number of top-scoring concepts and increases the scores of their out-linked concepts.

Let $ESA^1(t) =< w_1^{(1)}, \ldots, w_n^{(1)} >$ be the *first-order* interpretation of term *t*, the *second-order* interpretation of term *t* is defined as the following:

$$ESA^{(2)}(t) =< w_1^{(2)}, \ldots, w_n^{(2)} > \tag{3.7}$$

where

$$w_i^{(2)} = w_i^{(1)} + \alpha \cdot \sum_{\{j \mid \exists link(c_j, c_i)\}} w_j^{(1)} \tag{3.8}$$

α is a factor that decreases the effect of linked concepts and Gabrilovich et al. used $\alpha = 0.5$

in their experiments.

Based on the hypothesis that overly specific concepts are worse than general ones, Gabrilovich et al. applies a *concept generality* filter on the set of concepts generated using inter-article links. As an example, features generated for ”artificial intelligence” includes ”John McCarthy” (computer scientist) and ”Logic”. Since ”Logic” is more general, Gabrilovich et al. state that it is more useful.

The *concept generality* filter only allows a linked concept (target) if it is more general than the linking concept (source). As a heuristic, concept c_a is defined to be more general than c_b if:

$$log_{10}(\#inlinks(c_a)) - log_{10}(\#inlinks(c_b)) > 1 \tag{3.9}$$

Experiments on WordSimilarity-353 test collection and further tests with Reuters-21578, RCV1 and OHSUMED datasets show that ESA is currently the state-of-the-art in semantic relatedness methods. In this thesis, we investigated the use of ESA for enriching short texts with features generated using Wikipedia concepts.

3.4.3 Wikipedia Link-based Measure

Wikipedia Link-based Measure (WLM) is a low-cost semantic relatedness measure introduced by Milne et al. [84]. Explicit Semantic Analysis (ESA) requires processing of all text in Wikipedia. On the contrary, only the link structure and anchors are used in WLM. This makes it both a cheaper and more accurate alternative since it avoids a heavy text processing burden and has a stronger relation with the manually defined semantics of Wikipedia. ESA derives the relational weight between a term and a article by analyzing the segmentation of text into topics, ignoring all other information. However, WLM focuses on links to derive the weights from manual connections.

In WLM method, anchor text is used to gather candidate senses (Wikipedia articles) for a given term. This approach is quite effective in handling polysemy and synonymy. Only the anchors that are used more than %1 of the links going to that Wikipedia article are considered in this process.

Then relatedness between each pair of candidate senses is computed. First, cosine similarity of the outlink vectors (*outlinkRelatedness*) from each Wikipedia article is computed by using link counts to determine the weights. If s and t are the source and target, weight of a link is defined as:

$$w(s \rightarrow t) = \log(|\frac{W}{T}|) \tag{3.10}$$

where T is the set of all articles that link to t, and W is the set of all articles in Wikipedia. Second part of relatedness computation (*inlinkRelatedness*) is modeled after Normalized Google Distance (NGD) [12]. For two articles a and b, it is computed as:

$$sr(a,b) = \frac{\log(\max(|A|,|B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))} \tag{3.11}$$

where A and B are sets of all articles that link to a and b respectively and W is the set of all articles in Wikipedia. The relatedness between two Wikipedia articles is given as the average of these two measures:

$$relatedness(a,b) = \frac{outlinkRelatedness(a,b) + inlinkRelatedness(a,b)}{2} \tag{3.12}$$

Next, candidate senses are weighed evenly by commonness and relatedness in the original study [84]. However, in the current version (revision 89) of WikipediaMiner Toolkit [50], only relatedness is considered. Most related pairs of candidates (within %40 of the most related pair) are collected after this and the most common pair among them is picked. For handling relatedness caused by belonging to the same phrase, two anchors are combined to check whether such a phrase has inlinks. If such a phrase exists, a boost defined as below is added to the final relatedness score:

$$phraseBoost = \frac{\log(\ \#inlinks(phraseAnchor)\ )}{30} \tag{3.13}$$

Results from [84] indicate that WLM achieves an average correlation performance of 0.68, compared to average ESA score of 0.76 as shown in Table 3.1. In this thesis, WLM is used to compared text fragments using Wikipedia.

3.5 Wikification

Wikification is the process of detecting text fragments in a document that can be linked to a Wikipedia concept. Previous studies focused on the detection of such fragments but detecting

Table 3.1: Performance of ESA and WLM for three standard datasets

Dataset	ESA	WLM
Miller and Charles	0.73	0.70
Rubenstein and Goodenough	0.82	0.64
WordSimilarity-353	0.75	0.69

such fragments is not enough. The advantage of wikifying a document is resolving all interesting concepts in the document, including ambiguous ones. To solve this, Milne et al. [49] introduced a new wikification method that tries to tackle this problem.

In this algorithm, each possible sense of an anchor is compared to its surrounding context by using unambiguous anchors in the context. Link probability and relatedness, computed with Equation (3.11) from WLM, are averaged to assign a weight for each context term. Then these weights are used to calculate a weighted average, indicating the compatibility with the context. Context relatedness and commonness attributes are then used to learn a disambiguation classifier with C4.5 algorithm, achieving an F-measure of %97.1 on test data.

For link detection, link probability, relatedness to context, disambiguation confidence, generality (minimum depth of the article in Wikipedia category tree), link location (first occurrence) and spread (distance between first and last occurrences) attributes are used to train a link detector with C4.5 algorithm. Over a randomly selected set of 100 Wikipedia articles, this new link detector achieved an F-measure of %74.1.

This wikification algorithm is included in WikipediaMiner Toolkit [50] and in this thesis, we use this algorithm to enrich snippets with concepts from Wikipedia by wikification. We apply a minimum link probability of 0.5 to generate Wikipedia concepts.

CHAPTER 4

METHODS

In section 4.1 we will give information about the data used. In section 4.2 we will describe the methods we used to increase the performance over the data. We will give detailed results in Chapter 5.

4.1 Data Description

In this work, three datasets are used for the experiments. Two of them, AMBIENT [8] and ODP239 [9], are established datasets prepared for English and third is the one we created for Turkish. AMBIENT dataset contains clustering information for snippets returned from a search engine as of January 2008. ODP239 dataset provides information about snippets from the Open Directory Project as of 2009. ODP TR-30 is created automatically from Turkish snippets in ODP in the same manner as ODP239.

4.1.1 English SRC Data

Established datasets for search result clustering in English are described in this section. *Carrot2* project includes both of these datasets for benchmarking.

4.1.1.1 AMBIENT

AMBIENT (AMBIgous ENTries) is a dataset created by Carpineto et al. [8]. As of September 2007, top 100 search results (snippets) were collected from *Yahoo!* search engine for 44 topics selected from the list of ambiguous Wikipedia entries, provided in [80]. Each set of results for these topics are manually annotated according to their relations with extracted subtopics. These 44 topics were selected according to the following constraints [5]:

- Each topic has at least 5 subtopics to ensure the significance of subtopic information.
- Each ambiguous topic from Wikipedia has fewer than 35 subtopics. A clustering engine typically processes 100 results. This constraint is imposed to represent a topic better, with more of its Wikipedia subtopics appearing in this limited set of results.
- Each topic has least 2 subtopics from Wikipedia in the first 10 results from the search engine. This constraint ensures that Wikipedia subtopics for the topic and subtopics from the search engine significantly overlap.

AMBIENT dataset provides a common ground for SRC researchers and it is accepted by different research groups.[6]

Example clustering data from AMBIENT is provided in Appendix A for a topic.

Table 4.1: General Statistics on AMBIENT

Property	Current/average	Minimum	Maximum
Number of topics	44	-	-
Number of Wikipedia subtopics	790	-	-
Number of retrieved subtopics	349	-	-
Number of documents	4400	-	-
Number of relevant documents	2257	-	-
Number of Wikipedia subtopics per topic	17.9545	6	37
Number of retrieved subtopics per topic	7.9318	3	15
Number of relevant docs per topic	51.2954	18	90
Number of relevant docs per subtopic	6.4670	1	76
Words per title	5.86	1	40
Words per snippet	24.15	0	46
Characters per title	37.57	3	138
Characters per snippet	147.2	0	279

4.1.1.2 ODP239

ODP23 is a dataset created by Carpineto et al. [9]. Open Directory Project presents human judgement about web pages as a hierarchy including titles and snippets about these pages. This hierarchical classification can naturally be used for clustering studies after conversion.

In 2009, Carpineto et al. automatically extracted clustering information for 239 topics. The following rules were applied during this extraction process:

- Data is extracted from all English sections in ODP, including ”Kids and Teens” special directory and excluding ”Adult” directory and multilingual content in ”World” category.
- Second-level categories (e.g. ”Arts → Animation”) are considered as topics.
- Editorial links, symbolic directory links, links to directories in alternative languages (in ”World” category), related links, newsgroup links and aliases are ignored.
- Subtopics with less than 4 documents are ignored.
- Topics with less than 6 such subtopics are ignored.
- Every topic has a maximum of 10 subtopics and subtopics are selected according to their size.
- Every subtopic gets represented according to the ratio of its size to all documents under the topic. The following formula is used to scale subtopic size:

  $$|D'_{st}| = \lfloor \frac{|D_{st}|}{|D_t|} * 100 \rfloor$$

  $$|D''_{st}| = \max(|D'_{st}|, 4)$$

  where $|D_{st}|$ is the original subtopic size and $|D''_{st}|$ is the size scaled down to $[4, 100]$ range.

  For example, if there are 6 subtopics under topic T and if their sizes are given as $|ST_1| = 218$,$|ST_2| = 184$,$|ST_3| = 84$,$|ST_4| = 45$,$|ST_5| = 5$,$|ST_6| = 4$, $|ST''_1|$ becomes 40 and $|ST''_2|$ becomes 33.

Total number of documents in a topic can exceed 100 because of this representation considering subtopic size.

This recently created dataset was used in studies of Carpineto et al.

Example clustering data from ODP239 is provided in Appendix A for a topic.

Table 4.2: General Statistics on ODP239

Property	Current/average	Minimum	Maximum
Number of topics	239	-	-
Number of subtopics	2285	-	-
Number of (labeled) documents	25580	-	-
Number of subtopics per topic	9.5607	6	10
Number of docs per topic	107.0293	98	131
Number of docs per subtopic	11.1947	4	94
Words per title	3.27	1	22
Words per snippet	15.62	0	90
Characters per title	23.28	1	162
Characters per snippet	110.52	7	633

4.1.2 Turkish SRC Data: ODP TR-30

In our experiments, we used Turkish clustering data, extracted from ”World/Türkçe” directory [53] of ODP. This new dataset we created, called *ODP TR-30*, is available at [10]. Before this work, there was no dataset for evaluating SRC algorithms on Turkish. Observing that ODP had sufficient data for creating a dataset, we prepared necessary scripts [51] that enable us to build such a dataset for every language existing in ODP, with the same rules as in ODP239.

However, since Turkish data in ODP is limited, ODP TR-30 only includes 30 topics. Regarding ”World/Türkçe” directory as root, second-level categories are considered as topics, as in ODP239. For example, ”World/Türkçe/Bilgisayar/Programlama” is a topic according to this approach, when it is at the fourth level indeed.

Example Turkish clustering data from ODP TR-30 is provided in Appendix A for a topic.

Comparing Table 4.2 and Table 4.3, one can observe that titles and snippets in Turkish section

Table 4.3: General Statistics on ODP TR-30

Property	Current/average	Minimum	Maximum
Number of topics	30	-	-
Number of subtopics	264	-	-
Number of (labeled) documents	2957	-	-
Number of subtopics per topic	8.8	6	10
Number of docs per topic	98.566	41	121
Number of docs per subtopic	11.2	4	84
Words per title	2.36	1	13
Words per snippet	11.54	1	37
Characters per title	16.54	2	94
Characters per snippet	91.31	6	279

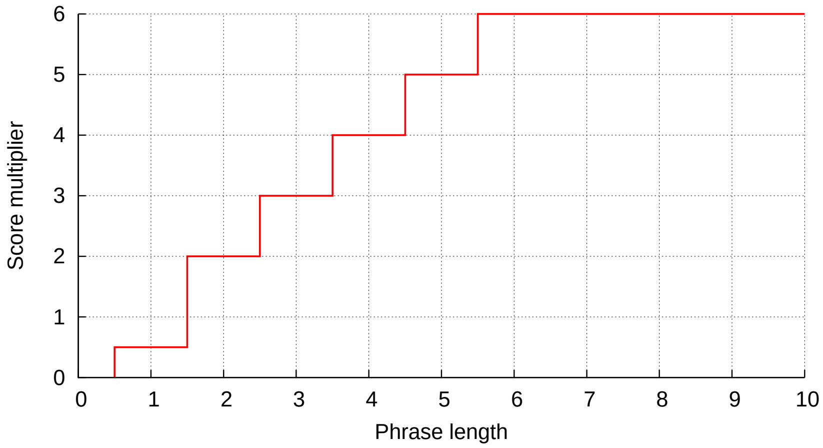
of ODP are slightly shorter than English. Moreover, minimum number of documents per topic is significantly lower for Turkish. This can be attributed to a smaller community with limited resources, the delay between creation of English and Turkish sections and the dominance of English on the Web.

4.2 Methods Used

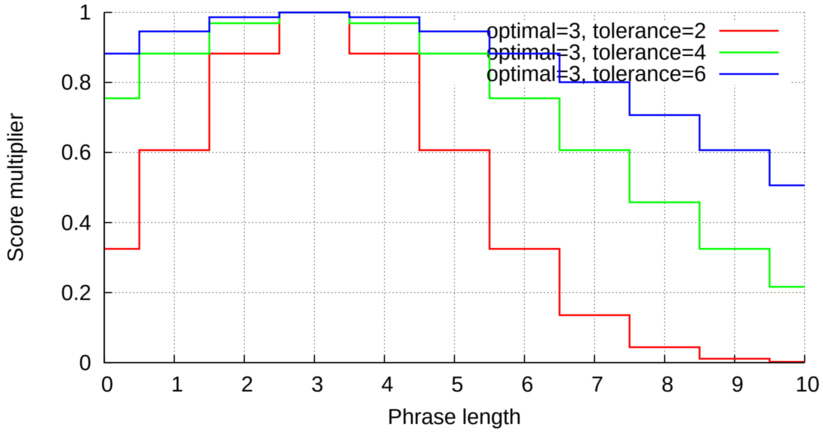
4.2.1 Baseline STC Implementation

As a baseline, STC [87] implementation from *Carrot2* clustering framework was used. However this implementation did not apply stemming as in classical STC so we modified it to use default stemming feature provided in *Carrot2*. Carrot2 uses Snowball for stemming but it allows custom stemmers to be added. For stemming Turkish words, we compared the performance of Turkish stemmer in Snowball and Zemberek. Upon observing that Zemberek performs slightly better, we decided to use Zemberek for Turkish.

Phrase length boost function used in Carrot2 is slightly different from the one used by Zamir et al, described in [86]. Instead of a linear function turning to a constant after a predefined phrase length as in Figure 4.1(a), a Gaussian function controlled by *optimum phrase length* and *tolerance* parameters is used, behaving as in Figure 4.1(b).



(a) Function from Zamir et al.



(b) Function from Carrot2

Figure 4.1: Phrase length boost functions from Zamir et al. and Carrot2

4.2.2 Wang et al. [73]

Base cluster merging formula in STC [87] prevents merging of subsets and dominated clusters. Moreover, non-overlapping parts of base clusters are not considered. Even when documents in two clusters are textually similar, they cannot be merged if there is not sufficient overlap.

As a solution to these issues, Wang et al. [73] proposed a new merging algorithm. They created a dataset from Google results by manually judging relevance of the results. According to their experiments on this dataset, this algorithm provides better precision, with top 20 documents covering more than %40 relevant documents whereas original STC stays below %40.

We implemented this algorithm in Carrot2 and performed experiments on AMBIENT, ODP239 and ODP TR-30 datasets to further investigate the effects.

According to our experiments, this new merging algorithm performed worse than the original in STC. Both ideas in [73] make sense but the modification of overlap formula to a more relaxed version causes problematic merging scenarios. Here are the reasons behind this:

- Base clusters labeled with a single word have a natural advantage in cluster cardinality. As a consequence, most supersets are single word clusters. This forces the algorithm to limit its representation with a higher emphasis on single words.

- Since overlap is only defined as shared documents between clusters, big clusters have direct advantage over small clusters with fewer irrelevant documents. When documents of a base cluster are fully subsumed by a bigger cluster, one cannot guarantee their relevance. Compared with the old merging method, small clusters with only 2 or 3 documents have a higher probability of getting merged to a big cluster. Repeated merging in this way yields highly contaminated clusters, even resulting in one cluster containing everything. If not properly eliminated, contextually irrelevant but frequently appearing words such as ”known” or ”used” can get ranked as top clusters.

To cope with these problems, we tested the following strategies:

4.2.2.1 Using Average Intra-cluster Distance

At least for base clusters that are not identical, pre-merge and post-merge states can be compared considering document compatibility. Intra-cluster distance (or similarity) is a common measure of cluster quality used in traditional clustering algorithms like K-Means. In this case, we used average intra-cluster distance to check whether the intersection of two clusters had equivalent or better quality than the original clusters themselves.

For a cluster C, average intra-cluster distance is computed with the following formula:

$$AvgIC(C) = \sum_{i \in C} \sum_{j \in C, j \neq i} \vec{doc}_i \cdot \vec{doc}_j \tag{4.1}$$

where $\vec{doc}_i$ denotes the term vector for i^{th} document.

If merging criterion defined by Wang et al.[73] is satisfied for two base clusters A and B, merging is only allowed if:

$$AvgIC(A \cap B) < AvgIC(A) \quad and \quad AvgIC(A \cap B) < AvgIC(B) \tag{4.2}$$

We also applied this control strategy on STC with the original merging algorithm.

4.2.2.2 Enforcing Connectedness of Merged Clusters

The purpose of merging is combining strongly linked base clusters about a common topic in one cluster. Document overlap amounts in STC define a direct graph with edge weights in [0,1] range.

Classical merging algorithm of STC only considers strongly linked base clusters whereas the new merging algorithm allows weak links.

Considering the directed graph of overlaps, one does not expect to find strongly linked, separated components related with only weak, one-way links to a node after a good merge operation. However, relaxing the overlap constraint can result in a big cluster (created because of a common but unimportant word) absorbing many small clusters in this manner.

An example for this situation is given in both the directed overlap graph in Figure 4.2 and undirected similarity graph in Figure 4.3. Using an appropriate common word threshold (e.g.

