University of Bergen

Master Thesis

Optimal Compact Trie Clustering - A genetic approach

Author:
Snorre Magnus Davøen

Supervisor:
Richard Elling Moe

in the

Department of Information Science and Media Studies

September 19, 2013

“Quote comes here"

Quote Author

UNIVERSITY OF BERGEN

Abstract

Faculty of Social Sciences

Department of Information Science and Media Studies

Masters degree

Optimal Compact Trie Clustering - A genetic approach

by SNORRE MAGNUS DAVØEN

Abstract comes here

Acknowledgements

Acknowledgements comes here

Contents

List of Figures

List of Tables

List of Listings

Nomenclature

CTC	Compact Trie Clustering
GA	Genetic Algorithm

Chapter 1

Introduction

Information Retrieval is a field in information, informatics and computer science that revolves around retrieving and classifying information in order to make information more accessible. Theories and practices developed in this field drives many of the information retrieval systems we use in our every day lives such as Google Search, TinEye (an image search engine), and many more. One major area of information retrieval is classification and grouping (or clustering) of text documents or other information artifacts. Clustering will be the subject of this master thesis.

This master thesis started as a master thesis proposal by my supervisor Richard Elling Moe . The proposal suggested that work be put into optimizing the Suffix Tree Clustering algorithm. The Suffix Tree Clustering algorithm was presented by Zamir and Etzioni (1998) in the paper "Web document clustering: a feasibility demonstration." The algorithm is a clustering algorithm which extracts phrases from text documents and finds clusters by inserting these phrases into a suffix tree. Richard Elling Moe has worked with the Suffix Tree Clustering algorithm in relation to a research project at the Department of Information Science and Media Studies. The project is described in the paper "Resirkulering av nyheter I nettavisene 1999 – 2009," (Elgesem, 2009).

In this work he experimented with the Suffix Tree Clustering algorithm and implemented the Compact Trie Clustering algorithm which does not limit itself to suffixes. The original Suffix Tree Clustering algorithm proposed by Zamir and Etzioni (1998) is demonstrated in use on search engine results and showed good results. In his work with the Klimauken corpus, a corpus comprising articles from several big Norwegian newspapers, Richard

Elling Moe found that the Suffix Tree Clustering algorithm yielded poorer results. This can be explained as search engine results have been pre-filtered by the search engine. The Klimauken documents are quite varied in content and as such the default implementation and parameters of the Suffix Tree Clustering algorithm was not sufficient to get good results. This is the background for this master thesis were an approach to finding a near optimal parameter set for a specific corpus will be explored.

1.1 Motivation

1.1.1 Academic motivations

The Suffix Tree Clustering algorithm has some benefits over many of the traditional clustering algorithms. It is computationally fast having a $O(n)$ computation time where n is the number of documents. It is also phrase based which means it takes into account the position of words when comparing the similarity of documents. Traditional clustering algorithms often use the bag of words model where each document is only a set of words with positional data disregarded. The drawback is however that the Suffix Tree Clustering algorithm has shown poor performance on unfiltered text corpora. It is of interest to see if the algorithms performance can be improved. There are evidence to suggest that changing parameters for the algorithm might improve results. This master will propose an automated way in which to find better optimized parameter sets for different corpora. This method of improvement could benefit researchers who use the algorithm as part of their work to identify text clusters.

1.1.2 Other motivations

I had previously taken two courses about information retrieval and web intelligence. This area of research is very interesting and this master thesis provided an opportunity to learn more about it. I also wanted to use previous knowledge in my master thesis, so I proposed to use the Genetic Algorithm to identify good parameter sets as genetic algorithms are well suited to the task of exploring large feature spaces. I also got the opportunity to learn a new programming language, Python, as this was already used by Richard Elling Moe .

1.2 Research question

The main research question of this thesis is:

"What are the optimal parameter values for Compact Trie Clustering with regard to the news corpus?"

The main goal of this master thesis is to identify an optimized parameter set for the Compact Trie Clustering algorithm. The number of possible parameter sets are extremely big, so there is no realistic way to computationally prove that the most optimal parameter set has been found. Instead the thesis will identify an optimized parameter set that is significantly better than the default, and an approach to finding such optimized parameter sets. An experiment will be performed to verify the results.

The thesis has two subsidiary goals:

- Develop a method for determining optimal parameters for the Compact Trie Clustering algorithm.
- Apply the method to determine recommended parameter values for news documents.

