Text Cluster Trimming
for Better Descriptions and Improved Quality

Magnus Rosell

KTH CSC
100 44 Stockholm
Sweden
rosell@csc.kth.se

Abstract

Text clustering is potentially very useful for exploration of text sets that are too large to study manually. The success of such a tool depends on whether the results can be explained to the user. An automatically extracted cluster description usually consists of a few words that are deemed representative for the cluster. It is preferably short in order to be easily grasped. However, text cluster content is often diverse. We introduce a trimming method that removes texts that do not contain any, or a few of the words in the cluster description. The result is clusters that match their descriptions better. In experiments on two quite different text sets we obtain significant improvements in both internal and external clustering quality for the trimmed clustering compared to the original. The trimming thus has two positive effects: it forces the clusters to agree with their descriptions (resulting in better descriptions) and improves the quality of the trimmed clusters.

1. Introduction

Text clustering can be used to find groups (clusters) of related texts in a larger set. A good clustering of a text set can serve as an overview, reflecting the actual content, rather than forcing it into predefined categories. This could be usefull in many circumstances: the result from a search engine (Zamir et al., 1997), or as a tool for exploring the contents of any text set (Cutting et al., 1992), a scientific database (Janssens et al., 2007), or a free text question in a questionnaire (Rosell and Velupillai, 2008).

Search engines that cluster the retrieved texts[1] present many clusters with few texts each. This can be useful for finding a particular piece of information. However, we are interested in the use of clustering as a proper exploration tool that also helps a user to expose broad themes of an (unknown) text set. Broad themes imply larger clusters, the contents of which have to be easily accessible to the user.

The text set exploration system Scatter/Gather (Cutting et al., 1992) presents clustering results in a straight-forward manner. It displays a *cluster digest* for each cluster, which consists of: *topical words*, usually words with high weight in the cluster centroid[2], and *typical titles*, the texts that are most similar to the centroid. We believe both parts are important. However, in this paper we focus on the topical words, which we will call a *cluster description*, or just a description, for short. We prefer cluster *description* over *label*. The latter may be confused with the cluster *name*, which could be any arbitrary string (like "Cluster 1" for instance)[3].

In the Scatter/Gather approach the user can re-cluster the entire text set or any of the presented clusters to search for new perspectives and focus on interesting trends. We believe this interaction is very important to exploit the potential of text clustering; the system can provide a result, but a human has to come to an understanding of it. In order for this interaction to be tolerable the system has to be fast and provide useful information.

Text clusters do not typically adhere to any well-known categories. Further, they are often diverse from a human perspective as there usually are many possible ways to divide a text set into content groups. Hence, it is often hard to capture the content of a cluster with a cluster description consisting of just a few words. This may well lead to *credibility* problems (Fogg and Tseng, 1999) if the user thinks that the description does not fit the content of the cluster.

We believe this difficulty in describing the cluster contents to a user is the main reason for the limited use of text clustering, compared to its potential as an automatic overview generator. *We present a trimming method that removes those texts from a cluster that do not fit its description.* Hence, the new trimmed cluster has a better cluster description than the original.

Our experiments show that the trimmed clusters have higher quality, as measured using both internal and external quality measures. For a text set exploration scenario the result of our method is very appealing: clusters with better descriptions and of higher quality.

The rest of the paper is organized as follows. In the next Section we discuss some previous related work. Section 3. discusses the method in more detail, while Section 4. describes the experiments we have conducted. Finally, in Section 5. we outline possible future work and draw some conclusions.

2. Previous Work

There is a review of clustering methods by Jain et al. (1999). Text clustering is covered in many Information Retrieval books (Frakes and Baeza-Yates, 1992; Manning et al., 2008). There is not very much work done on cluster descriptions and we have found nothing on removing texts from clusters based on descriptions.

[1] Such as `clusty.com/` and `www.iboogie.com/`

[2] The center representation of a cluster.

[3] Also *labeling* could mean to assign texts to clusters, i.e. give each text the label of belonging to a certain cluster.

Clustering algorithms are sensitive to outliers. In Garcia-Escudero et al. (2008) two model-based approaches for handling outliers are identified: *mixture modeling* approaches model the outliers with additional components, while *trimming* approaches attempt to disregard them when forming the clusters. In our trimming method, see Section 3., we remove texts after the clustering.

Frequent Term-Based Text Clustering (Beil et al., 2002) constructs text clusters by considering sets of frequent terms/words. However, text clusters may be better described by less frequent words.

In Suffix Tree Clustering (Zamir et al., 1997), cluster descriptions are constructed as a part of the process. Word-n-grams with information on which texts they belong to are put into a trie. The nodes of the trie represent possible text clusters that share a part of such a phrase. That part serves as a cluster description.

As noted by Dhillon (2001) and several others a text clustering has a dual word clustering – for each text cluster a corresponding word cluster with the highest weighted words in the text cluster. The word clusters could be considered extensive text cluster descriptions for their respective text clusters. In Dhillon (2001) the text and word clusters are constructed simultaneously.

There is an interesting discussion of description extraction by Mei et al. (2007). They extract several phrases from the texts under consideration and use these as possible descriptions. To choose among the phrases they compare them to a topic model (which could be a cluster centroid) by means of a semantic score. However, they do not remove texts from the set to make it fit the descriptions better.

Several papers (Mei et al., 2007; Popescul and Ungar, 2000; Kulkarni and Pedersen, 2005) distinguish between descriptions that are *descriptive/representative*, describe the actual content of the cluster, and *discriminating/specific* for the clusters in some way, separating a cluster from the other clusters. Both types of descriptions could help in the understanding of the cluster content. Indeed, descriptions that both reveal the content of the cluster and how it differs from the other clusters, would be preferred. Proposed methods consider only representative descriptions that are also specific, and descriptions that get a high product of some measures of how representative and specific they are.