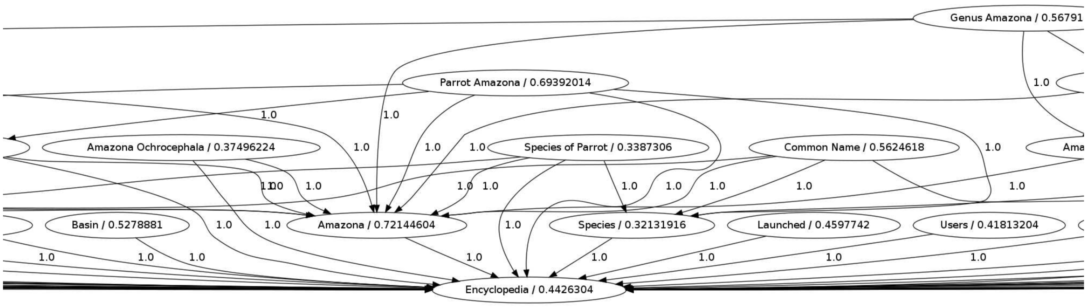


Figure 4.2: ”Encyclopedia” part of the directed overlap graph for *amazon* query on Wikipedia

ignoring words if they appear in more than 40% of the collection, instead of 90%) can help with this problem. However, the problem is still embedded in the merging algorithm and can hurt precision according to case.

Observing that properly merged clusters can often be decomposed only to their individual members instead of smaller subgroups or only one group, we tested two strategies to solve this: Checking for more than one isolated, connected component and checking for a node not linked to any other node aside from its weak connection the root. (Root is the node that dominates others, such as ”Encyclopedia” node in Figure 4.2.)

4.2.2.3 Using Normalized Wikipedia Distance (NWD) to measure relevance

Performing clustering without considering semantic relations is problematic, especially in STC. In both the original merging algorithm and the new one proposed by Wang et al. , merging phase is susceptible to errors caused by dominating but meaningless supersets and such associations.

We propose using a semantic distance measure to correct these errors. Normalized Google Distance (NGD) [12] was shown to be useful for computing semantic distance between single words and short phrases. However, NGD method cannot be used for relating long phrases, sentences and documents. Our case with cluster labels satisfies this requirement.

In NGD method, querying Google index would be a bottleneck because of network delays. To cope with this, we used Wikipedia as our knowledge base.

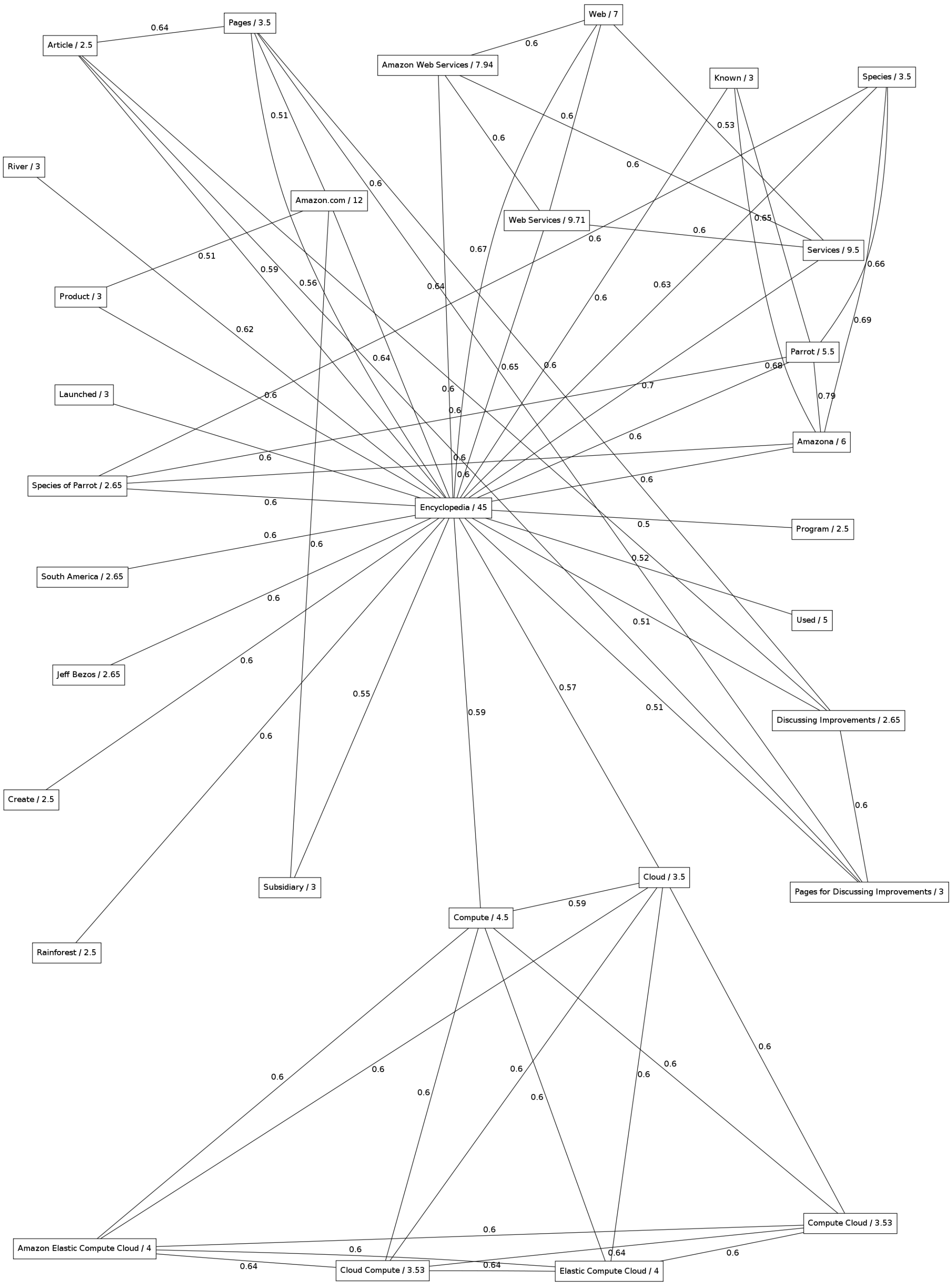


Figure 4.3: Similarity graph for *amazon* query on Wikipedia after running STC with merging method of Wang et al.

The use of NWD in merging phase of STC is straightforward: Since we need to check whether two cluster labels are relevant or not, we first compute NWD standard deviation and NWD mean of 1000 semantically related pairs, randomly selected from ”See Also” sections, ”Related Articles” sections and corresponding templates of Wikipedia articles. One standard deviation above the mean is identified as threshold and two labels are assumed to be semantically related only when their NWD distance is below this threshold. Then, semantically unrelated merge operations are canceled.

In our experiments, we tested this semantic noise elimination strategy on STC with both the old merging algorithm and the one proposed by Wang et al.

Sample pairs from our list of related pairs used to select NWD threshold is provided in Appendix A.

4.2.3 Boosting Document Scores According to Coverage

Crabtree et al. [13] focused on cluster coverage problem in STC and proposed selecting clusters according to their coverage with 1-step look-ahead. This strategy eliminates sub-optimal combinations covering fewer documents.

In this light of this strategy, we wanted to investigate better cluster combinations and increase coverage. We experimented with a simpler strategy: scoring documents according to the number of base clusters covering them.

If $occur_i$ denotes the number of occurrences of document i in all base clusters, score for a base cluster is computed as:

$$s(B) = f(|P|) \cdot \left(|B| - \frac{\sum_{i \in B} occur_i}{\max \{ i \in B \mid occur_i \}} \right)$$

where B is the set of documents in base cluster and $|P|$ is the *effective phrase length*.

4.2.4 Part-of-Speech: Filtering with Noun Clauses in Turkish

Common, non-descriptive words (e.g. ”used”, ”known”) significantly decrease the quality of clustering in STC. In a previous study [3], part-of-speech information was considered by using a simple form of POS tagging, based on a hashed morphological lexicon for English.

Cluster labels must only contain adjectives and nouns, including proper names. A label is allowed only if each of its words is either defined as adjective or noun in the list of lexicons, or the list does not contain the word.

We applied a similar strategy to STC in Turkish by using Zemberek. Compared to English, agglutinative character of Turkish allows us to identify nouns with better precision from a single word. For SRC task, the aim of using POS tagging is essentially identifying noun clauses. Observing that noun clauses in Turkish can be identified easier than English by checking the last word, we tagged the last word of every cluster label with Zemberek and filtered it out if the last word was not a noun. We performed our experiments on *ODP TR-30* dataset.

4.2.5 Combining Semantic Relatedness with STC

As in Section 4.2.2.3, semantic relatedness measures can be used to improve various phases of STC. In this section, our ESA implementation is described and the modifications on STC using that can be applied by using either ESA or WLM are explained.

4.2.5.1 Explicit Semantic Analysis (ESA)

ESA [25] provides a way to utilize knowledge organized by humans and retrieve additional features for a text fragment using a concept mapping. This method is called *explicit* because it represents the input using concepts based on human cognition rather than implicit, abstract concepts as in Latent Semantic Analysis (LSA).

In ESA implementation of Gabrilovich et al. [25], documents are first preprocessed to remove overly specific concepts and too short articles. Articles are filtered according to the following rules:

- Articles that have fewer than 100 non stop words are discarded.
- Articles that have fewer than 5 incoming and 5 outgoing links are discarded.
- Articles describing dates (including years, centuries etc.) are discarded.
- Disambiguation pages, category pages and the like (e.g. lists) are discarded.

Redirections and templates are resolved and anchor texts (including link anchors without a surface form) pointing to remaining articles are added to their targets.

We used our own implementation of ESA, available as an open source project at Github. [83] We have used Wikiprep to preprocess Wikipedia dumps. Wikiprep performs redirect resolution, template inclusion and converts Wikipedia markup into HTML. Then we used Python scripts to perform filtering and anchor processing, Lucene to index data and a few Java classes to process the Lucene index and apply the same post-processing (pruning etc.) steps as Gabrilovich et al.

Here are the actions of our scripts in further detail:

- Only the articles from Main namespace are considered.
- Articles that have fewer than 100 unique word stems in their content (not the raw Wiki markup but clean text) are discarded.
- Articles with titles in the forms given in Table 4.4 are discarded:
- Articles including disambiguation and set index template tags in the forms given in Table 4.5 are discarded. Disambiguation and set index template tags for English Wikipedia are listed in [18]. We have used this list as of June 12, 2010 by adjusting tags used by Wikiprep to detect disambiguation pages.
- Articles belonging to a stop category list are discarded. For 2005-2006 Wikipedia dumps, the stop category list of Gabrilovich et al. provided in Appendix A is used. For 2009 dump, we added all categories including ”disambig” in their title or start with ”Lists of” and ”Indexes of” phrases to this stop category list and removed the categories in the old list of Gabrilovich et al. that are deleted from Wikipedia.

  For Turkish Wikipedia, disambiguation tags are listed in [69] and disambiguation categories are found with a search as in [68].
- Articles having fewer than 5 incoming and 5 outgoing links are discarded. The number of inlinks and outlinks are computed by considering all articles in Main namespace.
- Redirects are resolved. All anchor text of links pointing to a target article are collected and inserted into the target article text. This allows retrieving a concept by its alternative names (e.g. ”JFK” instead of ”John F. Kennedy”).

Table 4.4: Wikipedia titles used for filtering

English Wikipedia	Turkish Wikipedia
title (disambiguation)	*title* (anlam ayrımı)
	title (anlam ayrım)
month day	*day month*
year	*year*
year BC/BH/BP	MÖ *year*
year AD/AH/AP	MS *year*
*n*s	
*n*th century/millenium	*n.* yüzyıl
*n*th century/millenium BC/BH/BP	MÖ *n.* yüzyıl
*n*th century/millenium AD/AH/AP	MÖ *n.* yüzyıl
Year in ...	
Year BC/BH/BP in ...	
Year AD/AH/AP in ...	
*n*th century/millenium in ...	
*n*th century/millenium BC/BH/BP in ...	
*n*th century/millenium AD/AH/AP in ...	
List of *title*	
Lists of *title*	
Index of *title*	

After these preprocessing steps, we index all remaining articles using *Lucene* search engine library. Then we scan this Lucene index to apply TF-IDF weighing model used in the study of Gabrilovich et al. [25], prune the resulting vectors and record each vector to the database. We used a sliding window threshold of $thres = 0.005$ in our implementation, as Gabrilovich et al.

For a given text fragment, the regular feature vector is retrieved by querying the database and using the matching scores for every concept (Wikipedia article) in the results.

We also implemented necessary classes to compute secondary interpretation of an ESA vector but this is not required to compute semantic relatedness. To compute regular ESA vector for a text fragment, vectors for every term in the text are averaged to combine a single concept vector. Semantic relatedness between two text fragments is computed as the cosine similarity between combined ESA vectors for these fragments.

We tested our implementation of ESA using WordSimilarity-353 dataset with preprocessed

Table 4.5: Wikipedia template tags used for filtering

Wikipedia edition	Template tags
English Wikipedia	Airport disambig, Callsigndis, Church disambig, Disambig, Disambig-Chinese-char-title, Disambig-cleanup, Geodis, Hndis, Hndis-cleanup, Hospitaldis, Hurricane disambig, LatinNameDisambig, Letter disambig, Letter-NumberCombDisambig, Mathdab, MolFormDisambig, NA Broadcast List, Numberdis, Roaddis, Schooldis, WP disambig, Dab, Disamb, Disambiguation, CJKVdab, Cleanup disambig, Cleanup-disambig, CleanupDisambig, Dab-cleanup, Dabclean, Disamb-cleanup, Disambig-CU, Geo-dis, Geodab, Bio-dab, Hndab, Hndisambig, Namedab, Numdab, Roadab, Roadis, Shorcut disambig, Shortcut disambig, WP Disambig, WP-disambig, SIA, Sia, Given name, Hawaiiindex, Mountainindex, Plant common name, Disambig-plants, Shipindex, Sportindex, Surname
Turkish Wikipedia	Anlam ayrımı, Anlam ayrım

dump from Gabrilovich et al. (20051105 dump) and achieved a similar Spearman correlation of 0.737 (compared to 0.74 for the same dump).

4.2.5.2 Filtering base clusters using semantic relatedness

Unlike NWD, ESA allows to compute similarities for arbitrary text fragments. We also extended WLM to provide such functionality. Using ESA or WLM, base clusters generated in STC can be checked whether meanings of their labels are consistent with meanings of their documents. Filtering out semantically inconsistent labels before merging step can significantly decrease the noise introduced by merge phase.

Checking semantic consistency for every document inside a cluster would increase the complexity of the algorithm. Because of this, we only check the consistency between the label and a number of randomly selected documents in the cluster. We use a sample ratio of 0.03 for each cluster and take at least 2 samples.

4.2.5.3 Merging phase modified using semantic relatedness

As in Section 4.2.2.3, labels to be merged can be checked, this time using ESA or WLM. We have used the same related pair set previously created using ”See Also” section, ”Related Articles” section and corresponding templates.

4.2.6 Clustering by using Wikipedia concepts as features

Since the input of SRC algorithms consist of short text fragments, we thought that these algorithms could benefit from using an enriched representation of text.

4.2.6.1 Wikification clustering

As a simple clustering strategy, we tested the idea of using Wikipedia concepts generated by wikification [49] as clusters for comparison purposes. We directly used the titles of Wikipedia concepts as cluster labels.

4.2.6.2 Hierarchical clustering using Wikipedia concepts

Influenced by the work of Gabrilovich et al, Banerjee et al.[2] have previously used a simple form of ESA to cluster short articles from Google News. They indexed Wikipedia with Lucene and for each text fragment, they queried the index separately with title and description. Combining 10 results for title query and 10 results for description query, they represented 20 Wikipedia concepts in a vector where weights correspond to frequencies of concepts in the list. Banerjee et al. then augmented this vector with the original term frequency vector to achieve a better clustering.

Inspired by this study, we augmented the concept vectors generated by wikification [49] with term frequency vectors and performed group-average hierarchical clustering to test the effectiveness of semantic features in SRC task. We clustered input documents but did not perform cluster labeling.

CHAPTER 5

RESULTS AND DISCUSSIONS

This chapter presents the results and analysis of the experiments described in Chapter 4.

For measuring the effectiveness of the methods presented in Chapter 4, different statistical measures are employed. For each topic of every dataset, precision, recall and F1 values are calculated and the average values over all clusters are reported for a dataset.

Precision, recall and F1 score are defined using True Positive, True Negative and False Negative amounts, which are described in Table 5.1.

Table 5.1: Precision, Recall and F1

	In True Cluster	Not in True Cluster
In Cluster	True Positive	False Positive
Not in Cluster	False Negative	True Negative

For cluster C_i on true cluster TC_i, precision denotes the ratio of elements that are correctly put into C_i (as they exist in TC_i) to all elements in C_i. This measure says nothing about whether all elements in TC_i exists in C_i.

Recall denotes the ratio of elements in C_i to all elements in TC_i. A perfect recall score of 1.0 means that all elements in TC_i are included in C_i. However it says nothing about the elements that should not have existed in C_i.

Separate information from Precision and Recall metrics are combined in F-score as their harmonic mean, complementing each other.

If TC_i is a true cluster from true cluster set TC and C_i is a cluster from cluster set C, these definitions correspond to the following in our case:

For each pair of TC_i and C_i, traditional definitions of these sets are applied as in Equation (5.1).

$$\text{True Positive} = C_i \cap TC_i \tag{5.1a}$$

$$\text{False Positive} = C_i \setminus TC_i \tag{5.1b}$$

$$\text{False Negative} = TC_i \setminus C_i \tag{5.1c}$$

Using the definitions from Equation (5.1), precision, recall and F-score for cluster C_i on true cluster TC_i are given as in Equation (5.2).

$$\text{Precision} = \frac{|TruePositive|}{|TruePositive \cup FalsePositive|} \tag{5.2a}$$

$$\text{Recall} = \frac{|TruePositive|}{|TruePositive \cup FalseNegative|} \tag{5.2b}$$

$$F = 2 \cdot (\text{precision} \cdot \text{recall})/(\text{precision} + \text{recall}) \tag{5.2c}$$

To compute precision, recall and F-score achieved for a true cluster TC_i, the best matching cluster C_{best} is considered and selected according to F-score. Cluster C_i with the best F-score becomes C_{best} in statistical computations for TC_i. For every TC_i, precision, recall and F-score computed with C_{best} becomes the score for TC_i.

For a topic t, precision, recall and F-score are defined as average of the scores for each true cluster, as in Equation 5.3 where TC is the set of true clusters in topic t:

$$\text{Avg. Precision} = \frac{\sum_{c \in TC}(\text{Precision}_c \cdot |c|)}{\sum_{c \in TC} |c|} \tag{5.3a}$$

$$\text{Avg. Recall} = \frac{\sum_{c \in TC}(\text{Recall}_c \cdot |c|)}{\sum_{c \in TC} |c|} \tag{5.3b}$$

$$\text{Avg. F} = \frac{\sum_{c \in TC}(\text{F}_c \cdot |c|)}{\sum_{c \in TC} |c|} \tag{5.3c}$$

We use two additional metrics for measuring clustering performance.