This thesis is for researchers and users of the Suffix Tree Clustering algorithm. With the optimization approach discussed in this thesis they will be able to find better parameter sets for their corpora. The thesis is also very much aimed at the research group working with the project discussed in “Resirkulering av nyheter I nettavisene 1999 – 2009” and other information retrieval researchers.

This research is performed as part of a master thesis. As such certain temporal and economic constraints are imposed upon the scope and detail of the research. There was not enough time to test more than two different corpora. I also opted to only use corpora that were free of charge as this would negate the need for funds. The tests were run on personal machines which made time efficiency comparisons between parameter sets difficult.

Chapter 2

Theory

In this chapter the thesis will introduce the basic concept of clustering underpinning all clustering methods. It will formally define clustering and explain how the Suffix Tree Clustering algorithm fit into this definition. It will then move onto explaining the Suffix Tree Clustering algorithm itself and the different stages of which it is comprised. It is in these sections that the parameters to be optimzed will be explained. The theory chapter will also go into some detail about the Compact Trie Clustering algorithm, a slightly modified version of the original Suffix Tree Clustering algorithm. An investigation into performance measures used in information retrieval in general and clustering in particular will provide a foundation for how one can test the Suffix Tree Clustering and Compact Trie Clustering algorithms. The section will also go shortly into two performance measurements suitable for Compact Trie Clustering . The chapter will also provide a short section about different corpora available for information retrieval research.

In my thesis work a genetic approach to optimization of the parameters was tested. The theory chapter will therefore provide a section about the Genetic Algorithm and how it can be used to optimize parameters.

2.1 Clustering and information retrieval

Baeza-Yates and Ribeiro-Neto defines text clustering as, "*[...]given a collection D of documents, a text clustering method automatically separates these documents into K*

clusters according to some predefined criteria". The variable K here refers to the number of clusters produced by the clustering algorithm given a document set. The variable D refers to the document set comprising the documents to be clustered. Given a size N of the document set D a clustering algorithm might produce a cluster set where $K \in \{1, .., N * N\}$. In other words, a clustering algorithm might produce anywhere from a single cluster to as many clusters as there are document combinations.

The K variable can either be pre-determined and given to the clustering algorithm as a variable as is the case with K-Means Clustering. In other algorithms the K variable is undefined and varies according to different criteria such as the document collection size, the contents of the documents, the parameters given to the clustering algorithm etc.

The Suffix Tree Clustering and Compact Trie Clustering algorithms conform to this definition of clustering. Examples of other clustering algorithms that fall under this definition include the previously mentioned K-Mean algorithm and the Hierarchical Clustering algorithm.

2.1.1 Suffix Trees and Suffix Tree Clustering

The Suffix Tree Clustering algorithm was first introduced by Zamir, Etzioni, Madani, and Karp (1997) in the paper "Fast and Intuitive Clustering of Web Documents." This article discuss how suffix tree clustering can be used on search engine results to improve the results. Later an improved version of the algorithm was presented in the paper "Web document clustering: a feasibility demonstration" (Zamir & Etzioni, 1998). In this later paper Zamir and Etzioni describes the requirements for the Suffix Tree Clustering algorithm and the stages involved in Suffix Tree Clustering . They also compare the effectiveness (i.e. performance) of the algorithm compared to other clustering algorithms.

The Suffix Tree Clustering algorithm has three basic steps:

1. Document cleaning

2. Suffix tree and Base Cluster creation

3. Base Cluster merging

2.1.1.1 Document Cleaning

Document cleaning involves cleaning the strings representing each document. This is done by stemming each work, marking sentences and removing non-word tokens such as HTML tags, numbers and punctuation. The strings comprise the document snippets. Each snippet is cleaned string from the original document. There are some possible algorithmic parameters that can be identified here. For example, which parts of the documents should be extracted? Using more of the text document for snippet extraction gives the algorithm more data to work with which could yield more accurate results. There is also the question of which parts of the documents that are the best signifier of the content of that document. A nice parameter to think of here would thus be which parts of, say a news document, should be included such as titles/headings, image captions, article introductions, article contents etc. Other possible parameters that has not been investigated are possible stemming techniques (lemmatisation vs. stemming) and differing stop word lists.

2.1.1.2 Suffix Tree and Base Cluster Creation

To make the concept Suffix Trees a bit clearer a short explanation of the trie data structure and suffixes will be provided.

The Suffix Tree Clustering algorithm use a trie data structure.

!TODO: INSERT TRIE DEFINITION HERE!

