Clustering in a News Corpus

Richard Elling Moe

Department of information science and media studies
University of Bergen

Abstract. We adapt the Suffix Tree Clustering method for application within a corpus of Norwegian news articles. Specifically, suffixes are replaced with *n*-grams and we propose a new measure for cluster similarity as well as a scoring-function for base-clusters. These modifications lead to substantial improvements in effectiveness and efficiency compared to the original algorithm.

1 Background

This investigation came about as part of a project with the long term goal to model the flow of news in Norwegian online newspapers over time and to visualize the concentration of coverage related to various topics. An essential question is then how much overlap and recirculation there is in news production.

Since 2006 the Norwegian Newspaper Corpus [9] has downloaded the front pages of the 8 largest online newspapers and stored them in HTML format. From this, a sample corpus consisting of the daily 10 top stories from December 7 to 18, 2009 had been extracted and prepared for experimentation [6]. A total of 960 articles had been manually coded based on categories that are used by media scholars to classify news, cf [1] and [3]. Each article received a tag, consisting of five categories, characterizing the content of the article. For example

$$International - Economy - FinanceCrisis - Debt - Dubai$$

The data had been further pre-processed by reducing words to their ground form and keeping only certain kinds of words: nouns, verbs, adjectives and adverbs. This was achieved by marking up the text with syntactic information using the Oslo-Bergen tagger [10] and subsequently filter the document to leave only the desired words in the desired form.

The ability to cluster documents on the basis of having similar content would be instrumental to our goal of detecting reuse and overlap. Therefore we have explored the application of a clustering technique to our news corpus.

Zamir and Etzioni [12] demonstrate that documents can be clustered by applying the Suffix Tree method to short excerpts from them, referred to as *snippets*. This is an attractive feature in our context since such snippets may be readily available for news articles in the form of front-page matter such as headlines, captions and ingresses. For this reason we chose to adapt their use of Suffix Tree Clustering and also because they report it to outperform a number of other algorithms. More recently, Eissen et al. [2] present a more nuanced picture. They point out that the technique has some weaknesses

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 301–307, 2014.

but maintains that these have little impact when applied to shorter texts and therefore represent no great problem in our specific context.

The current investigation is a continuation of our previous reports [8] on the initial charting of territory and [7] exploring the potential for improving the Suffix Tree Clustering in general terms. Now the focus is on the Norwegian Newspaper Corpus and the adaptation of the technique to that specific context in order to further improve its performance.

2 Suffix Tree Clustering

The backbone of Suffix Tree Clustering is the data structure known as a *compact trie* [11]. A trie is a tree for storing sequences of tokens. Each arc is labelled with a token such that a path from the root represents the sequence of labels along the path. This simple structure effectively represents sequences and their subsequences as paths whereas the branching captures shared initial sequences. Note that a path does not necessarily represent a *stored* sequence. A stored sequence will have an end-of-sequence mark attached to its final node. The trie structure can be refined for the purpose of saving space. The idea is that sections of a path containing no end-of-sequence marks and no branching can be collapsed into a single arc. The *compact* trie thus allows arcs to be labeled with sequences of tokens. Figure 1 shows the compact trie for the sequences *aa*, *ab*, *abab*, *abc*, *babb*, *bc*, *cbba*, and *cbbc*, with the S_i as end-of-sequence marks.

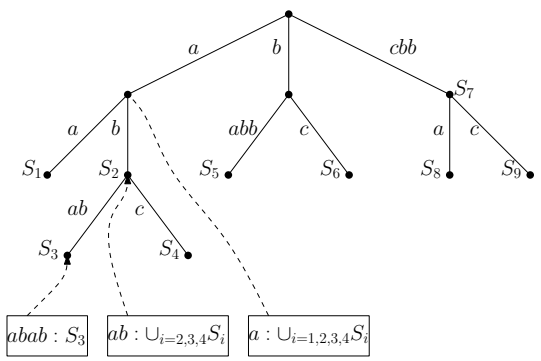

Fig. 1. A compact trie and base-clusters

The suffix tree employed by Zamir and Etzioni is a compact trie storing all the suffixes of a set of given *phrases*, i.e. the snippets. That is, the arcs are labeled with sequences of *words*. Furthermore, the end-of-sequence mark is now the set of documents that the phrase occurs in. (In practice, the set of document ID's.)