For a given cluster, contamination denotes the distribution of elements from multiple true clusters. In a perfect contamination value of 0, the cluster consists of elements from only a single true cluster. In the worst case value of 1, elements from each true cluster are evenly distributed in the cluster. This measure is calculated as in [76].

Mutual information (MI) measures the increase in the amount of information about true clusters by knowing the clusters. In a perfect MI value of 1, the clusters completely match true clusters. In the worst case value of 0, clustering is random with respect to true clusters. However single document clusters yield maximum MI. Normalization incorporates entropies to remedy this, yielding Normalized Mutual Information (NMI) metric. It is calculated as described in [45].

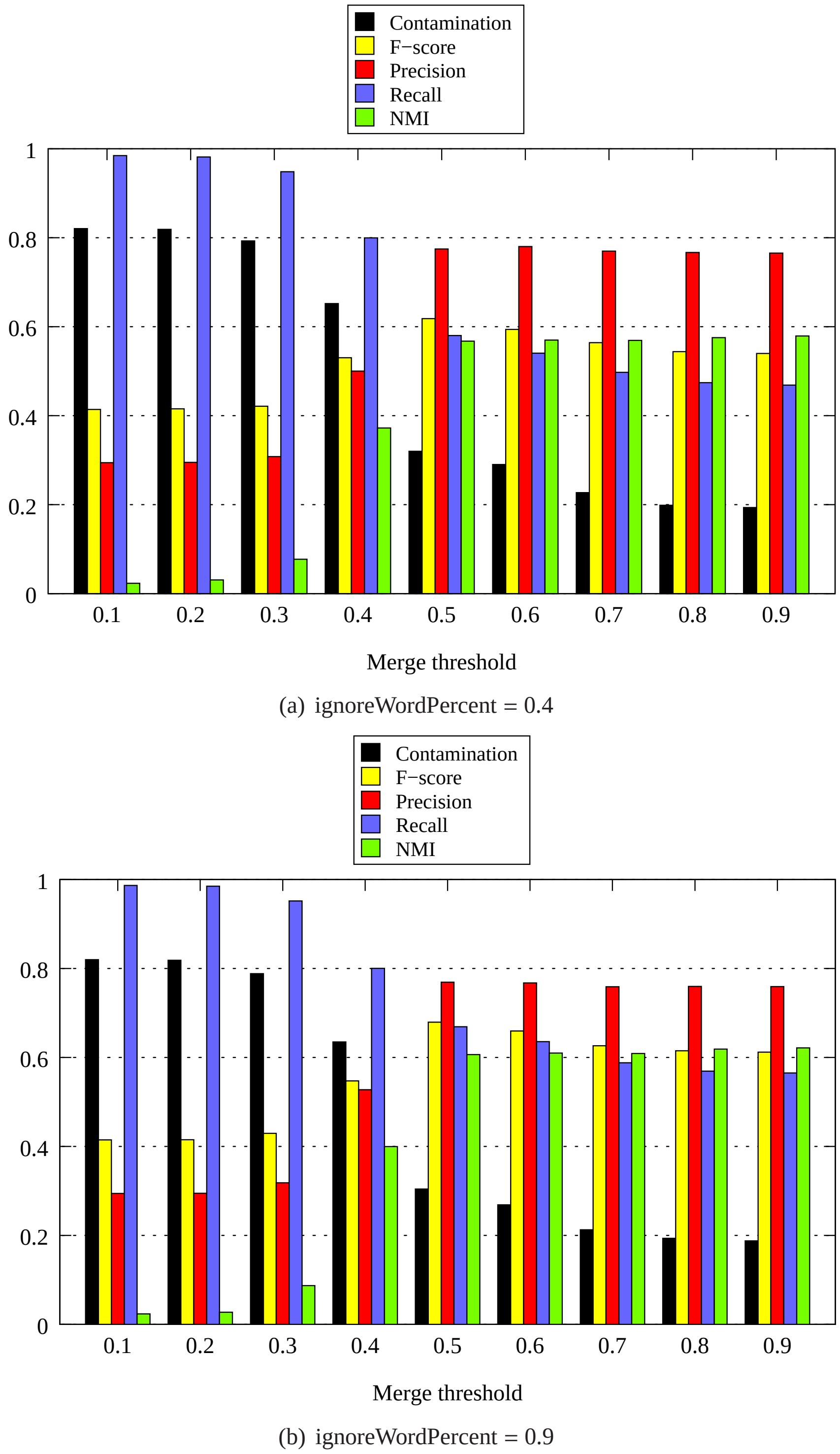
5.1 STC Baseline

We have used the STC implementation from Carrot2 framework as reference. With this implementation, it is possible to tune various parameters of STC, such as word ignore percentage and merging threshold. Word ignore percentage (ignoreWordPercent) denotes the ratio of snippets containing a word to all search results. Words that appear in too many documents, such as the query and other contextual phrases, are eliminated with this parameter.

To show the effect of these parameters on STC and to provide a baseline, we have run STC with two different cases, ignoreWordPercent = 0.4 and ignoreWordPercent = 0.9, and variable merging thresholds. The results shown in Figure 5.1 are generated using AMBIENT dataset.

In the original STC [86], a threshold ratio of %40 was set for ignoreWordPercent. From Figure 5.1 shows that this filtering strategy can be too aggressive and actually decreases F-score. Enforcing a maximum ratio of %40 eliminates highly dominant cluster labels and indeed these labels can sometimes be good cluster labels properly representing the topic. Further results are shown in Figure 5.2 which are generated using ODP-239 dataset.

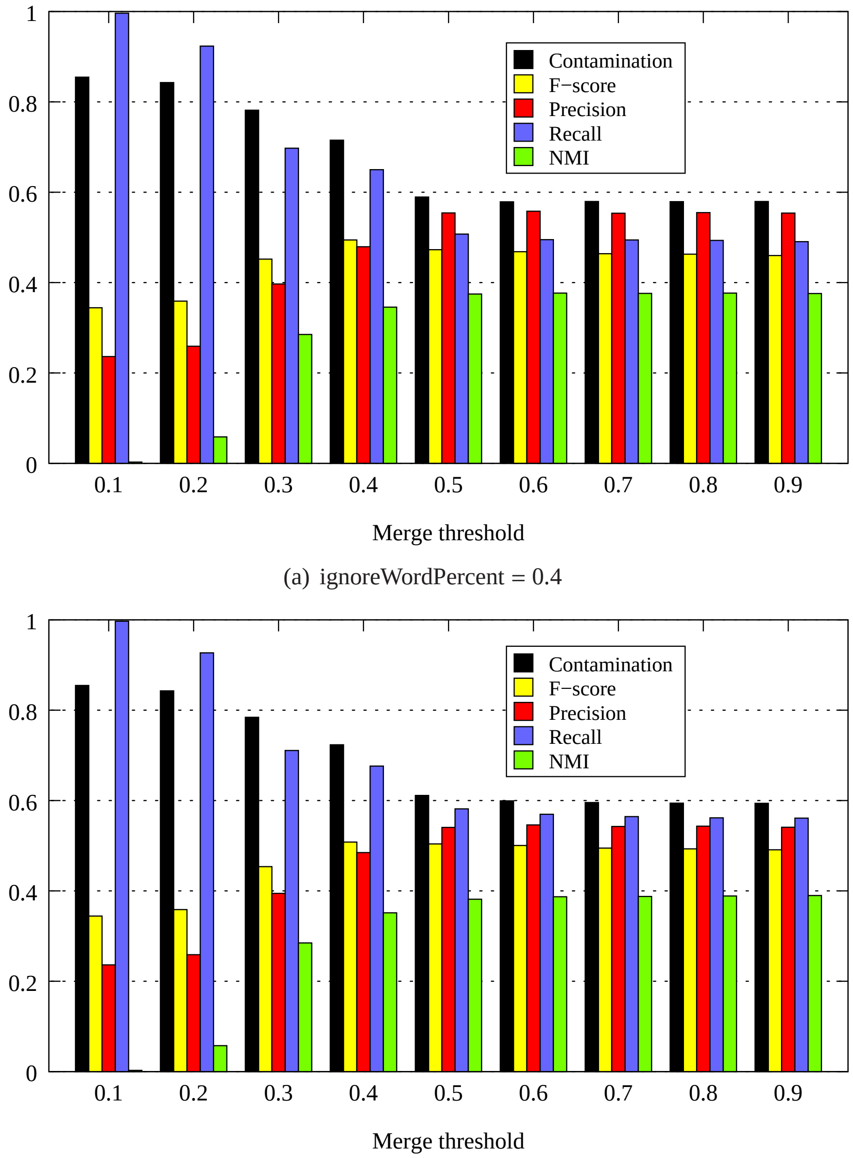
Zamir et al. [86] used a merge threshold of 0.5 in their study. In the light of this, we select this threshold to reporting the results for STC. Table 5.2 shows the STC performance on AMBIENT and ODP-239 datasets. We have also observed that selecting 0.5 as threshold provided the best performance in both datasets.



(a) ignoreWordPercent = 0.4

(b) ignoreWordPercent = 0.9

Figure 5.1: STC benchmarks on AMBIENT dataset with top 10 clusters



(a) ignoreWordPercent = 0.4

(b) ignoreWordPercent = 0.9

Figure 5.2: STC benchmarks on ODP-239 dataset with top 10 clusters

Table 5.2: STC performance

Dataset	Merge threshold	Contamination	F-score	Precision	Recall	NMI
AMBIENT	0.5	0.3042	0.6792	0.7693	0.6689	0.6064
ODP-239	0.5	0.6112	0.504	0.5404	0.5815	0.3814

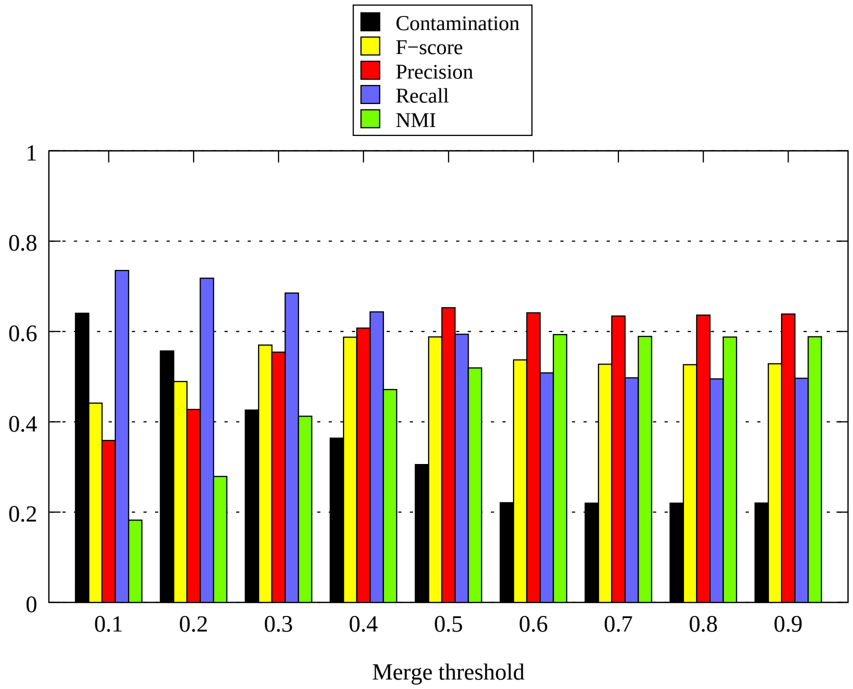
5.2 Wang et al.

To test the effects of the new similarity formula introduced by Wang et al.[73], we implemented this modified version of STC in Carrot2. Wang et al. [73] used a similarity threshold of 0.5 in their experiments. In the light of this, we select this threshold for reporting the results for their method. Table 5.3 shows the results from AMBIENT and ODP-239 datasets. To investigate the whole spectrum, we have also generated the results using different thresholds, as in Figure 5.3.

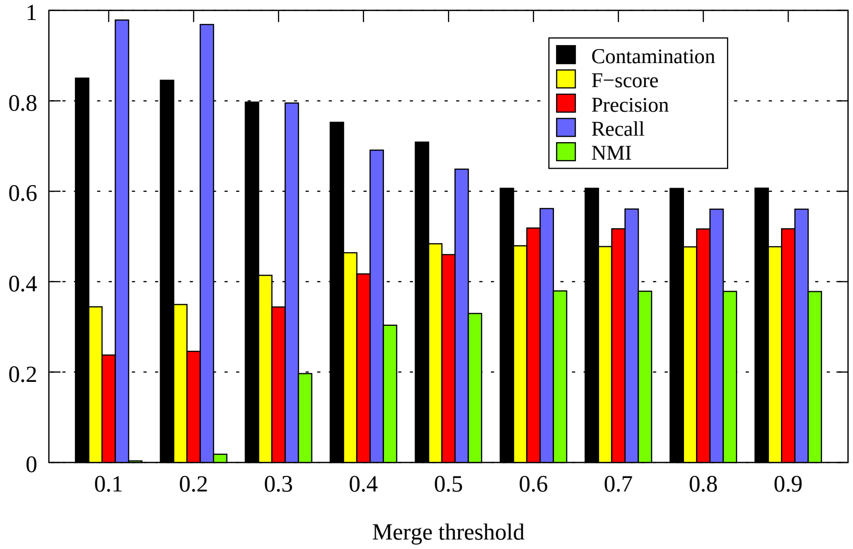
Table 5.3: Wang et al. performance

Dataset	Merge threshold	Contamination	F-score	Precision	Recall	NMI
AMBIENT	0.5	0.3052	0.588	0.6528	0.5939	0.5192
ODP-239	0.5	0.7086	0.4839	0.4599	0.6488	0.3295

According to these results, the new similarity formula does not improve STC. As previously mentioned in Section 4.2.2, subsumption does not always imply relevance. Single word cluster labels have a natural advantage in cluster cardinality. Compared to the conservative merging rule from STC, overlap part of similarity formula is worse because of this. Requiring textual similarity with this overlap formula limits the amount of noisy clusters but still, the problems with overlap formula are reflected into the results. Figure 5.4 shows this better, comparing STC and the method of Wang et al. side by side.

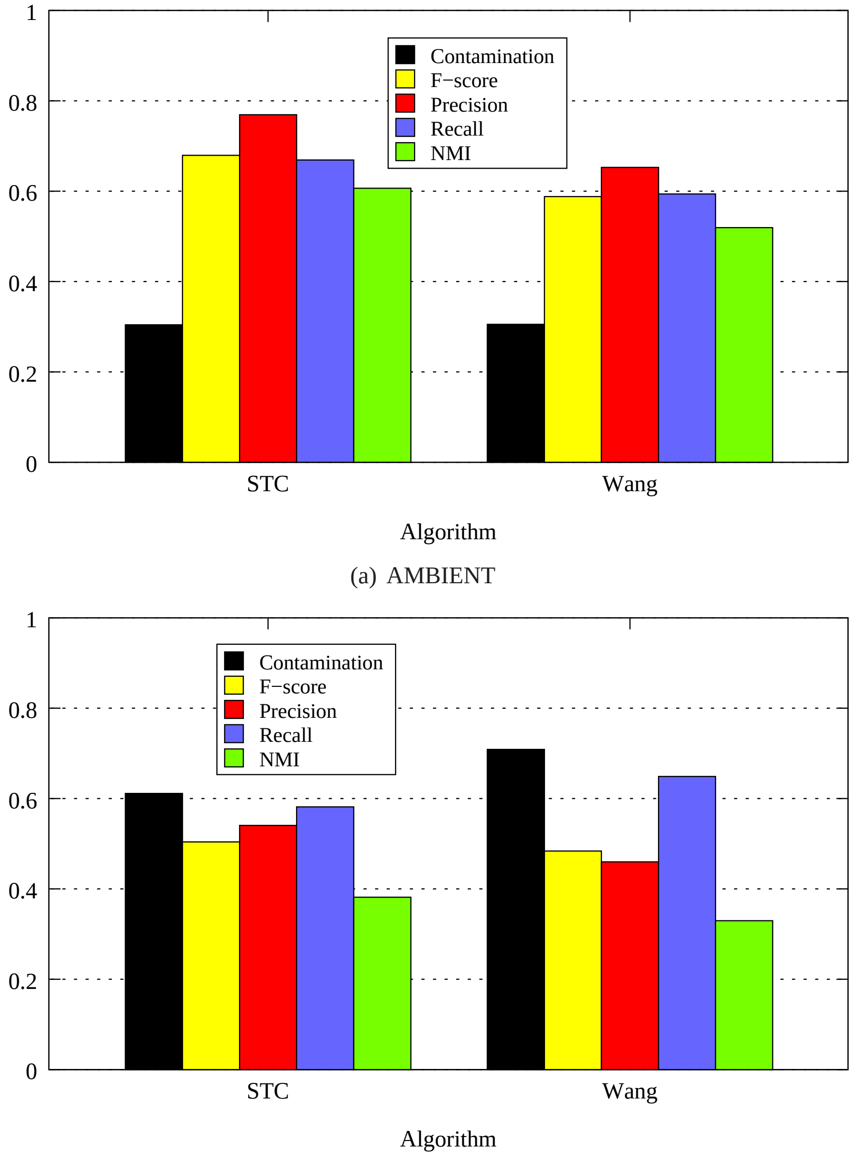


(a) AMBIENT



(b) ODP-239

Figure 5.3: Wang et al. benchmarks with top 10 clusters, ignoreWordPercent = 0.9



(a) AMBIENT

(b) ODP-239

Figure 5.4: Comparison of STC and the method of Wang et al.

5.2.1 Using Average Intra-Cluster Distance

For handling the erroneous merge operations caused by new similarity formula of Wang et al, we thought that the effect of these merge operations could be checked to see if they yield a worse cluster than the original state. We introduced a simple condition to do this: the intersection of merged clusters should not have a bigger average intra-cluster distance than the original clusters themselves. The reasoning behind this strategy is as follows:

An increase in intra-cluster distance is expected from union of two clusters, because more cluster members increase the variance in the cluster. However, common members of both clusters must have better compatibility with each other, to imply an advantage over being in their original cluster. Table 5.4 and Figure 5.5 shows that this strategy helps to reduce contamination and increase NMI, with a significant trade-off in Recall.