In context of a term $t = [c_1, ..., c_n]$ a suffix is a subterm $s = [t[m], ..., t[n]]$ where m is smaller than or equal to (in which case only one character is selected) n. To exemplify this definition the term suffix itself can be used. The suffixes of the term "suffix" are *1*) suffix; *2*) uffix; *3*) ffix; *4*) fix; *5*) ix; and *6*) x .

Zamir and Etzioni, 1998 treat documents as a collection of snippets where each snippet is a normalized sentence from a document. Each snippet contains a sequence of words. The Suffix Tree Clustering algorithm extracts its suffixes from phrases (snippets) rather than single terms. A suffix in this context would thus be defined as all the sub-phrases of a given phrase. Following the same rules as apply for the example above the phrase "clustering is fun" therefore has the following suffixes: *1*) clustering is fun; *2*) is fun; and *3*) fun .

By combining the concepts of suffixes and trie structures you can build a suffix tree. Zamir and Etzioni, 1998 formally define a suffix tree of a string S with the following requirements:

- Suffix trees are rooted and directed.
- All internal nodes have at least two children.
- Each edge is labeled with a non-empty substring (suffix) of S.
- A node's label is defined as the concatenated labels of the nodes in the path from the root node to the given node.
- Edges from the same node can not have the same edge-labels (sub-phrases)
- For each suffix of the string S there is a suffix node which is equal to that suffix.

Following these requirements, a suffix tree can then be understood to be a tree wherein edges are sub-phrases of or whole suffixes and the nodes connected by these are the documents (or sources) from which these suffixes come. The internal nodes in the tree are phrase clusters made up of all those documents that share that sub-phrase. An example of a suffix tree can be seen in Figure 2.1. Here three phrases each comprising a document are split into suffixes and then put into the suffix tree structure. As an example one can see that the two phrases *1*) "cat ate cheese" and *2*) "cat ate mouse too" share the phrase "cat ate". This is represented in the graph (Figure 2.1) by node a which contains pointers to the first suffix of both phrase 1 and phrase 2.

Each internal node is a phrase cluster made up of all the documents that share that phrase (i.e the union of the documents in it's descendant nodes). Each node makes up a base cluster Zamir and Etzioni, 1998. The base clusters are scored according to the scoring function:

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

where $|B|$ is the number of documents in the cluster B and $f(|P|)$ is a function on the length of the cluster phrase P (excluding stop words) which penalizes short phrases ($|P| < 2$), gives a linear score for regular phrases ($|P| = 2, \ldots, 6$) and a constant score

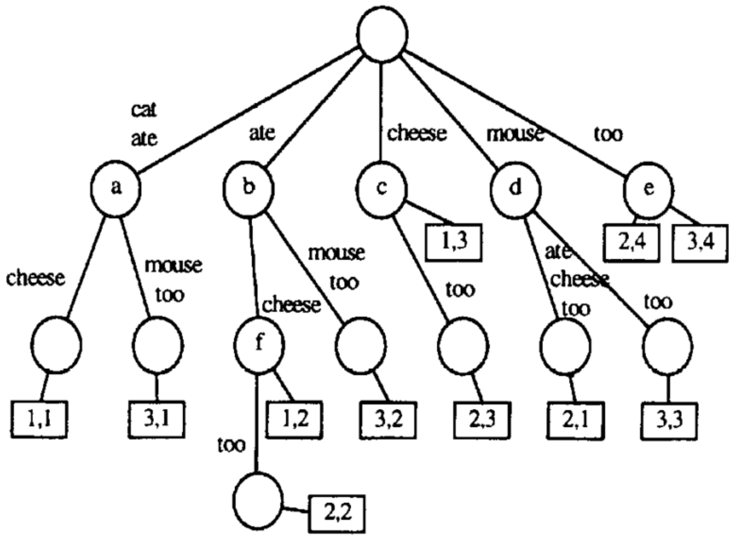

FIGURE 2.1: A suffix tree generated from the strings "cat ate cheese", "mouse ate cheese too" and "cat ate mouse too". From "Web document clustering: a feasibility demonstration" (Zamir & Etzioni, 1998, p. 48)

to longer phrases ($|P| > 6$). Clusters with many documents and/or long phrases receive higher scores than clusters with few documents and/or short phrases. A promising parameter identified in this step would be the score-threshold used by the $f(|P|)$-function. The scoring-threshold determines which of the terms in a phrase that contributes to that phrase's length.If a term is contained within 3 or less documents or more than 40% of the documents in the collection, then the term receives a score of zero and does not contribute to the phrase's length. This could influence the scores of base clusters as some base clusters could become longer or shorter. The base cluster score is used to sort the base clusters for the final step in the Suffix Tree Clustering algorithm.