The suffix tree forms the basis for constructing clusters of documents. Each node in the tree corresponds to a *base-cluster*. A base-cluster $\sigma : S$ is basically the set S of documents associated with the subtree rooted in the node. The *label* σ is composed

of the labels along the path from the root to the node in question. Three examples of base-clusters are illustrated in Figure 1.

The base-clusters will be further processed to form the final clusters. That is, they are merged on grounds of being similar. Specifically, two base-clusters $\sigma : S$ and $\sigma' : S'$ are similar if and only if $\frac{|S \cap S'|}{|S|} > 0.5$ and $\frac{|S \cap S'|}{|S'|} > 0.5$.

Now consider the *similarity-graph* where the base-clusters are nodes and there is an edge between nodes if and only if they are similar. The final clusters correspond to the connected components of the similarity-graph. That is, a cluster is the union of the document-sets found in the base-clusters of a connected component. Originally the cluster is not given a designated label of its own. However, we will find use for such a label so we add one by collecting the words from the base-cluster labels and sort them by their frequencies therein.

Clearly, the construction of final clusters requires every base-cluster to be checked for similarity with every other base-cluster. This is a bottleneck in the process but Zamir and Etzioni circumvent the problem by restricting the merging to just a selection of the base-clusters. For this purpose they introduce a *score* and form the final clusters from from only the 500 highest scoring base-clusters. We refer to this limit as λ, i.e. in [12] we have $\lambda = 500$.

The score of a base-cluster $\sigma : B$ is defined to be the number $|B| \times f(\sigma)$ where the function f returns the *effective length* of σ. The effective length of a phrase is the number of words it contains that are neither too frequent, too rare nor appear in a given *stop-list* of irrelevant words. Specifically, 'too frequent' means appearing in more than a percentage $\theta_{\max}$ of the (total collection of) documents whereas a word is too rare if it appears in less than $\theta_{\min}$ documents. Originally these thresholds are set to $\theta_{\min}$ =3 and $\theta_{\max}$ =0.4. Furthermore, f penalizes single-word phrases, is linear for phrases that are two to six words long and becomes constant for longer phrases. See [12] and [5].

3 Modifications

In the present context we can make use of the front page matter, i.e. headline, ingress and caption, should there be a photo. So, for each news article its snippet will be the collection of such phrases. Given a snippet containing multiple phrases, each of them will be inserted into the trie separately, i.e. a suffix-tree would hold all the suffixes of each phrase in the snippet.

Our first modification is to abandon the confinement to suffixes and instead fill the compact trie with all n-grams of snippets for a suitable n. One reason is that the use of suffixes shifts emphasis towards the end of the phrase in the sense that a word will appear more times in the suffix-tree than the words that precede it. For example, the suffixes of the phrase 'one two three four' are 'one two three four', 'two three four', 'three four' and 'four' so the suffix-tree contains four occurrences of the word 'four' and three of 'three' whereas 'two' occurs twice and 'one' only once. In contrast, the corresponding 2-grams are 'one two', 'two three' and 'three four'. Now the words 'one' and 'four' appear once while 'two' and 'three' appears twice. Generally, the words on the rims of the phrase will have some fewer occurrences in the trie, but the heavy bias toward the end is gone. Furthermore, except from some uninteresting special cases, the number of

words contained in the n-grams of a phrase is strictly fewer than the number of words in the corresponding suffixes. With fewer words to process we expect the algorithm to work faster but there is also the concern that the information held by the data then becomes impoverished. In an attempt to strike a balance, we choose n so as to maximize the number of words inserted into the trie. This is achieved by expanding a phrase of length k into its $\lceil k/2 \rceil$-grams.

Secondly, we disregard clusters where the label consists of a single word only. The presence of very frequent words may cause texts to gravitate towards each other when clusters are formed. Even if the use of a stop-list can help reduce the impact of some very common and irrelevant words we can not blacklist every common word there is. There will inevitably be clusters cemented by the co-occurrences of a single common word. Such clusters are often large and inaccurate. A one-word cluster is not necessarily a bad cluster but it seems reasonable to assume that this is the case more often than not. Then the net effect of removing them would be positive.