Table 5.4: Average intra-cluster distance check on Wang et al. method and STC, on AMBIENT dataset

Method	Contamination	F-score	Precision	Recall	NMI
Wang	0.3052	0.588	0.6528	0.5939	0.5192
Wang-AvgIntra	0.2201	0.536	0.6466	0.5043	0.5817
STC	0.3042	0.6792	0.7693	0.6689	0.6064
STC-AvgIntra	0.1893	0.6158	0.7566	0.5712	0.6236

5.2.2 Enforcing Connectedness of Merged Clusters

Consider we query our clustering search engine with ”apple” query and results include both ”steve jobs” and ”steve wozniak”. There are snippets related to only ”steve jobs”, only ”steve wozniak” and both, with Apple Inc. connection. With the new overlap definition from Wang et al, the clustering engine would pick ”steve” as the superset of ”steve jobs” and ”steve wozniak”. Most of the time, including textual similarity in the equation helps us to avoid directly merging such clusters without considering the textual difference between snippets. However, in this case the context of the subsets ”steve jobs” and ”steve wozniak” are highly similar. All snippets from these clusters would probably contain terms about software development, Apple Inc. This would yield a high textual similarity between elements unique to these sub-

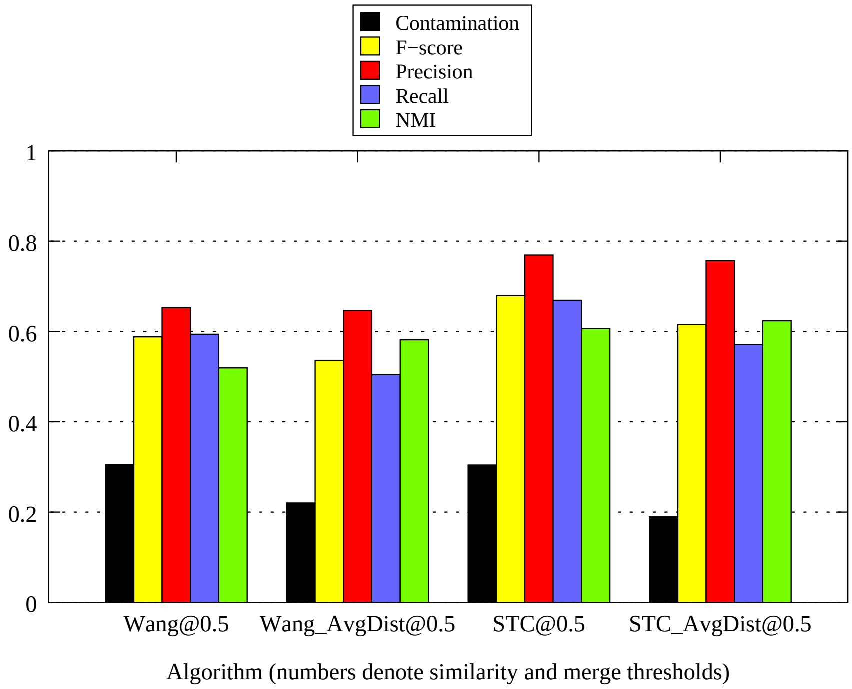


Figure 5.5: Average intra-cluster distance check on Wang et al. method and STC (AMBIENT, top 10 clusters, ignoreWordPercent = 0.9)

sets and yield ”steve” cluster in the end, eliminating discrimination between ”steve jobs” and ”steve wozniak”.

To avoid such a scenario, grouping of the elements inside the superset (e.g. ”steve”) can be investigated. We implemented two strategies to achieve this. Our first strategy was seeking only one connected component inside the superset, removing it otherwise. Table 5.5 and Figure 5.6 shows the result of this approach.

Table 5.5: Connected component check on Wang et al. method and STC, on AMBIENT dataset

Method	Contamination	F-score	Precision	Recall	NMI
Wang	0.3052	0.588	0.6528	0.5939	0.5192
Wang-ConnComp	0.2768	0.5357	0.6729	0.5129	0.5058
STC	0.3042	0.6792	0.7693	0.6689	0.6064
STC-ConnComp	0.2817	0.5742	0.8103	0.5112	0.5341

Thinking that this strategy may be too aggressive, we checked dependencies between child

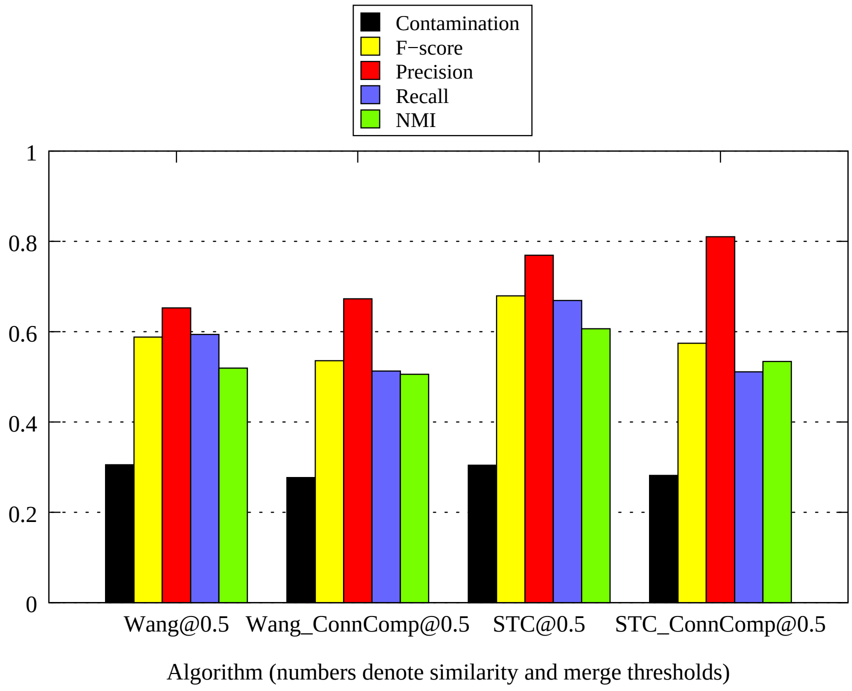


Figure 5.6: Connected component check on Wang et al. method and STC (AMBIENT, top 10 clusters, ignoreWordPercent = 0.9)

nodes. If all the nodes inside the superset had links to another node in the set, we accepted the superset as valid. Table 5.6 and Figure 5.7 shows the result of this second strategy.

Table 5.6: Connected component check on Wang et al. method and STC, on AMBIENT dataset

Method	Contamination	F-score	Precision	Recall	NMI
Wang	0.3052	0.588	0.6528	0.5939	0.5192
Wang-IndChild	0.288	0.5407	0.6662	0.5248	0.5058
STC	0.3042	0.6792	0.7693	0.6689	0.6064
STC-IndChild	0.2914	0.5906	0.8025	0.5354	0.5428

With both strategies, a slight decrease in contamination and a significant increase in precision can be observed. However these improvements are handicapped by a decrease in recall, F-score and NMI, which means that these strategies serve only to choose precision over recall, without providing any advantage to the algorithm. A superset may actually represent a hypernym or meronym containing consistent subgroups. Our strategies do not handle such cases,

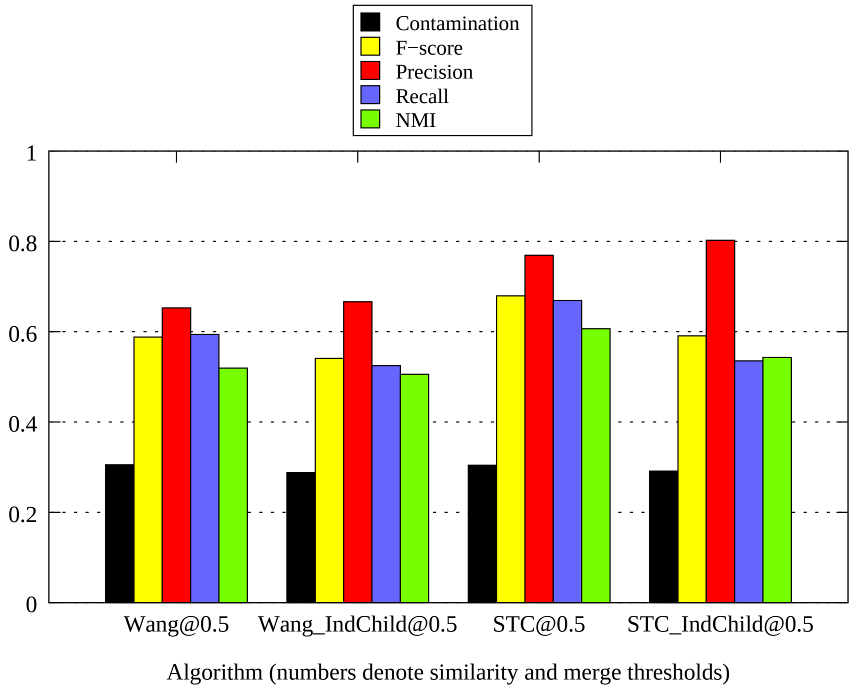


Figure 5.7: Independent child node check on Wang et al. method and STC (AMBIENT, top 10 clusters, ignoreWordPercent = 0.9)

which explains the results in Figure 5.6 and Figure 5.7.

5.2.3 Using Normalized Wikipedia Distance (NWD) to measure relevance

Phrases may not always carry a high informational value despite of their high frequency in the snippets. Snippets from the same source (e.g. ”Wikipedia”, ”UEFA.com” and such website names or identifiers of the website, such as ”Encyclopedia”) can yield examples of such cases.

Set relations based on the overlapping formula proposed by Wang et al. can be coincidental for these examples and merge operations based on these overlapping sets become noisy. A semantic relatedness measure such as NWD can be used to consider additional knowledge about cluster labels in the form of short phrases.

To apply this idea, we first computed NWD for 1000 related pairs from Wikipedia to identify a similarity threshold. The mean and standard deviation for these related pairs were 0.42160 and 0.29547 respectively. We selected a maximum NWD of 0.71707, one standard above the mean, as similarity threshold.

Then we allowed merging of base clusters only when NWD of cluster labels was below this similarity threshold. We also allowed merging operations for undefined NWD results. Table 5.7 shows the results and Figure 5.8 presents a comparison with Wang et al. As observed from the results, this method decreased the noise in merged clusters, yielding a decreased contamination, increased precision and NMI. However, this hurts our ability to represent all relevant documents in top 10 clusters since cluster sizes decrease, resulting in a sharp decrease in recall. Still, the decrease in F-score is small and this method can be used to focus on higher quality clusters, with a trade-off in document representation.

Table 5.7: Performance of Wang et al. with NWD merge controls

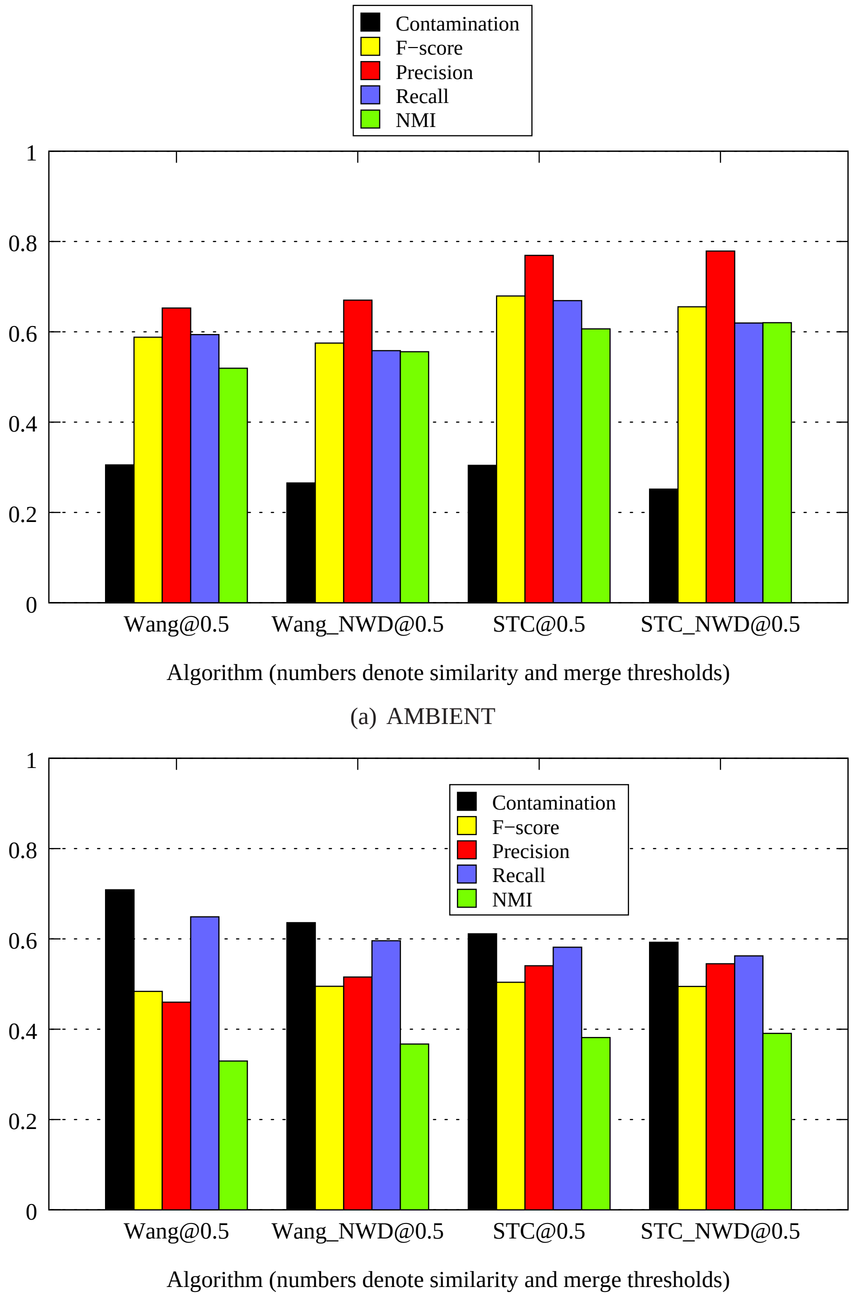
Algorithm	Dataset	Contamination	F-score	Precision	Recall	NMI
Wang-NWD	AMBIENT	0.2652	0.5749	0.6702	0.5582	0.5558
Wang-NWD	ODP-239	0.6359	0.4949	0.5155	0.596	0.3672
STC-NWD	AMBIENT	0.2516	0.6553	0.7788	0.6195	0.6202
STC-NWD	ODP-239	0.5925	0.4946	0.5451	0.5625	0.3907

5.3 Boosting Document Scores According to Coverage

Inspired by the work of Crabtree et al. [13] on cluster coverage, we tested an idea about scoring documents in base clusters according to their coverage, as explained in Section 4.2.3. Figure 5.9 shows that this strategy can increase coverage without hurting other performance measures. Table 5.8 shows the results in detail.

Table 5.8: Document coverage boost performance

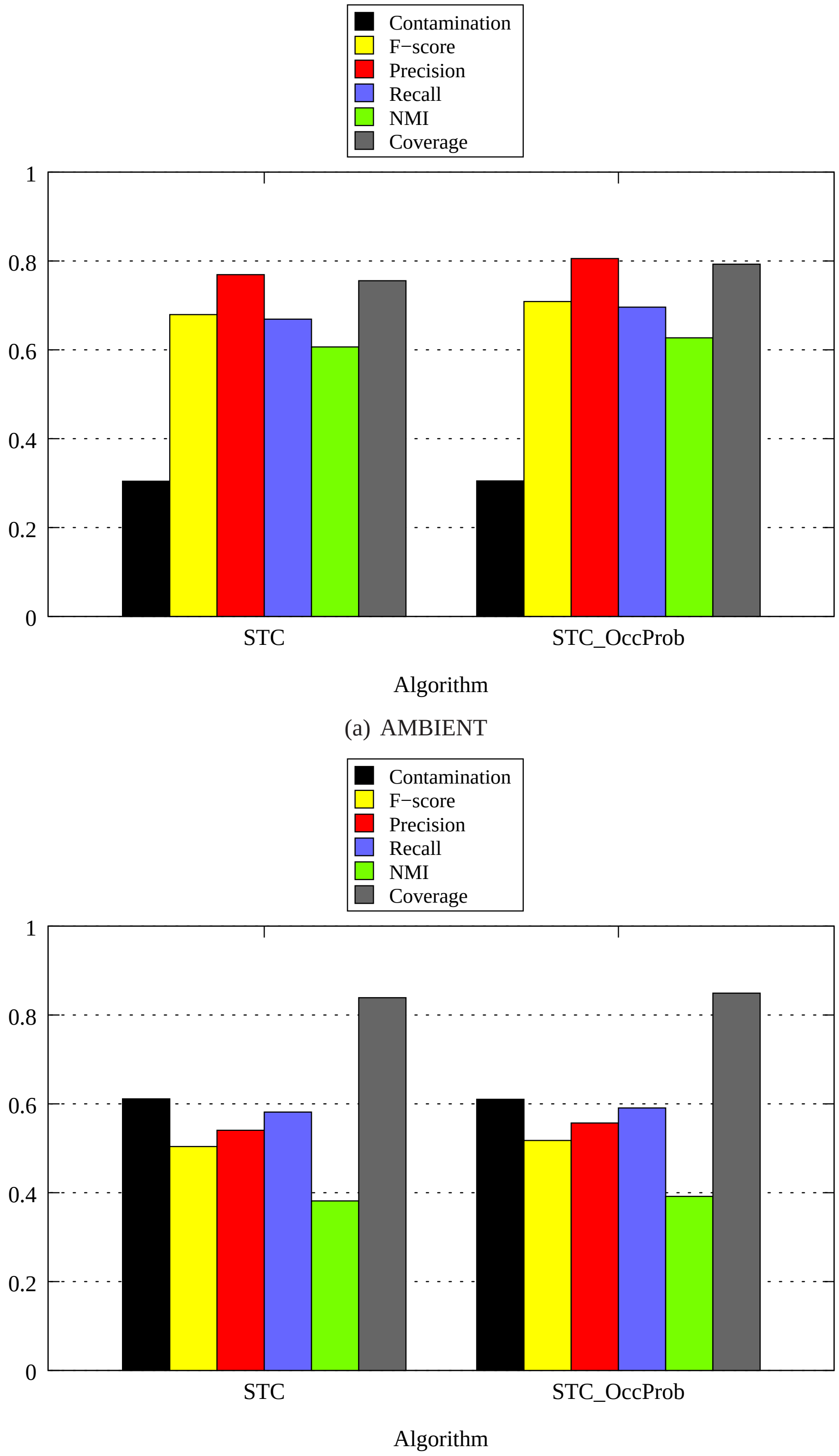
Dataset	Contamination	F-score	Precision	Recall	NMI	Coverage
AMBIENT	0.305	0.7087	0.8055	0.6959	0.627	0.7928
ODP-239	0.6102	0.5176	0.557	0.5908	0.3917	0.8493



(a) AMBIENT

(b) ODP-239

Figure 5.8: NWD merge check on Wang et al. method and STC (top 10 clusters, ignoreWordPercent = 0.9)



(a) AMBIENT

(b) ODP-239

Figure 5.9: Comparison of STC and document coverage boosted STC

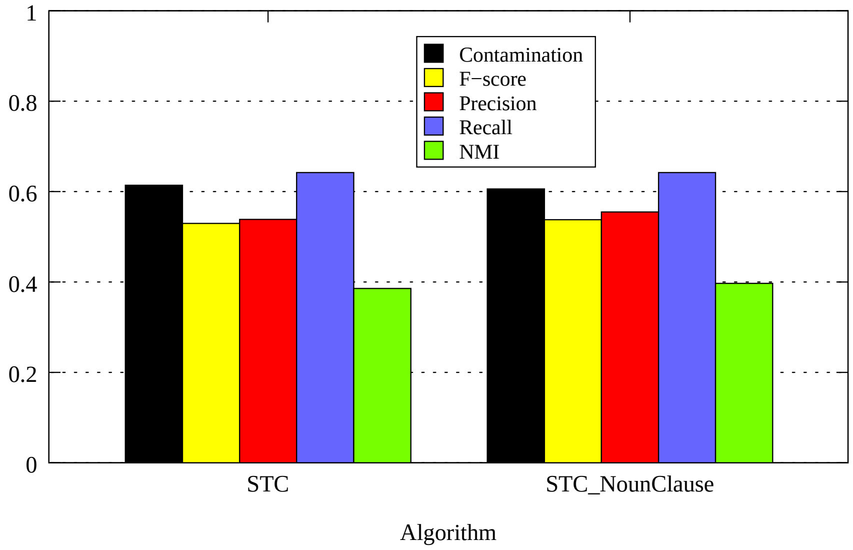


Figure 5.10: STC and noun clause filter in Turkish (ODP-TR30, top 10 clusters, ignoreWordPercent = 0.9)

5.4 Part-of-Speech: Filtering with Noun Clauses in Turkish

Carpineto et al. [3] used a hashed lexicon list to filter non-descriptive words using a simple POS filter. In their work, they only allowed labels containing adjectives and nouns, including proper names. To see the effect of such POS filtering in Turkish data, we tagged the last words of labels with Zemberek and only allowed labels in noun clause form. Figure 5.10 shows the effect of this strategy on ODP-TR30 dataset containing Turkish snippets from ODP. Table 5.9 shows the results in detail. These results show that noun clause filter slightly improves STC in Turkish as we expect, by eliminating non-descriptive labels and increasing precision.