2.1.1.3 Base Cluster Merging

Base cluster merging is the final step in the Suffix Tree Clustering and is an improvement of the original algorithm. Zamir and Etzioni note that,

> Documents may share more than one phrase. As a result, the document sets of distinct base clusters may overlap and may even be identical. To avoid the proliferation of nearly identical clusters, the third step of the algorithm merges base clusters with a high overlap in their document sets [...] Zamir and Etzioni, 1998, p. 3

Clusters are merged based on their similarity. Zamir and Etzioni use the Jaccard Similarity Coefficient which is defined as $\frac{|X \cap Y|}{|X \cup Y|}$, where the similarity can vary between zero if there are no common elements in set X and Y and one if the two sets have all elements in common Van Rijsbergen, 1979.

Given two base clusters $B_(m)$ and $B_(n)$ and the formula for the Jaccard Similarity Coefficient the similarity between the two clusters can be calculated with:

$$J_m(B_m, B_n) = \frac{|B_m \cap B_n|}{|B_m|}$$

$$J_n(B_m, B_n) = \frac{|B_m \cap B_n|}{|B_n|}$$

Iff $J_m > 0.5$ and $J_n > 0.5$ then the similarity of the two clusters m and n is equal to one. All base cluster pairs with a similarity of 1 is connected in a base cluster graph. This results in a set of connected components in the graph. Each connected component is considered a cluster in the final clustering result. In this last step two parameters can be identified. One could adjust the number of base clusters used to create the final clusters. The lowest scoring base clusters might not be relevant for the final results as the documents in these clusters share few of their phrases with each-other. On the other hand it might hurt the results if the algorithm use too few of the base clusters resulting in a lower number of merged clusters, or even less accurate merged clusters.

It is also possible to adjust the similarity threshold used by the Jaccard coefficient, or use another similarity measure all together. One such similarity measure is the cosine similarity measure that can be used when documents are represented in a vector space model. This similarity measure is used by Chim and Deng, 2007, albeit with phrases in a hierarchical clustering algorithm. The cosine similarity formula returns a value between -1 and 1, where -1 denotes that the vectors being compared are completely dissimilar, 1 that they are identical, and 0 that they are somewhere in between. The cosine similarity value can easily be applied as a similarity measure when combining clusters.

2.1.2 Compact Trie Clustering

The Compact Trie Clustering algorithm is a slightly modified version of the Suffix Tree Clustering algorithm, developed by Richard Elling Moe . The algorithm is so named because it use k-grams instead of suffixes, and because it also follows the compact tree structure. A k-gram is a subsequence of k number of characters occurring in a sequence of characters (a word or term). In the word “mouse” all the 3-grams would be (1) mou (2) ous (3) use (ignoring start and end $-signs) Manning, Raghavan, and Schütze, 2009. A k-gram can also be applied on sentences. the k-grams of a sentence is all the subsequences of k words occurring in that sentence. The 3-grams of the sequence “mouse ate cheese too” is (1) mouse ate cheese (2) ate cheese too . Tests performed by the LLI research group indicate that using k-grams in the expansion phase may yield better time performance when compared with suffix expansion. Because speed is one of the requirements set by Zamir and Etzioni, 1998 for web document clustering it is interesting to investigate the use of k-grams for snippet expansion.

2.1.3 Performance measures

What kind of performance measures are used for clustering...

2.1.4 Available corpora

Which corpora are used in clustering and/or classification research? Which ones are suited to clustering? Which ones are used in this master thesis research? Explain scope...

2.2 Genetic Algorithms

General overview of a genetic algorithm here.

2.3 Related Work

Introduce related research work in this chapter.

Chapter 3

Methodology

An introduction to the chapter.

3.1 Experimental Evaluation

Will rewrite and use the corresponding section from the project proposal.

3.1.1 Evaluation Measures

Provide info about the two forms of evaluation measures used (one from Improved suffix tree clustering article and one used by Richard). How do they work, what are the differences...

3.2 Corpora

Write a bit about the corpora used in the research (most likely just Klimauken and the reuters set). If time permits it could be feasible to use an additional news corpus.

3.3 Experimental Research

Rewrite and use corresponding section from project proposal. Try to flesh out the methodology a bit (how did I perform the testing, what where the hypotheses etc). Talk about experimental constraints and data used.