Thirdly, we believe the original scoring is somewhat arbitrary and sensitive to the kind of text it is applied to. In the case at hand there is a high proportion of articles that should make up a cluster of its own, being the only texts on their topics. This is a natural characteristic for a corpus such as ours because of petty local news that are only reported once and do not spread nationwide. Unfortunately, the original scoring favors bigger base-clusters and so singleton clusters are never passed on to further processing. Experimentation with different scoring-functions revealed a significant potential for improvement relative to our specific data. Here we use the scoring obtained from the original one by tweaking the θ_{min} and θ_{max} thresholds, to 6 and 0.5 respectively, and reversing the order.

Finally, we will apply a more sophisticated similarity measure. Originally, the similarity of two base-clusters is determined solely on the basis of the amount of overlap between the document-sets they are composed of. It seems likely that the decision would benefit from taking into account additional cluster characteristics such as word frequencies and label overlap.

Notation: We write $\hat{\sigma}$ to denote the set of words occurring in a label σ.

We define a new similarity measure by making the additional requirement that the labels should have a certain amount $\theta_{\cap}$ of overlap and that the average frequency of the words they contain is below a certain limit θ_{freq}. Specifically, base-clusters $\sigma : S$ and $\sigma' : S'$ are similar iff they satisfy the original measure in conjunction with

$$|\hat{\sigma} \cap \hat{\sigma}'| \geq \theta_{\cap} \quad \text{and} \quad \frac{\sum_{w \in \hat{\sigma} \cup \hat{\sigma}'} cf(w)}{|\hat{\sigma} \cup \hat{\sigma}'|} \leq \theta_{\text{freq}}$$

where $cf(w)$ denotes the *corpus frequency* of the word w, i.e. the total number of times w occurs in our documents. In our experiments we set $\theta_{\cap}$ =2 and $\theta_{\text{freq}} = 4$.

4 Ground Truth, Precision and Recall

The manually tagged portion of our corpus can serve as ground truth for evaluation in terms of precision and recall. Since the tags represent a human judgement as to what the

document is about we think it is fair to assume that a high degree of overlap in tags will indicate overlap in content.

A *ground truth cluster* consists of all documents having identical tags, and only those documents. Thus, ground truth clusters are identifiable by tags.

Precision/recall studies rely on a notion of *relevance*. Here, the basic idea is that a good cluster contains only documents with the same tags. That is, a cluster is considered relevant if it matches a subset of some ground truth cluster. The question is, should we require a *perfect* match?

Consider a cluster containing three articles, two of which are tagged

$$International - Politics - Climate - Obama - Copenhagen$$

and one with the tag

$$International - Politics - Climate - Draft - Copenhagen$$

These articles are all about the 2009 Copenhagen Climate Change Conference, and the cluster would appear to be good. However, there can be no matching ground truth cluster because of the discrepancy of one word in the tags. Is it reasonable to deem this cluster irrelevant? This is largely a matter of the intended use and human opinion so the question has no definite answer. However, by incorporating a degree of perfection we get the flexibility that might allow for the cluster to be considered relevant. A cluster C *matches ground truth with discrepancy* $5-d$ if and only if d is the number of categories common to all tags in C. Intuitively, discrepancy 0 means a perfect match, i.e. $C \subseteq G$ for some ground truth cluster G, while 5 means that there is no category that appears in all tags and we can hardly claim a match at all.

Assuming that C is the set of clusters generated by the algorithm and R the set of relevant clusters, precision would measure the proportion $\frac{|C \cap R|}{|C|}$ of relevant clusters among the clusters generated by the algorithm. Recall measures the extent to which the algorithm will recreate the set of relevant clusters, i.e. $\frac{|C \cap R|}{|R|}$.

The computation of recall-values presents us with a serious challenge. Clearly, checking to what extent the relevant clusters has been generated involves checking the $2^{|G|}$ subsets of each ground truth cluster G. If ground truth clusters are large this job becomes too massive. We escape the problem by introducing a limit λ_{rec} on the size of relevant clusters to be considered. That is, if $|G|$ exceeds this limit a random selection G' of size λ_{rec} is extracted from G and only the subsets of G' are considered for recall. Inspection of our sample corpus reveals that only 3 ground clusters have more than 9 elements. We set $\lambda_{\text{rec}} = 9$ for our experiments.

As described above, the algorithm sets the limit λ on the number of base-clusters that proceed to be merged into final clusters. This poses a serious threat to recall. Specifically, only 500 of the original 25,378 base-clusters are retained. When the initial data for generating clusters is cut short by such a large amount we can not expect the algorithm to be able to fully recreate the ground truth. In fact, our data contains more than 4,500 relevant clusters, while Zamir and Etzioni's original setup of the algorithm produces a total of only 372 clusters. With our corpus this has a devastating effect, causing great harm to recall. For these reasons we prioritized precision over recall when modifying the algorithm.