Table 5.9: STC and noun clause filter performance on ODP-TR30 dataset

Method	Contamination	F-score	Precision	Recall	NMI
STC	0.6139	0.5296	0.5386	0.6421	0.3855
STC (Noun clause)	0.6058	0.5378	0.5548	0.6419	0.3969

5.5 Evaluation of semantic relatedness measures

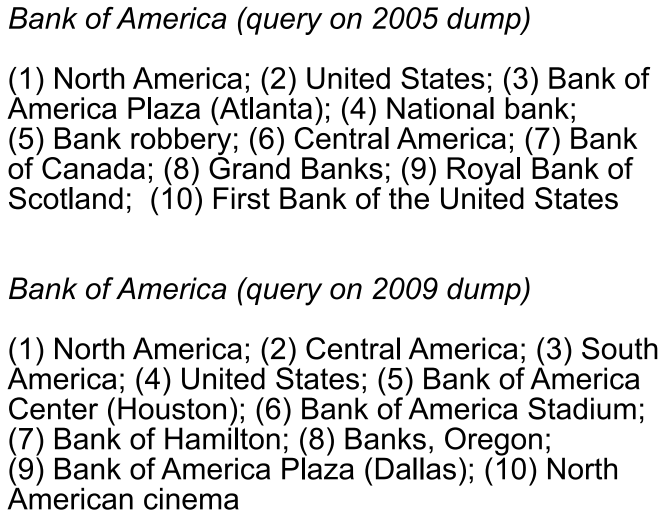
To make use of semantic relatedness methods as binary classifiers, we evaluated NWD, ESA and WLM to identify corresponding thresholds. If such a threshold exists for a given method, we can use that method to check whether two text fragments are semantically relevant or not.

To identify these thresholds, we compute semantic relatedness between 1000 random related pairs from Wikipedia. These pairs and the scripts we used to extract these pairs are available at [81]. Excluding unknown pairs, we compute the mean and standard deviation of results. If evaluated method returns semantic distance, then we pick one standard deviation above the mean as threshold. For methods returning semantic relatedness, we pick one standard deviation below the mean.

As discussed in Section 5.2.3, we identified a maximum distance of 0.71707 as threshold for NWD. We used English Wikipedia dump from 18.6.2009 as knowledge base for NWD.

Our ESA implementation yields a Spearman correlation of 0.737 for November 2005 dump from Gabrilovich et al. (compared to 0.74 from the original study) and yields a correlation of 0.675 for the dump from 18.6.2009. However, we computed a mean of 0.26 and standard deviation of 0.31 for 2005 dump and a mean of 0.24 and standard deviation of 0.31 for 2009 dump with ESA. One standard deviation below the mean is -0.05 and -0.07 respectively and such values cannot be used as threshold. Moreover, we observed that top 10 concepts from regular ESA feature vectors were often noisy despite the results from [25]. Figure 5.11 shows such a noisy example. We decided not to use ESA after these observations.

WLM method treats Wikipedia data more directly than ESA and provides a sound relatedness metric by computing relatedness with inlinks and outlinks. For WLM, we used preprocessed English Wikipedia dump from 6.3.2009 provided at the WikipediaMiner project page [82] and computed a mean of 0.70441 and a standard deviation of 0.20029, yielding a threshold of 0.50411. For Turkish Wikipedia dump from 6.4.2010, we computed a mean of 0.74038 and a standard deviation of 0.18865, yielding a threshold of 0.55172.



Bank of America (query on 2005 dump)

(1) North America; (2) United States; (3) Bank of America Plaza (Atlanta); (4) National bank; (5) Bank robbery; (6) Central America; (7) Bank of Canada; (8) Grand Banks; (9) Royal Bank of Scotland; (10) First Bank of the United States

Bank of America (query on 2009 dump)

(1) North America; (2) Central America; (3) South America; (4) United States; (5) Bank of America Center (Houston); (6) Bank of America Stadium; (7) Bank of Hamilton; (8) Banks, Oregon; (9) Bank of America Plaza (Dallas); (10) North American cinema

Figure 5.11: Example ESA query ”bank of america” on 2005 and 2009 dumps

5.6 Combining semantic relatedness with STC

We tested two strategies to control noisy clusters in STC: Checking whether cluster labels are consistent with cluster content and checking whether clusters to be merged have semantically similar labels. Following subsections summarize results of applying these strategies.

5.6.1 Filtering base clusters with WLM

Consistency between cluster label and content can be checked by taking a number of sample snippets from the cluster and computing the relatedness between the label and these snippets. For cluster C, the number of samples is computed as in Equation (5.4):

$$\max(\lfloor |C| * sampleRatio \rfloor, 2) \tag{5.4}$$

where *sampleRatio* is the ratio of samples to all documents in the cluster. We use a *sampleRatio* of 0.3 in this study.

WLM did not originally include a way to compare two text fragments but fortunately, the result of another study from Milne et al. [49] provided a way to wikify arbitrary text fragments. Wikification is the process of linking relevant phrases in a given text to corresponding Wikipedia concepts. Using this wikification method, we added a method to compare arbitrary text fragments to WikipediaMiner Toolkit. In this method, given text fragments are first wikified and WLM comparison is applied with the all inlinks and all outlinks of resulting concepts.

In this case, we find all senses for the cluster label, compute the relatedness between every sense and the given concept group (obtained by wikifying the snippet) and take the maximum as the final relatedness between the label and snippet.

Since we compute relatedness between the label and multiple snippets, there are multiple options to assess relatedness. We chose to compute the average relatedness and allow a cluster if this value is above 0.50411. We also allow a cluster if it has a label not defined in WLM.

Table 5.10 presents the results of this strategy for English and Table 5.11 presents the results for Turkish. Filtering directly reduced noisy clusters and resulted in a significant increase in precision and decrease in contamination. However, these improvements are coupled with a significant decrease in recall. This decrease is probably the result of losing coherent clusters with meaningless labels. Such clusters may eventually end up with a good cluster label in original STC, since they get scored, merged to other clusters and most of the time, eliminates uninformative labels with selection rules defined in [87]. Removing such clusters causes a direct information loss that we cannot recover from.

Table 5.10: STC and WLM-filtered STC performance on AMBIENT dataset

Method	Contamination	F-score	Precision	Recall	NMI
STC	0.3042	0.6792	0.7693	0.6689	0.6064
WLM-Filter	0.2271	0.6395	0.8074	0.5905	0.5973

Table 5.11: STC and WLM-filtered STC performance on ODP TR-30 dataset

Method	Contamination	F-score	Precision	Recall	NMI
STC	0.6139	0.5296	0.5386	0.6421	0.3855
WLM-Filter	0.562	0.5238	0.573	0.595	0.3923

5.6.2 Merging phase modified with WLM

Without any semantic consideration, STC allows merging of base clusters as long as they sufficiently overlap. For an example ”amazon” query on English Wikipedia, phrase pairs such as ”Category - Following”, ”Parrot - Known”, ”Mainly Green - Long” turn out to be such merged pairs. To prevent such merge operations, we check the semantic relatedness between cluster labels with WLM and allowed merging if the score is above 0.50411. We also allow merging if either label is not defined in WLM.

The results for English can be observed from Table 5.13. There is an improvement in contamination, precision and NMI. We expected such an improvement since we directly controlled merging of noisy pairs. However our strategy hurt recall. Primary reason for this is the free-floating noisy clusters, that were previously contained in a larger cluster. These noisy clusters may stay small with our checks, but their amount increases. Another reason might be our choice of relatedness threshold. For ”amazon” query, the pairs in Table 5.12 are not related according to our choice, when they are actually related. The results for Turkish, presented in Table 5.14, do not reflect the same effect, aside from a contamination improvement.

Table 5.12: Semantic relatedness between cluster labels from ”amazon” query

Label1	Label2	Score
Feminism	Amazon Feminism	0.4568
Feminism	Female	0.4180
TV Series	Documentary	0.4147
Type 21 Frigate	Royal Navy	0.4757
Type 21 Frigate	Navy	0.4757

Table 5.13: STC and WLM merge control performance on AMBIENT dataset

Method	Contamination	F-score	Precision	Recall	NMI
STC	0.3042	0.6792	0.7693	0.6689	0.6064
WLM-Merge	0.2611	0.6575	0.7711	0.6263	0.625

Table 5.14: STC and WLM merge control performance on ODP TR-30 dataset

Method	Contamination	F-score	Precision	Recall	NMI
STC	0.6139	0.5296	0.5386	0.6421	0.3855
WLM-Merge	0.6094	0.527	0.5364	0.6432	0.3852

5.6.3 Filtering and merging with WLM

Apply filtering from Section 5.6.1 and merge controls from Section 5.6.2 simultaneously can help with free-floating clusters with no meaning. Filtering can handle most of these clusters before merge step. Table 5.15 presents the results of this combination for English and Table 5.16 presents the results for Turkish. Internal quality of clusters are further increased (best contamination and precision for STC is achieved) but the recall problem still persists. Our method produces smaller clusters with higher quality, instead of large clusters. We can represent fewer relevant documents in top 10 clusters because of this problem with cluster sizes.

Table 5.15: STC and WLM filter + merge control performance on AMBIENT dataset

Method	Contamination	F-score	Precision	Recall	NMI
STC	0.3042	0.6792	0.7693	0.6689	0.6064
WLM-FilterMerge	0.2065	0.6346	0.8061	0.5818	0.6186

Table 5.16: STC and WLM filter + merge control performance on ODP TR-30 dataset

Method	Contamination	F-score	Precision	Recall	NMI
STC	0.6139	0.5296	0.5386	0.6421	0.3855
WLM-FilterMerge	0.5632	0.5105	0.5793	0.5675	0.3865

5.7 Term-frequency based clustering baseline

In subsequent sections we focus on using semantic relatedness information from WLM in a more direct way. In this section, we present a baseline for these efforts with term-frequency based group average hierarchical clustering (GAHC). We clustered term-document matrices using hierarchical clustering algorithm from Weka [30] by feeding distances between normalized vectors directly to the algorithm. We used $1000 * (1 - \frac{V_i \cdot V_j}{|V_i| * |V_j|})$ to set the distance, where $\frac{V_i \cdot V_j}{|V_i| * |V_j|}$ denotes the cosine similarity of two term-frequency vectors. Table 5.17 shows the results of this baseline.

Table 5.17: Term-frequency based clustering performance

Dataset	Contamination	F-score	Precision	Recall	NMI
AMBIENT	0.1883	0.5331	0.8831	0.4418	0.5148
ODP-239	0.3643	0.4085	0.7296	0.3271	0.3797
ODP-TR30	0.2871	0.4664	0.782	0.3805	0.4298

5.8 Wikification clustering

Another way of using semantic information in a direct way is wikifying every snippet with the method of Milne et al. [49] and then using every Wikipedia concept as a cluster. Table 5.18 presents the results of this approach. We observe that clustering by wikification has a performance similar with STC, with a significant advantage in NMI. This stems from the fact that Wikipedia concepts encapsulate the topics among the snippets with a better distinction. Knowing a snippet belongs to a Wikipedia concept gives a higher amount of information than STC clusters, which can sometimes carry no meaning other than co-occurrence.

Table 5.18: Wikification clustering performance

Method	Dataset	Contamination	F-score	Precision	Recall	NMI
STC	AMBIENT	0.3042	0.6792	0.7693	0.6689	0.6064
Wikify	AMBIENT	0.3286	0.6513	0.7358	0.6921	0.6633
STC	ODP-239	0.6112	0.504	0.5404	0.5815	0.3814
Wikify	ODP-239	0.5808	0.498	0.5621	0.5523	0.4031
STC	ODP-TR30	0.6139	0.5296	0.5386	0.6421	0.3855
Wikify	ODP-TR30	0.5816	0.509	0.5978	0.5481	0.3992

5.9 Hierarchical clustering with WLM

To test our hypothesis that semantic features can improve search result clustering, we made use of WikipediaMiner to generate Wikipedia concepts as additional features for every snippet and clustered these snippets using group-average hierarchical clustering (GAHC) algorithm from Weka.

5.9.1 GAHC using Wikipedia concepts from wikification

To investigate the effect of Wikipedia concept vectors, we generated all possible Wikipedia links for every snippet. WikipediaMiner also returns a probability weight for each concept, to indicate the tendency of a Wikipedian to link to that concept. Then we compute the distance matrix by using $1000 * (1 - \frac{C_i \cdot C_j}{|C_i| * |C_j|})$ to set the distance, where $\frac{C_i \cdot C_j}{|C_i| * |C_j|}$ denotes the cosine similarity of two concept vectors. By directly feeding this distance matrix to GAHC algorithm from Weka, we generated the results shown in Table 5.19.

From the results, we observed that this approach improved recall. We expected such a result since wikification enriches our representation of snippets with additional features. However, it can also be seen that the noise introduced by wikification hurts precision. The reason behind this is probably the noise associated with wikification process. Wikification cannot do much to resolve ambiguous anchors when input text is limited, as in our case. We also observe a high contamination introduced by weakly related concepts. These concepts having only a weak connection with the snippet or appear because of an error in word-sense disambiguation often result in many unrelated documents in the same cluster. Despite the noise, Wikipedia concepts are more general than terms and encapsulate more documents, causing an increase in recall.

Table 5.19: Performance of GAHC using Wikipedia concepts vs. term-frequency baseline

Method	Dataset	Contamination	F-score	Precision	Recall	NMI
TermFreq	AMBIENT	0.1883	0.5331	0.8831	0.4418	0.5148
GAHC-Wiki	AMBIENT	0.3717	0.5174	0.7306	0.4847	0.432

5.9.2 GAHC with term frequencies and Wikipedia concepts from wikification

Inspired by Banerjee et al. [2], we thought that a hybrid approach can perform better by combining best properties of term-frequency vectors and concept vectors and combined these vectors.

Table 5.20: Performance of GAHC using term frequencies + Wikipedia concepts

Method	Dataset	Contamination	F-score	Precision	Recall	NMI
TermFreq	AMBIENT	0.1883	0.5331	0.8831	0.4418	0.5148
GAHC-Wiki	AMBIENT	0.3717	0.5174	0.7306	0.4847	0.432
GAHC-Hybrid	AMBIENT	0.1289	0.6047	0.9108	0.5056	0.5574
TermFreq	ODP-239	0.3643	0.4085	0.7296	0.3271	0.3797
GAHC-Hybrid	ODP-239	0.3486	0.4407	0.7348	0.3607	0.3991

Since the weights for the concept vector are in the [0,1] range, we scaled these values with the maximum frequency from the term vector. Then, as in Section 5.9.1, we computed the distance matrix by using $1000 * (1 - \frac{C_i \cdot C_j}{|C_i| * |C_j|})$ to set the distance. By running GAHC algorithm from Weka with this matrix, the results presented in Table 5.19 are generated.

Table 5.19 shows that his hybrid method outperforms both the baseline and concept-based clustering. Generality of Wikipedia concepts and noise reduction advantage from term-frequency features complement each other to provide such a result. This clearly indicates that semantic features improve search result clustering. Table 5.21 shows that the result is also valid for Turkish.

Table 5.21: Performance on Turkish data with GAHC using term frequencies + Wikipedia concepts

Method	Dataset	Contamination	F-score	Precision	Recall	NMI
TermFreq	ODP-TR30	0.2871	0.4664	0.782	0.3805	0.4298
GAHC-Hybrid	ODP-TR30	0.2519	0.4711	0.7798	0.3873	0.4389

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis aimed to improve Suffix Tree Clustering (STC), a search result clustering (SRC) method, evaluate a previous study from Wang et al and investigate the effectiveness of explicit semantic features on STC algorithm and SRC task. With the intention of initiating SRC study in Turkish, we have created a dataset for Turkish data and performed the same methods in this dataset.

Running STC on two standard datasets for English, AMBIENT and ODP-239, to produce 10 clusters resulted in F1-scores of %67.92 and %50.4 respectively. We have also recorded two other metrics indicating cluster quality, contamination and NMI. We observed a contamination of %30.42 and %61.12 and an NMI of %60.64 and %38.14 respectively, for these two datasets.

We performed experiments to evaluate the improvements of Wang et al. on STC. We could only acquire F1-scores of %58.8 and %48.39 on AMBIENT and ODP-239, indicating no improvement. Contamination (%30.52 and %70.86) and NMI measurements (%51.92 and %32.95) supported that new similarity formula of Wang et al. provided no improvement over the similarity formula from Zamir et al. We were expecting such a result because overlap score from Wang et al. had a flaw: clusters with single-word labels have an unfair advantage. Such clusters tend to also have many small subsets, increasing noise when merged. Textual similarity prevents some of the noise but still, this flaw affects the effectiveness of the new similarity formula.