Chapter 4

Development and Testing

Chapter introduction. Describe the development process in some detail. I.e. The learning phase (learning Python and familiarizing myself with the algorithm). The second stage (modifying the algorithm slightly, refactoring). Third stage (implementing a genetic algorithm with which to test different parameter sets). Fourth stage (making a distributed version of the algorithm to make testing of larger parameter sets feasible/possible). Fifth stage (final touches on the algorithm, implementing support for converting other corpora to the snippet file format). (Sixth stage) Storing results in a database and developing a method of extracting statistical numbers. Overview of system?

Describe how the algorithm and parameters where tested. I.e. how data and results were gathered. Write about different testing stages. First stage (testing individual parameter options to determine usable value ranges. Are results correct, why?). Testing of non-distributed genetic algorithm (does the small test-case give indication of viability?). Testing distributed algorithm (larger parameter sets tested show better results?). Final testing (testing hypotheses, include two corpora?).

4.1 Stages of development

This section will outline the main sequence in which work on was done. The overview of the iterations will explain when the initial development of the different parts were started to give an short explanation of the development process.

4.1.1 Learning stage

Familiarize with algorithm ...

4.1.2 Modification of algorithm

Modified so and so ...

4.1.3 Genetic algorithm

Explain how and why it was developed as it was ...

4.1.4 Distribution of genetic algorithm

Why distribute? How? Possible problems...

4.1.5 Results storage and corpus processing

Storing results, tracking top chromosomes over generations, extracting averages for graphs etc.

4.1.6 Overview of the completed system

Give a short overview of the system ...

4.2 Testing

Give overview of testing process

4.2.1 Value Range Tests

Describe tests to discover reasonable parameter ranges ...

4.2.2 Genetic Algorithm Test

Describe first test with about 200 individuals and 50 generations.

4.2.3 Distributed Genetic Algorithm Test

Describe distributed test with more individuals and more generations

4.2.4 Final testing

Describe final test and how it answers resesearch question

Chapter 5

Analysis and Discussion

Chapter introduction. Should here provide some information about which parts of the work are going to be discussed. Should talk about the test results and how they correspond to the hypotheses. I.e. Does the testing reveal that a better parameter set has been found than the default one. Does this parameter set perform better than the default in different corpora? (Should perhaps test on two additional corpora?). This chapter should also investigate wether the test results are statistically significant (can I say yes or no on the null hypotheses?). Give definitive or estimated answer pending test results.

5.1 Results

Summarize and discuss results.

5.2 Validity and relevance

Show that data gathered are both valid and relevant. I.e. is the method of research rigorous and correct (methods of data gathering and testing). And does the data answer the hypotheses. Also discuss the statistical significance of the data in relation to hypotheses.

5.2.1 Data autenticity

Discuss how the validity of data should not be an issue even though the algorithm is distributed. (I.e. results from clients are validated). Algorithm deterministic...

5.2.2 Effects of two different measurements

Discuss how the varying measurements might affect the results... Does using one measurement over the other invalidate results? Should both be used (one to measure single category documents, the other to measure multiple category documents)?

Chapter 6

Summary and Conclusion

Summarize motivation

Restate research question ""

6.1 Results

Summarize results

6.2 Future research

What did I not have time to use? What was out of scope for this thesis? What would I like to investigate further.

6.3 Conclusion

Final remarks

Bibliography

Baeza-Yates, R., & Ribeiro-Neto, B. (2011). Text Classification. In *Modern information retrieval: the concepts and technology behind search (acm press books)* (2nd ed., Chap. 8, pp. 281–335). Addison Wesley.

Chim, H., & Deng, X. (2007). A new suffix tree similarity measure for document clustering. In *Proceedings of the 16th international conference on world wide web* (pp. 121–130). WWW '07. New York, NY, USA: ACM. doi:10.1145/1242572.1242590

Elgesem, D. (2009). Resirkulering av nyheter I nettavisene 1999 – 2009.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). Wildcard queries. In *An introduction to information retrieval* (Chap. 3, 100, pp. 51 –56). Cambridge University Press. Retrieved from Cambridge University Press Cambridge, England: http://dspace.cusat.ac.in/dspace/handle/123456789/2538

Van Rijsbergen, C. J. (1979). *Automatic Classification* (2nd ed.). London: Butterworths.

Zamir, O., & Etzioni, O. (1998). Web document clustering: a feasibility demonstration. In *Proceedings of the 21st annual international acm sigir conference on research and development in information retrieval.* Melbourne, Australia: ACM. doi:10.1145/290941.290956

Zamir, O., Etzioni, O., Madani, O., & Karp, R. M. (1997). Fast and Intuitive Clustering of Web Documents. In *Proceedings of the third international conference on knowledge discovery and data mining.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.9803