5 Evaluation

Tests have been carried out to evaluate our modifications of the algorithm. Our benchmark is the performances of the original algorithm shown in Table 1.

Table 1. Original algorithm with $\lambda = 500$.

Discrepancy	Precision	Recall
$d = 0$	0.726	0.053
$d \leq 1$	0.742	0.114
$d \leq 2$	0.753	0.252

Initial experiments have shown that each of our three modification will improve performance. Together they make a considerable difference. Table 2 a) shows precision and recall for the modified algorithm, as well as the average change in performance compared to the benchmark.

Table 2. Modified algorithm

	a) $\lambda = 500$		b)$\lambda = 25,000$	
Discrepancy	Precision	Recall	Precision	Recall
$d = 0$	0.984	0.052	0.954	0.200
$d \leq 1$	0.984	0.052	0.965	0.510
$d \leq 2$	0.984	0.052	0.969	0.608
Average change	+33%	−63%	+30%	+217%

As expected the modified algorithm is more efficient, running 24% faster than the benchmark run.

We have already noted that recall suffers as a result of discarding base-clusters. Indeed, increasing the mass by merging more base-clusters will boost recall. This can be observed in Table 2 b) but, unfortunately, it comes with a punishing cost in running-time. Clearly, the higher number of base-clusters floods the bottleneck of merging them.

6 Conclusion

We have modified the Suffix Tree Clustering technique with some success. Because of our focus on a particular data set we can not claim external validity for our results. Beyond the adaptation for the Norwegian Newspaper Corpus, our contribution is merely to demonstrate an interesting potential for improvement and also to point out directions for further work.

There are several issues that could be pursued. First, we believe that there are other varieties of similarity measures that deserve to be explored. Secondly, we observed that scoring can be sensitive to the kind of text it is applied to. We believe scoring could

make more sophisticated use of cluster characteristics, such as labels, size and word-frequencies. Finally, we see that the potential for good recall values is severely hampered by the computational bottleneck of merging base-clusters. A research challenge lies in finding faster algorithms or alternative ways of forming clusters from base-clusters.

References

1. S. Allern. Newsvalue: On marketing and journalism in ten norwegian newspapers (in Norwegian). IJ Forlaget (publisher) (2001)
2. S. M. zu Eissen, B. Stein, M. Potthast. The Suffix Tree Document Model Revisited. In: Tochtermann, Maurer (Eds.): Proceedings of the I-KNOW '05, Graz 5th International Conference on Knowledge Management, Journal of Universal Computer Science, pp. 596–603 (2005)
3. D. Elgesem, H. Moe, H. Sjøvaag, E. Stavelin. The national public service broadcaster's (NRK) news on the internet in 2009 (in Norwegian). Report to the Norwegian Media Authority, Department of information science and media studies, University of Bergen (2010)
4. J. Erdal. Where does the news come from? On the flow of news between newspapers, broadcasters and the internet (in Norwegian). Official Norwegian Reports NOU2010:14, appendix 1. (2010)
5. J. A. Gulla, H. O.Borch, J. E. Ingvaldsen. Contextualized Clustering in Exploratory Web Search. In: H. A. do Prado; E. Ferneda. Emerging Technologies of Text Mining: Techniques and Applications, pp. 184–207. IGI Global (2007)
6. G. Losnegaard. Automatic extraction of news text from online newspapers. Project report, Department of information science and media studies, University of Bergen (2012)
7. R. Moe. Improvements to Suffix Tree Clustering. In: M. de Rijke et al. (Eds.): Advances in Information Retrieval, Proceedings of ECIR 2014. LNCS 8416, pp. 662–667 (2014)
8. R. Moe, D. Elgesem. Compact trie clustering for overlap detection in news. In: Proceedings of the Norwegian Informatics Conference (NIK'13) (2013)
9. Norwegian Newspaper Corpus, `http://avis.uib.no/om-aviskorpuset/english`
10. Oslo-Bergen Tagger, `http://tekstlab.uio.no/obt-ny/english/index.html`
11. B. Smyth. Computing Patterns in Strings. Addison Wesley (2003)
12. O. Zamir, 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, pp. 46–54. ACM New York (1998)