We believed that the similarity formula from Wang et al. could work if merging problems were fixed. We tried to control incorrect merge operations by three different approaches. Using average intra-cluster distance reduced noise, increasing cluster quality (contamination decreased from %30.52 to %22.01, NMI also increased from %51.92 to %58.17 on AMBIENT) but top 10 clusters did not include as many documents because of canceled merge operations and recall decreased significantly from %59.39 to %50.43. We were expecting that noise reduction would balance this trade-off and yield an improvement.

In order to prevent large clusters from subsuming informative groups, we checked whether clusters had multiple connected components. We expected that this would eliminate wrapper clusters with no relevant meaning. Despite the increase in precision (%65.28 to %67.29) and decrease in contamination (%30.52 to %27.68), F1-score decreased from %58.8 to %53.57 because of recall, similar to our previous method.

We also checked whether a cluster had any disconnected subcluster in it, to identify incorrectly merged clusters inside wrappers. Similar to our previous efforts, F1-score decreased from %58.8 to %54.07, despite improvements in precision (%65.28 to %66.62) and contamination (%30.52 to %28.8).

The last strategy we tested to fix problems with the similarity formula of Wang et al. was preventing incorrect merge operations by checking semantic compatibility of cluster labels. We used NWD since it is suitable for computing relatedness between words and short phrases. Despite improvements on contamination, precision and NMI, F1-score did not improve because of the decrease in recall. We were expecting a higher increase in precision, resulting in a higher F1-score. Smaller clusters may have decreased our ability to represent relevant documents in top 10 clusters. However, the decrease in F1-score is small (%58.8 to %57.49) and this method can still be used to focus on clusters with higher quality.

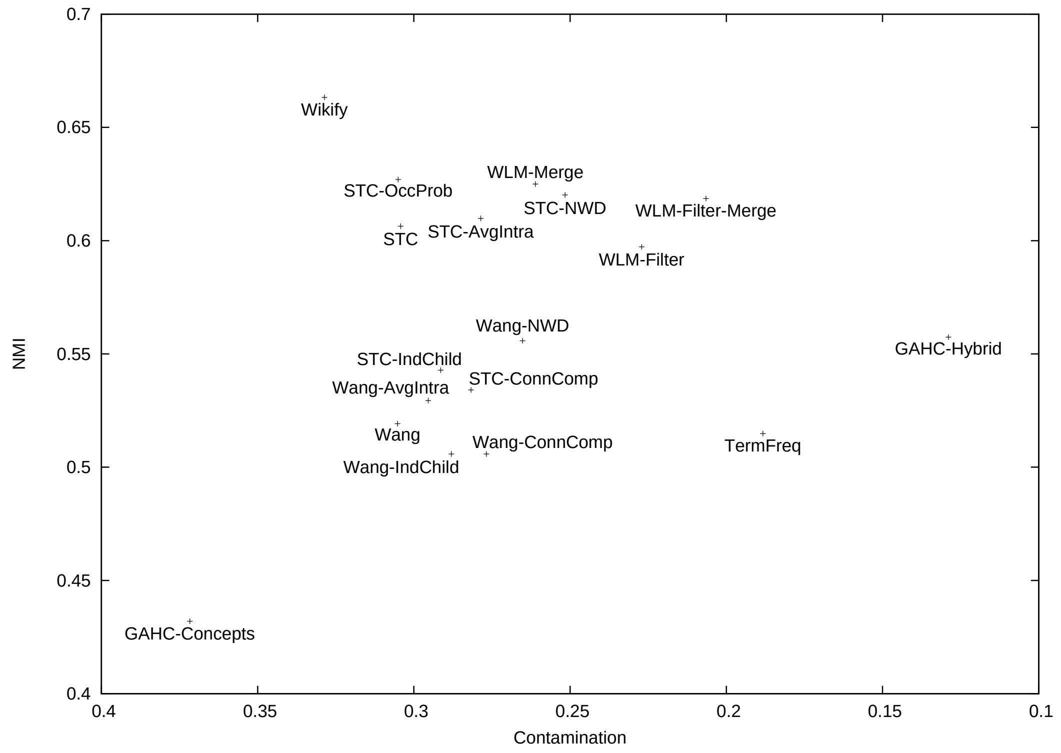
In the light of Crabtree et al.[13], we tested the idea of scoring documents by the number of clusters they appear in. With this strategy, we emphasized the documents that are hard to represent by boosting them. We acquired a higher F1-score of %70.87 compared %67.92 from original STC, coupled with improvements on precision, recall and NMI. We were expecting such an increase since our boost would increase recall, but it is interesting that this did not hurt precision. The increase in NMI can be explained by better separation of documents across top 10 clusters, in order to represent rare ones. We also measured the ratio of snippets represented in top 10 clusters to all snippets (coverage) and found that it increased from %75.54 to %79.28 as expected.

As Carpineto et al. [3] similarly did for English data, we wanted to test whether allowing only well-formed cluster labels increased performance in Turkish. We prepared ODP-TR30 dataset for Turkish following the same principles used when creating ODP 239. Turkish data in ODP is limited but we could still extract such a dataset beneficial for SRC research in Turkish. Our experiments on ODP-TR30 showed that choosing labels in noun-clause form resulted in an increase in F1-score from %52.96 to %53.78, coupled with small improvements in contamination (%61.39 to %60.58) and NMI (%38.55 to %39.69).

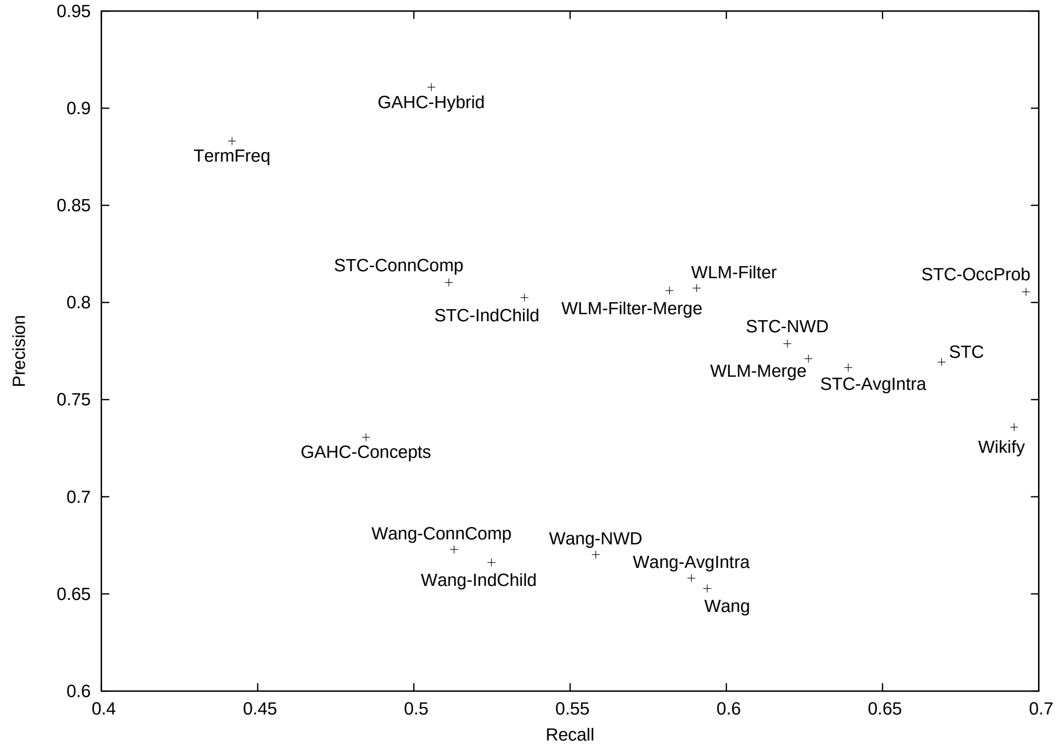
Semantic filtering and merging did not provide an improvement on STC in F-score. On AMBIENT dataset, F1-score decreased to %63.95 with filtering, %65.75 with merging and to %63.46 when both filtering and merging were applied. However this may be caused by irrelevant Wikipedia concepts generated for a snippet, used when comparing a label to a snippet. We added an artificial method to compute relatedness between a term and a text fragment to WLM. This method can also be ineffective and incorrect since it uses all links from generated concepts. Our method can be modified to use only the most common subset of inlinks and outlinks. Moreover we also observed that our choice of threshold can prevent a number of legitimate merge operations.

Hierarchical clustering experiments showed that using concepts from Wikipedia as features improved the performance when used together with term frequencies from bag-of-words approach. Applying group-average hierarchical clustering with hybrid vectors resulted in an F1-score increase: from the baseline F1-score of %53.31 to %60.47 on AMBIENT dataset and from %40.85 to %44.07 on ODP 239.

The main proposal of this thesis was that the use of explicit semantic features would increase clustering performance. The result from hierarchical clustering experiments support this claim. However our proposal that the performance of STC could be increased by filtering base clusters and controlling merge operations with semantic relatedness failed with our current semantic tools. Still, we think that there is room for improvement in STC with such controls.

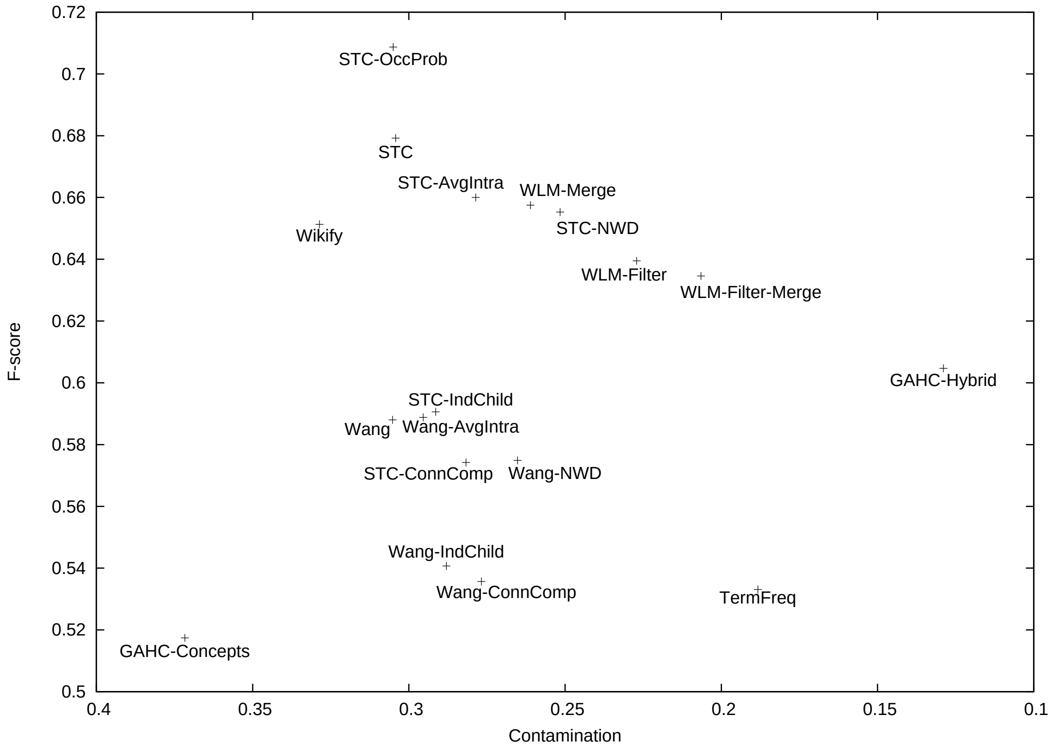


(a) NMI vs. Contamination

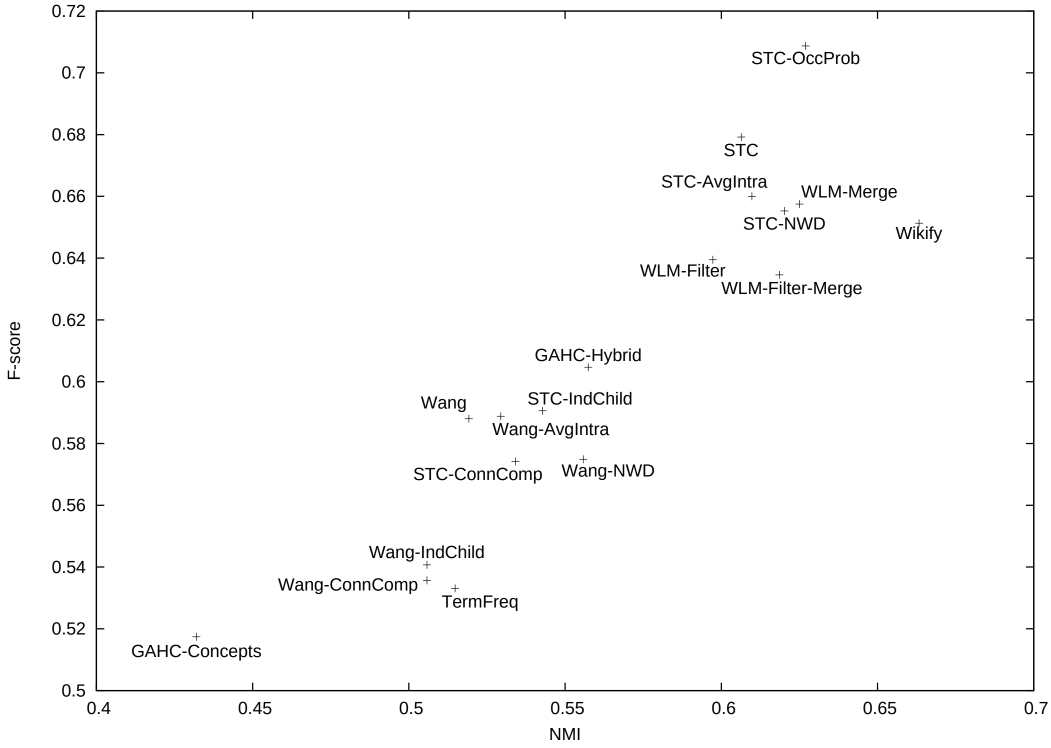


(b) Precision vs. Recall

Figure 6.1: All results on AMBIENT dataset (best performance at top-right corner)

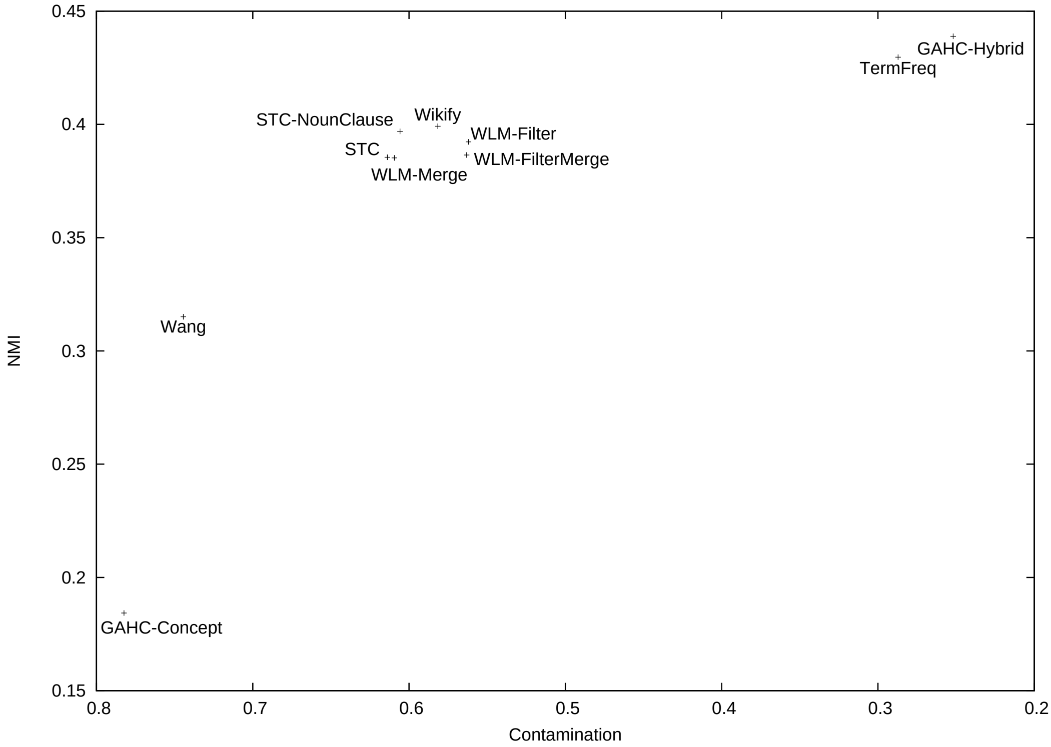


(a) F-score vs. Contamination

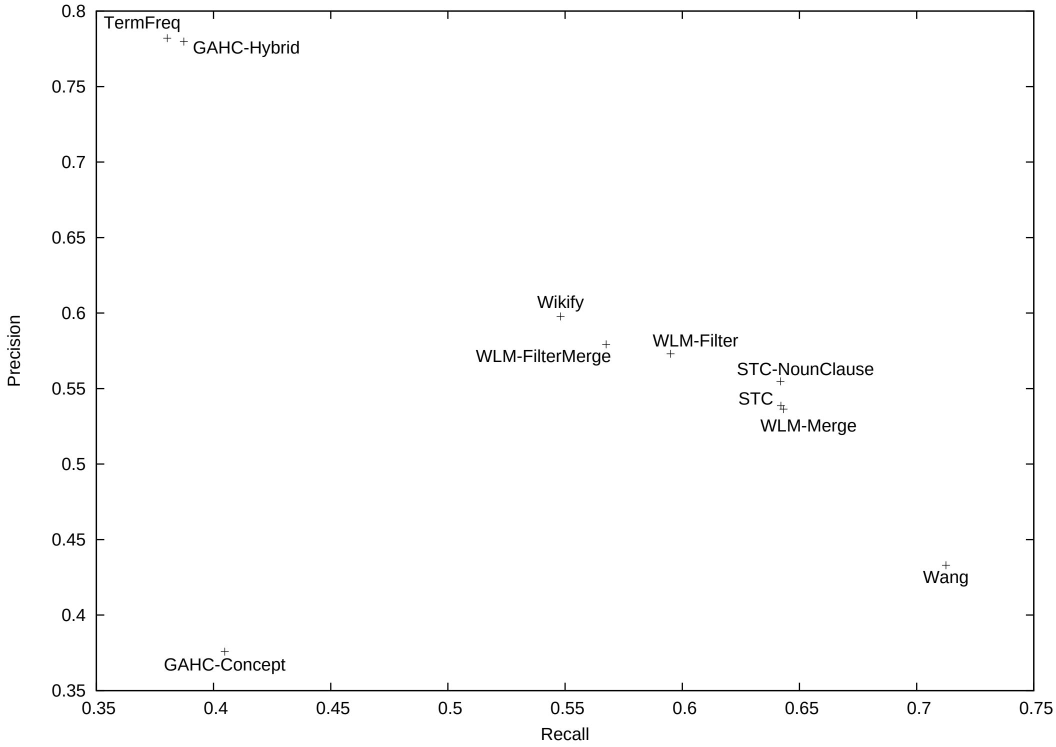


(b) F-score vs. NMI

Figure 6.2: All results on AMBIENT dataset (continued)

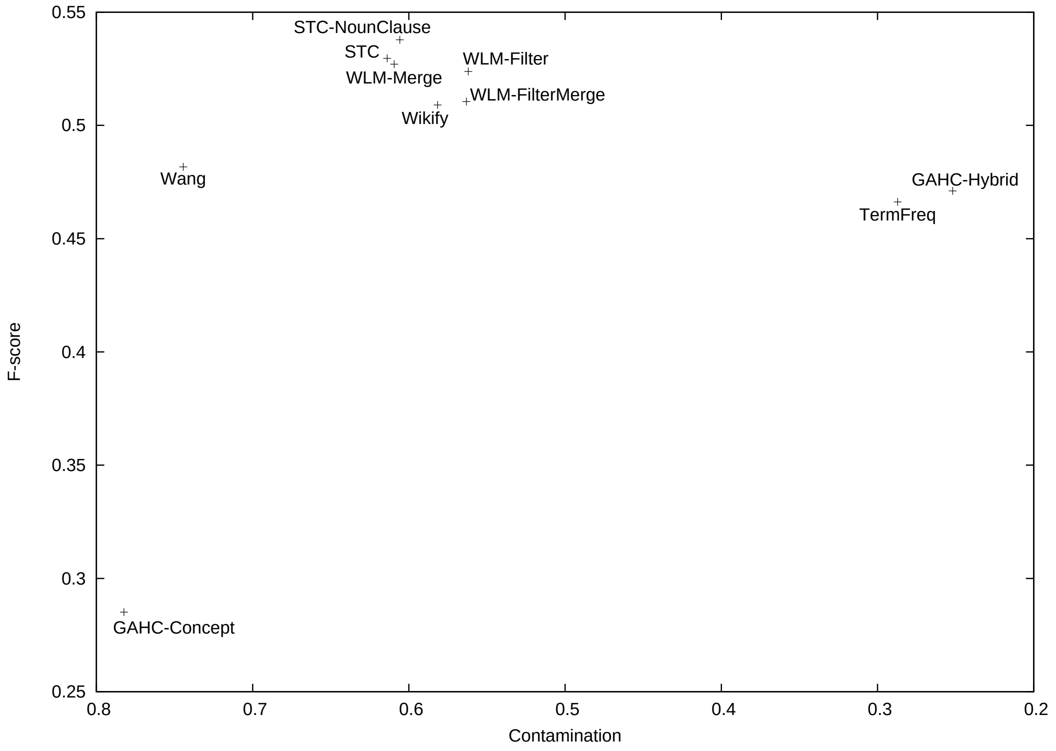


(a) NMI vs. Contamination

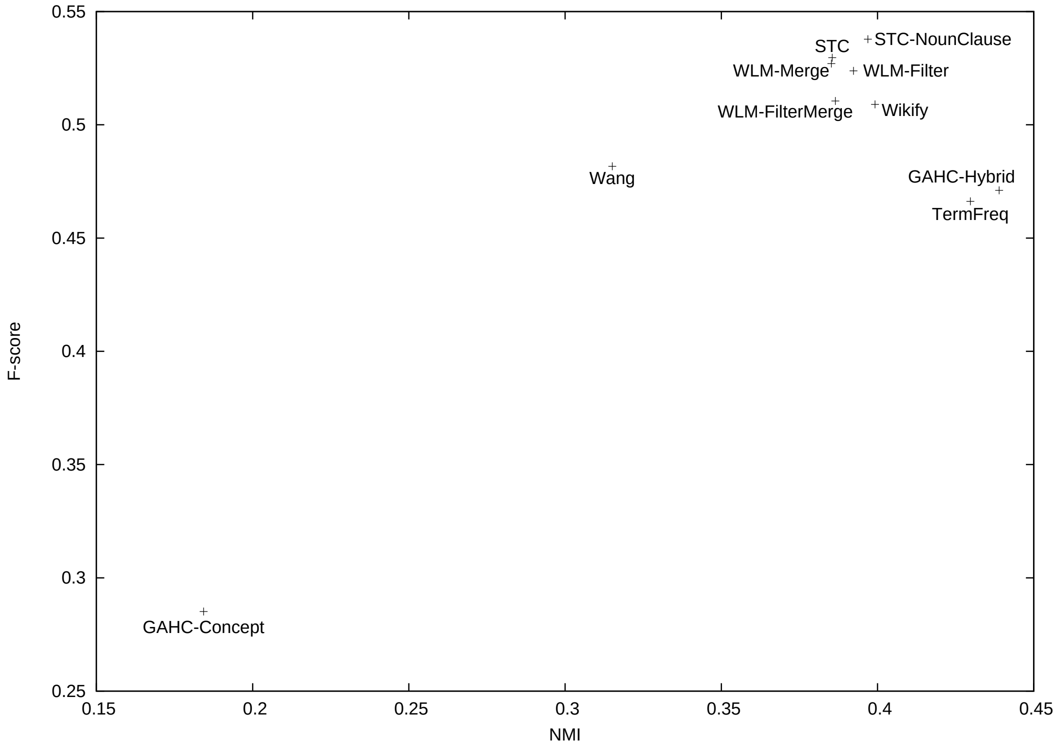


(b) Precision vs. Recall

Figure 6.3: All results on ODP-TR30 dataset (best performance at top-right corner)



(a) F-score vs. Contamination



(b) F-score vs. NMI

Figure 6.4: All results on ODP-TR30 dataset (continued)

In Figure 6.1 and 6.2, results from all experiments on AMBIENT dataset are aggregated to provide a comparison among these methods for English. In the same manner, Figure 6.3 and 6.4 provide a comparison for Turkish by displaying the results from all experiments on ODP TR-30 dataset.

From these figures, it can be observed that semantic filter and merge controls on STC can bring improvements in different aspects. Figure 6.2(a) and Figure 6.4(a) show that all semantic approaches, except clustering by wikification, reduced contamination. Figure 6.1(a) and Figure 6.3(a) stress that semantics can help to generate clusters that are cleaner (lower contamination) and separated better (higher NMI). From user perspective, this results in more intuitive clusters. Figure 6.1(b) also show that there is a large recall gap between node connectivity controls on STC (STC-IndChild, STC-ConnComp) and original STC, despite reduced contamination and better precision. These controls can be used in a task where precision is preferred.

Obviously, STC-OccProb (STC with document scoring according to representation) is better than STC in every way except contamination. STC-OccProb can be used in tasks where users care less about the noise in clusters and focus on topic discovery.

6.2 Future Work

Filtering and merge control with WLM failed to improve STC but in our implementation, we used a quick modification of WLM to accommodate comparison of text fragments. WLM can be improved further to increase the accuracy of concept group comparisons or another effective method can be employed. These results depend on current semantic relatedness tools and new approaches have the potential to change the conclusions from these experiments.

ODP-TR30, Turkish SRC dataset introduced in this thesis, can be used to investigate other SRC methods in Turkish. We believe that better, more complex NLP methods can be employed for Turkish. We also provide the scripts that we used to extract this dataset and it can be used to extract such datasets for other languages in ODP. SRC experiments can be repeated for other languages using such datasets.

Hierarchical clustering with the combination of term vectors and semantic feature vectors proved to be effective but in our current implementation, querying WikipediaMiner for semantic information takes a long time, especially for wikification. Other ways to make use of such features should be investigated to achieve competitive running times with other algorithms. We also did not consider cluster labeling problem with such methods.

Converting our binary semantic relatedness definitions, similarity formula from Zamir et al. and similarity formula from Wang et al. to fuzzy alternatives can improve all related methods.

REFERENCES

[1] B. Alberti, F. Anklesaria, P. Lindner, M. McCahill, and D. Torrey. The Internet Gopher protocol: A distributed document search and retrieval protocol. *University of Minesota Microcomputer and Workstation Networks Center, Spring*, 1991.

[2] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using Wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, page 788. ACM, 2007.

[3] A. Bernardini, C. Carpineto, and M. D'Amico. Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering. *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09*, 1, 2009.

[4] A. Broder. A taxonomy of web search. In *ACM Sigir Forum*, volume 36, page 10. ACM, 2002.

[5] C. Carpineto, S. Mizzaro, G. Romano, and M. Snidero. Mobile information retrieval with search results clustering: Prototypes and evaluations. *Journal of the American Society for Information Science and Technology*, 60(5):877–895, 2009.

[6] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of Web clustering engines. *ACM Computing Surveys (CSUR)*, 41(3):17, 2009.

[7] C. Carpineto and G. Romano. Exploiting the potential of concept lattices for information retrieval with CREDO. *Journal of universal computer science*, 10(8):985–1013, 2004.

[8] Romano G. Carpineto C. Ambient dataset. http://credo.fub.it/ambient/, 2008. Last visited on: 9 September 2010.

[9] Romano G. Carpineto C. ODP239 dataset. http://credo.fub.it/odp239/, 2009. Last visited on: 9 September 2010.

[10] Ç. Çallı. ODP TR-30 dataset. http://github.com/faraday/odp-tr30/, 2010. Last visited on: 8 October 2010.

[11] H. Chim and X. Deng. A new suffix tree similarity measure for document clustering. In *Proceedings of the 16th international conference on World Wide Web*, page 130. ACM, 2007.

[12] R. Cilibrasi and P. Vitanyi. Automatic meaning discovery using Google. *Manuscript, CWI*, 2004.

[13] D. Crabtree, X. Gao, and P. Andreae. Improving web clustering by cluster selection. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 172–178, 2005.

[14] D.R. Cutting, D.R. Karger, J.O. Pedersen, and J.W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329. ACM New York, NY, USA, 1992.

[15] Peter Deutsch. An internet archive server server (was about lisp). http://groups.google.com/group/comp.archives/msg/a77343f9175b24c3?output=gplain&pli=1, Sep 1990. Last visited on: 19 May 2010.

[16] F.C. Ekmekcioglu, M.F. Lynch, and P. Willett. Stemming and n-gram matching for term conflation in Turkish texts. *Information Research*, 2(2):2–2, 1996.

[17] Alan Emtage and Peter Deutsch. Archie - an electronic directory service for the internet. In *Proceedings of the USENIX Winter Conference*, pages 93–110. USENIX, January 1992.

[18] English Wikipedia. MediaWiki:Disambiguationspage. http://en.wikipedia.org/wiki/MediaWiki:Disambiguationspage. Last visited on: 9 June 2010.

[19] G. Eryiğit and E. Adalı. An Affix Stripping Morphological Analyzer for Turkish. In *Proceedings of the IASTED International Conference Artificial Intelligence And Applications*, pages 299–304. Citeseer, 2004.

[20] P. Ferragina and A. Gulli. The anatomy of SnakeT: A hierarchical clustering engine for web-page snippets. *Lecture notes in computer science*, pages 506–508, 2004.

[21] P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, page 810. ACM, 2005.

[22] E. Gabrilovich. *Feature Generation For Textual Information Retrieval Using World Knowledge*. PhD thesis, Technion - Israel Institute of Technology, December 2006.

[23] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1301. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[24] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 6–12, 2007.

[25] E. Gabrilovich and S. Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34(1):443–498, 2009.

[26] Evgeniy Gabrilovich and Shaul Markovitch. Feature generation for text categorization using world knowledge. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1048–1053, Edinburgh, Scotand, August 2005.

[27] F. GERACI. Fast Clustering for Web Information Retrieval.

[28] F. Geraci, M. Pellegrini, M. Maggini, and F. Sebastiani. Cluster generation and labeling for web snippets: a fast, accurate hierarchical solution. *Internet Mathematics*, 3(4):413–443, 2006.

[29] F. Geraci, M. Pellegrini, P. Pisati, and F. Sebastiani. A scalable algorithm for high-quality clustering of web snippets. In *Proceedings of the 2006 ACM symposium on Applied computing*, page 1062. ACM, 2006.

[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[31] Donna Harman, editor. *Proceedings of the Third Text Retrieval Conference TREC-3*. National Institute of Standards and Technology Special Publication 500-225, 1995.

[32] X. He, H. Zha, C. HQ Ding, and H. D. Simon. Web document clustering using hyperlink structures. *Computational Statistics and Data Analysis*, 41(1):19–45, 2002.

[33] M.A. Hearst and J.O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 76–84. ACM New York, NY, USA, 1996.

[34] W. Hersh, C. Buckley, TJ Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–201. Springer-Verlag New York, Inc. New York, NY, USA, 1994.

[35] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *Proc. of the SIGIR 2003 Semantic Web Workshop*, pages 541–544. Citeseer, 2003.

[36] J. Hu, L. Fang, Y. Cao, H.J. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging Wikipedia semantics. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 179–186. ACM, 2008.

[37] B.J. Jansen, D.L. Booth, and A. Spink. Determining the user intent of web search engine queries. In *Proceedings of the 16th international conference on World Wide Web*, page 1150. ACM, 2007.

[38] T. Joachims, C. Nedellec, and C. Rouveirol. Text categorization with support vector machines: learning with many relevant. In *Machine Learning: ECML-98 10th European Conference on Machine Learning, Chemnitz, Germany*, pages 137–142. Springer, 1998.

[39] A. Joshi and Z. Jiang. Retriever: Improving web search engine results using clustering. *Managing Business with Electronic Commerce: Issues and Trends*, page 59.

[40] A. Lang. Tolerance Rough Set Approach to Clustering Web Search Results. *Informatics and Mechanics Warsaw University*, pages 1–77, 2003.

[41] K. Lang. Newsweeder: Learning to filter netnews. In *In Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[42] D.D. Lewis, Y. Yang, T.G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[43] Y.S. Maarek, R. Fagin, I.Z. Ben-Shaul, and D. Pelleg. Ephemeral document clustering for web applications. *IBM research report RJ*, 10186, 2000.

[44] U. Manber and G. Myers. Suffix arrays: A new method for on-line string searches. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 319–327. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 1990.

[45] C.D. Manning, P. Raghavan, and H. Schtze. Introduction to Information Retrieval. 2008.

[46] I. Maslowska. Phrase-based hierarchical clustering of Web search results. *Lecture notes in computer science*, pages 555–562, 2003.

[47] I. Maslowska and R. Slowiriski. Hierarchical Clustering of Text Corpora Using Suffix Trees. In *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM'03 Conference Held in Zakopane, Poland, June 2-5, 2003*, page 179. Springer Verlag, 2003.

[48] G. Mecca, S. Raunich, and A. Pappalardo. A new algorithm for clustering search results. *Data & Knowledge Engineering*, 62(3):504–522, 2007.

[49] D. Milne and I.H. Witten. Learning to link with Wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[50] D. Milne and IH Witten. An open-source toolkit for mining Wikipedia. In *Proc. New Zealand Computer Science Research Student Conf., NZCSRSC*, volume 9, 2009.

[51] ODP-extract project at Github. http://github.com/faraday/odp-extract/. Last visited on: 8 October 2010.

[52] Open Directory Project. http://www.dmoz.org/. Last visited on: 19 May 2010.

[53] Open Directory Project - World:Türkçe. http://www.dmoz.org/World/Türkçe. Last visited on: 9 June 2010.

[54] S. OSIŃSKI. *AN ALGORITHM FOR CLUSTERING OF WEB SEARCH RESULTS*. PhD thesis, University of Technology, Poland, 2003.

[55] S. Osiriski, J. Stefanowski, and D. Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Intelligent information processing and web mining: proceedings of the International IIS: IIPWM'04 Conference held in Zakopane, Poland, May 17-20, 2004*, page 359. Springer Verlag, 2004.

[56] S. Osiriski and D. Weiss. Conceptual clustering using lingo algorithm: Evaluation on open directory project data. In *Intelligent information processing and web mining: proceedings of the International IIS: IIPWM'04 Conference held in Zakopane, Poland, May 17-20, 2004*, page 369. Springer Verlag, 2004.

[57] S. Osiriski and D. Weiss. Carrot2: Design of a flexible and efficient web information retrieval framework. In *Advances in web intelligence: Third International Atlantic Web Intelligence Conference, AWIC 2005, Lodz, Poland, June 6-9, 2005: proceedings*, page 439. Springer-Verlag New York Inc, 2005.

[58] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics Morristown, NJ, USA, 2002.

[59] X.H. Phan, L.M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. 2008.

[60] J. Pitkow and P. Pirolli. Life, death, and lawfulness on the electronic frontier. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 390. ACM, 1997.

[61] MF Porter. Snowball: A language for stemming algorithms, 2001. *URL http://snowball. tartarus. org/texts/introduction. html*.

[62] M.A. Rahurkar, D. Roth, and T.S. Huang. Which” Apple” are you talking about? 2008.

[63] C. Sacarea, R. Meza, and M. Cimpoi. Improving conceptual search results reorganization using term-concept mappings retrieved from Wikipedia. In *IEEE International Conference on Automation, Quality and Testing, Robotics, 2008. AQTR 2008*, volume 3, 2008.

[64] S. Schockaert. Het clusteren van zoekresultaten met behulp van vaagmieren (clustering of search results using fuzzy ants). *Unpublished master's thesis, University of Ghent, Ghent, Belgium*, 2004.

[65] M.J. Silva and B. Martins. Web Information Retrieval with Result set Clustering. *Lecture notes in computer science*, pages 450–454, 2003.

[66] J. Stefanowski and D. Weiss. Carrotˆ 2 and Language Properties in Web Search Results Clustering. *Lecture notes in computer science*, pages 240–249, 2003.

[67] A. Strehl. Relationship-based clustering and cluster ensembles for high-dimensional data mining. 2002.

[68] Turkish Wikipedia. Anlam ayrımı Kategorileri. http://tr.wikipedia.org/w/index.php?title=Özel:Ara&redirs=1&search=anlam+ayrımı&fulltext=Search&ns14=1&title=Özel:Ara&advanced=1&fulltext=Advanced+search. Last visited on: 9 June 2010.

[69] Turkish Wikipedia. MediaWiki:Disambiguationspage. http://tr.wikipedia.org/wiki/MediaWiki:Disambiguationspage. Last visited on: 9 June 2010.

[70] E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, 1995.

[71] CJ Van Rijsbergen. Information retrieval, chapter 7. *Butterworths, London*, 2:111–143, 1979.

[72] Vivisimo. http://www.vivisimo.com. Last visited on: 9 September 2010.

[73] J. Wang and R. Li. A NEW CLUSTER MERGING ALGORITHM OF SUFFIX TREE CLUSTERING. In *Intelligent information processing III: IFIP TC12 International Conference on Intelligent Information Processing (IIP 2006), September 20-23, Adelaide, Australia*, page 197. Springer, 2006.

[74] Y. Wang and M. Kitsuregawa. Link based clustering of Web search results. *Lecture Notes in Computer Science*, pages 225–236, 2001.

[75] Y. Wang and M. Kitsuregawa. On combining link and contents information for web page clustering. *Lecture notes in computer science*, pages 902–913, 2002.

[76] D. Weiss. Cluster Contamination Measure.

[77] D. Weiss. Introduction to search results clustering. In *Proceedings of the 6th International Conference on Soft Computing and Distributed Processing, Rzeszów*. Citeseer, 2002.

[78] D. Weiss and J. Stefanowski. Web search results clustering in polish: Experimental evaluation of carrot. In *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM'03 Conference Held in Zakopane, Poland, June 2-5, 2003*, page 209. Springer Verlag, 2003.

[79] Wikipedia. http://www.wikipedia.org/. Last visited on: 19 May 2010.

[80] Wikipedia - List of disambiguation pages. http://en.wikipedia.org/wiki/Wikipedia: Links_to_(disambiguation)_pages. Last visited on: 9 June 2010.

[81] Wikipedia related pair extractor at Github. http://github.com/faraday/wikipedia-tools/tree/master/extract-related/. Last visited on: 8 October 2010.

[82] WikipediaMiner Toolkit files on Sourceforge.net. http://sourceforge.net/projects/wikipedia-miner/files/. Last visited on: 9 September 2010.

[83] Wikiprep-ESA project at Github. http://github.com/faraday/wikiprep-esa/. Last visited on: 9 September 2010.

[84] I.H. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.

[85] Yippy clustering search engine. http://search.yippy.com/. Last visited on: 9 September 2010.

[86] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54. ACM New York, NY, USA, 1998.

[87] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to Web search results. *Computer Networks-the International Journal of Computer and Telecommunications Networkin*, 31(11):1361–1374, 1999.

[88] O. Zamir, O. Etzioni, O. Madani, and R.M. Karp. Fast and intuitive clustering of web documents. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 287–290, 1997.

[89] O.E. Zamir. *Clustering web documents: a phrase-based method for grouping search engine results*. PhD thesis, Citeseer, 1999.

[90] Zemberek 2 is an open source NLP library for Turkic languages. http://code.google.com/p/zemberek/. Last visited on: 19 May 2010.

[91] H.J. Zeng, Q.C. He, Z. Chen, W.Y. Ma, and J. Ma. Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217. ACM New York, NY, USA, 2004.

[92] D. Zhang and Y. Dong. Semantic, hierarchical, online clustering of web search results. *Lecture notes in computer science*, pages 69–78, 2004.

[93] Y. Zhang and B. Feng. Clustering Search Results Based on Formal Concept Analysis. *Information Technology Journal*, 7(5):746–753, 2008.

APPENDIX A

EXAMPLE DATA

A.1 Stop category list from Gabrilovich et al.

The categories from Wikipedia, used for preprocessing phase of Explicit Semantic Analysis (ESA), are shown in Table A.1.

A.2 Samples from AMBIENT Dataset

In the following sections, subtopics, documents and subtopic relevance data for ”JAGUAR” topic from AMBIENT dataset [8] are provided.

A.2.1 Subtopics for JAGUAR Topic

For JAGUAR topic in AMBIENT, subtopics extracted from Wikipedia are provided in Table A.2 with their descriptions. Subtopic IDs are used in Table A.3 and A.4 to map results to subtopics.

A.2.2 Search Results for JAGUAR Topic

Sample results retrieved from Yahoo! for JAGUAR topic in AMBIENT are provided in Table A.3 and A.4 with ID of corresponding subtopic (if there is one), URL and content.

A.3 Samples from ODP-TR30 Dataset

In the following sections, subtopics, documents and subtopic relevance data for ”Bilim - Sosyal Bilimler” topic from ODP-TR30 dataset are provided.

A.3.1 Subtopics for *Bilim - Sosyal Bilimler Topic*

For *Bilim* - *Sosyal Bilimler* topic in ODP-TR30, subtopics extracted from Turkish data in ODP are provided in Table A.5 with their descriptions.

A.4 Sample related pairs used for threshold selection

In the following sections, Wikipedia title pairs sampled from 1000 random, related pairs are provided.

A.4.1 Pairs from English Wikipedia

A number of pairs from 1000 randomly selected, related pairs in English Wikipedia are listed in Table A.6. These pairs are extracted from ”See Also” content, ”Related Pages” content and their corresponding ”See Also” and ”Related Articles” templates embedded in articles.

A.4.2 Pairs from Turkish Wikipedia

A number of pairs from 1000 randomly selected, related pairs in Turkish Wikipedia are listed in Table A.7. These pairs are extracted from ”Ayrıca Bakınız” sections and corresponding ”Bakınız” templates embedded in articles.

Table A.1: Stop categories used in [25]

Category ID	Category title
694492	Category:Star name disambiguations
696996	Category:America
706360	Category:Disambiguation
707272	Category:Georgia
708635	Category:Lists of political parties by generic name
720975	Category:Galaxy name disambiguations
722675	Category:Lists of two-letter combinations
787611	Category:Disambiguation categories
1039940	Category:Towns in Italy (disambiguation)
1125125	Category:Redirects to disambiguation pages
1169671	Category:Birmingham
1756037	Category:Mathematical disambiguation
1935906	Category:Public schools in Montgomery County
2031328	Category:Structured lists
2133730	Category:Identical titles for unrelated songs
2391391	Category:Signpost articles
2453533	Category:Township disambiguation
2495113	Category:County disambiguation
2620466	Category:Disambiguation pages in need of cleanup
2634660	Category:Human name disambiguation
2645680	Category:Number disambiguations
2645816	Category:Letter and number combinations
2649076	Category:4-letter acronyms
2655288	Category:Acronyms that may need to be disambiguated
2664682	Category:Lists of roads sharing the same title
2803431	Category:List disambiguations
2803858	Category:3-digit Interstate disambiguations
2826432	Category:Geographical locations sharing the same title
2866961	Category:Tropical cyclone disambiguation
2891248	Category:Repeat-word disambiguations
2900842	Category:Song disambiguations
2906246	Category:Disambiguated phrases
2907532	Category:Subway station disambiguations
2907812	Category:Lists of identical but unrelated album titles
2909071	Category:5-letter acronyms
2911539	Category:Three-letter acronym disambiguations
2929221	Category:Miscellaneous disambiguations
3055424	Category:Two-letter acronym disambiguations
952890	Category:Days
1712083	Category:Eastern Orthodox liturgical days

Table A.2: Subtopics of JAGUAR from AMBIENT dataset

Subtopic ID	Subtopic Name	Description
16.1	Jaguar(Panthera onca)	a New World mammal(a”big cat”) of the Felidae family native to South and Central America
16.2	Jaguar(car)	a British luxury car manufacturer, owned by Ford as of 1990
16.3	Jaguar	the mascot of Owens Community College in Toledo and Findlay, Ohio
16.4	Aimée and Jaguar	a character in the 1999 German war and drama movie
16.5	Atari Jaguar	a video game console made by Atari
16.6	Fender Jaguar	guitar introduced in 1962, built by Fender
16.7	Jaguar(cartoonist)	Sérgio Jaguaribe, a Brazilian cartoonist
16.8	Jaguar	a pseudonym for the German musician Alec Empire
16.9	Jaguar	the Transformers character Ravage name in the Japanese version
16.10	The Jaguar(Impact Comics)	a DC Comics superheroine
16.11	Jaguar	a brief incarnation of the Joshua Perahia fronted band Joshua(band).
16.12	Jaguar(rocket)	a British elevator research rocket
16.13	Jaguar	the codename for Mac OS X v10.2
16.14	JAGUAR	a computational chemistry software program
16.15	Jaguar 1 and Jaguar 2	German tank destroyers
16.16	Jaguar class fast attack craft	German S-boats
16.17	SEPECAT Jaguar	a military aircraft
16.18	XF10F Jaguar	Grumman F10F Jaguar, a military aircraft
16.19	Claas Jaguar	a range of forage harvesting equipment by German manufacturer Claas
16.20	Jacksonville Jaguars	a professional American football(NFL) team based in Jacksonville, Florida
16.21	South American Jaguars	a combination international rugby union team in the 1980s.
16.22	Calico	alternative name for the Jaguar cat

Table A.3: Search Results and Associated Subtopics for JAGUAR topic from AMBIENT dataset

Result ID	Subtopic ID	URL	Result
16.1	16.2	http://www.jaguar.com/	Jaguar Official site of the Ford Motor Company division featuring new Jaguar models and local dealer information.
16.2	-	http://www.oneworld journeys.com/jaguar	One World Journeys — Jaguar: Lord of the Mayan Jungle A multimedia expedition into the heart of the Mexican jungle, searching for the elusive jaguar.
16.3	16.1	http://www.bluelion.org/jaguar.htm	Jaguar Compares jaguars and leopards and provides information about the animal's shrinking habitat and relationship with man.
16.4	16.1	http://lynx.uio.no/lynx/catsgportal/cat-website/catfolk/onca-01.htm	Jaguar (Panthera onca) Provides information on the Jaguar, the largest cat of the Americas. Covers the Jaguar's physical features, behavior, habitat, distribution, and population status.
16.6	16.2	http://www.jaguar.com/us/en/home.htm	Jaguar US - Home Jaguar USA Official Home Page ... Build Your XK. Build Your Jaguar. Request Brochure. Get Email Updates. Locate a Dealer. Search Your Profile Site Map Contact Us ...
16.7	16.2	http://www.jaguar.co.uk/	Jaguar UK - Jaguar Cars XK. XJ. S-TYPE. X-TYPE. Used. Latest. Jaguar &amp;amp;amp; Ownership. Highlights. Gallery. Models &amp;amp;amp; Pricing ... SEARCH SITEMAP COMPANY Privacy Policy Accessibility ...
16.21	16.2	http://media.ford.com/brand.cfm?make_id=95	Media.Ford.com: The Products :Jaguar OFFICIAL NEWS, PHOTOS, VIDEOS, MEDIA KITS, EXECUTIVE BIO&amp;amp;146;S, PRESS RELEASES - Ford, Volvo, Mazda, Lincoln, Jaguar, Mercury, Land Rover
16.22	16.17	http://www.fas.org/man/dod-101/sys/ac/row/jaguar.htm	Jaguar ... and tactical support aircraft, the Jaguar has been transformed into a potent fighter-bomber. ... The Jaguar strike fighter was equipped also with Magic air ...
16.47	-	http://www.okayplayer.com/jaguarwright/	Okayplayer: Jaguar Wright - Official Web site News, audio, video, and photo gallery. ... Also look for Jaguar on tour this fall. ... Let Jaguar get you open when you read her editorial in Billboard magazine. ...

Table A.4: Search Results and Associated Subtopics for JAGUAR topic from AMBIENT dataset - 2

Result ID	Subtopic ID	URL	Result
16.48	16.13	http://www.apple.com/pr/library/2002/may/06jaguar.html	Apple Previews ”Jaguar,” the Next Major Release of Mac OS X ... of Mac®OS X, code-named ”Jaguar,” to more than 2,500 Macintosh developers ... Jaguar” will be available to customers in late summer 2002, and will further ...
16.62	16.13	http://www.amazon.com/Mac-10-2-Jaguar-Old-Version/dp/B00006F7S2	Amazon.com: Mac OS X 10.2 Jaguar [Old Version]: Software ... version, mac osx operating system, macintosh office suites, mac os x jaguar ... included with every copy of Jaguar, empowering Java, C, and AppleScript Studio ...
16.63	16.5	http://www.gamewinners.com/JAG/index.html	Game Winners - Atari Jaguar cheats, codes, hints, walkthroughs, FAQs Cheats, codes, hints, walkthroughs, and FAQs for the Atari Jaguar video game system. ... Get more help and discuss various Jaguar games in our Classic Systems Forum. ...
16.83	16.6	http://www.fender.com/products/search.php? section=guitars&amp;cat=jaguar	.:: Fender®.com ::. Official Website of Fender Musical Instruments ... Jaguar®American Vintage. Special Edition. Jazzmaster®Mustang®Other Guitars. Artist Models ...

Table A.5: Search Results and Associated Subtopics for *Bilim - Sosyal Bilimler* topic from ODP-TR30 dataset

Result ID	Subtopic ID	URL	Title	Snippet
4.3	4.59	http://www.psiko farmakoloji.org	Klinik Psikofarmakoloji Bülteni	Derginin ilgili konulardaki bilimsel makaleleri ve mesaj panosu.
4.3	4.60	http://www.kisisel basari.com/	Kişisel Başarı Eğitim Danışmanlık	Psikoloji, eğitim, sağlık, bilim, insan kaynakları ve kişisel gelişimle ilgili makaleler, haberler ve bir tartışma forumu içeriyor.
4.3	4.61	http://www.sanal psikolog.com/	Sanal Psikolog	Psikopatoloji, afet sonrası, aile, çocuk, kadın, iletişim ana konuları ve psikoloji alanında çeşitli yazılara ek olarak soru-cevap ve anket sayfaları sunuluyor.
4.4	4.70	http://www.akmed. org.tr/	Akmed	Akdeniz Medeniyetleri Araştırma Enstitüsü.
4.4	4.71	http://mezopotamya. tripod.com/	Mezapotamya	Arkeoloji metinleri, makaleleri ve haberleri.
4.4	4.72	http://solikilikia. 8m.com/	Soli-Pompeiopolis Antik Kazıları	Soli-Pompeiopolis antik kent kazıları hakkında bilgiler yer alıyor.
4.6	4.82	http://www.felsefe seminerleri.com	Felsefe Seminerleri	Thales'ten günümüze filozoflar ve ekoller hakkında bilgiler, bilgelik hikayeleri, özlü sözler ve bağlantılar içeriyor.
4.6	4.84	http://www.tfk.org.tr/	Türkiye Felsefe Kurumu	Kurumun yayınlarına, etkinliklerine ve kurumla ilgili haberlere yer verilmektedir.
4.6	4.85	http://www.ayrinti.net/ nietzsche	Nietzsche	Nietzsche'nin yaşam öyküsünün, felsefesinin, eserlerinin, fotoğraflarının ve aforizmalarının bulunduğu bir sitedir.
4.9	4.96	http://www.mustafa aksoy.com/	Altaylardan Anadoluya Damgalar	Mustafa Aksoy'un Türkiye'de -özellikle Doğu ve Güneydoğu Anadolu'da-, Türk cumhuriyetlerinde ve bazı ülkelerde saha araştırmaları yaparak hazırladığı, kültür sosyolojisi, Türk sanatı ve etnografya konusunda bir site.
4.9	4.99	http://www.sosyal hizmetuzmani.org/	Sosyal Hizmet Uzmanı	Sosyal hizmet uzmanlığı ile ilgili çok kapsamlı incelemeler içeren sitede psikolojik ve sosyolojik yazılar ile toplumsal sorunlara çözüm önerileri sunuluyor.

Table A.6: Sample Related Pairs from English Wikipedia

Title 1	Title 2
Beckmann rearrangement	Schmidt reaction
Chi (Chobits)	Chobits
Microfungi	Hyphae
Sensory illusions in aviation	Pilot error
Kusići, Zenica	Jezera, Teslić
Lapsed Catholic	Catholic guilt
Headingley	Headingley Stadium
Israel and the apartheid analogy	Israeli West Bank barrier
Édouard Lucas	Lucas-Lehmer test
WikidPad	Personal wiki
Set-top box	Cable Converter Box
Defection	Renegade
Ulnar artery	Allen test
Modular synthesizer	Synthesizer
Elementary symmetric polynomial	Representation theory
Outliner	Mind map
GNU Prolog	SWI-Prolog
Lambeth Waterworks Company	London water supply infrastructure
Transference	Countertransference
Maculinea alcon	Maculinea alcon arenaria
Athens, Tennessee	Tennessee Wesleyan College
American craft	Glass art
Tuzk-e-Jahangiri	Akbarnama
Backplane	Switched fabric
Yap Kwan Seng	Yap Ah Loy
Cannon Air Force Base	Tactical Air Command
Travelling exhibition	National Touring Exhibitions
Doubletracking	Punching in
Code monkey	Real programmer
Kappa (company)	Fila (company)
La mian	Chinese noodles
Heart failure	Cardiogenic shock
Mount Tahan	Gunung Yong Belar
Cap Arcona	Hell ship
PSTricks	LaTeX
Object relations theory	John Bowlby
Abkhazian wine	Georgian wine
Chromosomal translocation	Chromosome abnormalities
National Rural Employment Guarantee Act	NREGS (Kerala)
YouTube API	YouTube
Homage (medieval)	Allegiance
Lake Khaiyr	Lake monster

Table A.7: Sample Related Pairs from Turkish Wikipedia

Title 1	Title 2
Eksiklik teoremi	Yinelgen
James Clerk Maxwell	Maxwell köprüsü
Çarukluğ boyu	Çaruk
Steve Biko	Güney Afrika Cumhuriyeti
Kemal Kerinçsiz	Türk Ceza Kanunu 301. maddesi
Prince George, British Columbia	Britanya Kolombiyası
Cıvata	Arşimet spirali
Ap ve Bp yıldızı	Yıldız sınıflandırma
Nesne Yönelimli Programlama	Nesne tabanlı programlama dili
Ottawa, Ontario	Ontario
Trabzon (merkez)	Trabzon Kuleleri
1. Afife Tiyatro ödülleri	Afife Tiyatro ödülleri
Aile hukuku Cezayir	Aile Yasası
Simple Portal	Simple Machines Forum
1421 (kitap)	çin coğrafi keşifleri
Sabit zaman	Logaritmik zaman
Laçın Koridoru	Qashatagh
Köprülü Amcazade Hacı Hüseyin Paşa	Amcazade Yalısı
Akyaka, Ula	Nail çakırhan
Türkiye'de Yahudilik	Sabetaycılık
Venüs (mitoloji)	Afrodit
Lokma (tatlı)	Tulumba tatlısı
Rhythm and Blues	Blues
Gökada kümesi	Gök adalar dizini
Deizm	Agnostisizm
Çin-Japon Savaşı	Japon-Sovyet Çatışmaları
Uçak mühendisliği	Uzay mühendisliği
1998 Dünya Halter şampiyonası - Erkekler 77 kg	2000 Yaz Olimpiyatları/Halter
Hughes-Ryan Yasası	İstihbarat Gözetimi Yasası
Brandon, Manitoba	Manitoba
Yorumlanan programlama dili	Yorumlayıcı
Politik iktisat	İktisadi İdeoloji
Cauchy integral teoremi	Cauchy-Riemann denklemleri
Rabindranath Tagore	Gandhi
Atopik dermatit	Ekzema
Georg Cantor	Cantor paradoksu
Gökbilim	Gök mekaniği
Kâfir	Müşrik