Treeratpituk and Callan (2006) train a score function for cluster labels on manually labeled data using a measure of overlap between extracted phrases and correct class descriptions. Hence their method requires data that has good (and long) descriptions for a set of categories (which is rather rare to come by) and it will be tuned to that particular set. We would like text cluster description extraction to be more flexible.

3. Trim Clusters to Fit Descriptions

We observe that it is hard to capture the contents of an entire cluster with a short description. Although it might summarize a part of the cluster there are often many texts that treat other topics. Our method forces each cluster to agree with its description by removing these texts. It consists of two steps:

1. Extract cluster descriptions.
2. Trim clusters using the descriptions.

Any method for description extraction could be used in the first step, see Section 3.1. We only investigate descriptions containing single words, i.e. no phrases.

In the second step we remove texts that do not fit the cluster descriptions, see Section 3.2. Our trimming method works as a post-processing step to the clustering algorithm and aims at descriptions that cover the content of their clusters.

3.1. Cluster Descriptions

There are many possible ways in which to construct a cluster description. A perfect system would read all texts and *generate* a suitable description, perhaps including words that do not appear in the texts. We investigate only single word *extraction* methods.

To create a cluster description we assign a score to words that appear in the cluster and present them as a list ordered accordingly. In this work we try four simple methods. Let w be a word, c be a cluster, and T be the whole text set. Each word gets as its score:

- $F(w|c)$, its frequency in the cluster.
- $FaP(w|c,T) = F(w|c)\frac{F(w|c)}{F(w|T)}$, where the first factor is the frequency and the second measures how discriminating/specific the word is for the cluster compared to the whole text set. See for instance Popescul and Ungar (2000).
- $Ce(w|c)$, its weight in the cluster centroid, see e.g. Cutting et al. (1992).
- $CeEnt(w|c)$, $Ce(w|c)$ multiplied by the pseudo-information gain based on $Ce(w|c)$ of the clustering compared to the entire set. $CeEnt(w|c) = Ce(w|c)\big[\log_2(\gamma) + \sum_{ci} p(w,c_i)\log_2(p(w,c_i))\big]$, where γ is the number of clusters, and $p(w,c_i) = Ce(w|c_i)/\sum_{c_i} Ce(w|c_i)$. Information gain is higher for words with a skewed weight distribution over the clusters; words with weight in few clusters get higher scores compared to Ce.

The simplest method is arguably F. It is descriptive/representative. Using only raw frequencies the FaP-method introduces the discriminating/specific aspect.

The centroid contains the average weights for all words in the cluster. The weights could be calculated using a tf*idf scheme (as we do, see Section 4.1.). Compared to the F method the Ce method thus uses more information. As the FaP method the $CeEnt$ method can be said to be discriminating/specific. However, it compares the distribution of the words in the cluster not to the entire text set, but to the distribution over the clusters.

The ordered lists of words are usually very long. We choose the x words with highest score as our description. We do not address how to choose the best number of words for a description. This will depend on many factors[4] and

[4] Among other things the text set, number of clusters, and the purpose of the clustering. It is also quite possible that different

might be best left to a user of a system in a particular circumstance. We believe interaction is the key to a useful clustering system. However, we investigate the effect of the number of words in the description.

3.2. Cluster Trimming

We want to remove texts that are not relevant to the description. For descriptions consisting only of single words a simple method is to remove all texts not including one, a few, or all of them.

A slightly more sophisticated version is to use the words with their scores as a “centroid” and only keep texts with a similarity[5] to it greater than a predefined threshold. It has the added benefit of generating an ordering of the texts, which is crucial for choosing the most representative texts, the other half/part of the cluster digest. We use this *description centroid trimming method* with the threshold set to zero. It could be interesting to vary the threshold, but we do not do that. The method is based on the clustering and such a parameter would make it less transparent for a user. However, in our experiments we do vary the number of words in the descriptions and the number of words from the description that must be in each text.

The trimming is similar to a search in a search engine. Texts that are similar/relevant to the description are retrieved from the text cluster and saved, while the others are disregarded. The trimmed clusters are thus more coherent – all texts are related to the cluster description[6]. If the words in a description are not treating the same subject the trimmed cluster will still be diverse, but at least it will be possible to to recognize the different subjects by studying the description.

Combined with the description extraction methods we have four trimming methods. We use the same symbols as in Section 3.1., i.e. we use Ce to denote the trimming method that utilize descriptions from the description extraction method Ce, and similarly for the other methods.

If all texts are required to be in the clustering there are many ways they could be included. For each original cluster we could make a rest cluster that can be presented together with the trimmed cluster. Alternatively, all removed texts could be clustered as a separate text set, and the result presented together with the trimmed clusters. We do not investigate any such inclusion of removed texts.

4. Experiments

We have conducted experiments on two text sets described in Section 4.1. with the K-Means clustering algorithm as described in Section 4.2. and the description extraction and cluster trimming methods described in the previous sections. In Section 4.3. we describe how we evaluate the results and in Section 4.4. we present the results and discuss them.

	Texts	Cat.	Words	w/t	t/w
20ng	7519	20	7172	71	74
DN	6877	5	6007	58	66

Table 1: Text set statistics. Number of texts, categories, and unique words (stems/lemmas) after preprocessing. Average number of unique words per text and number of texts each unique word appears in. We have used quite aggressive filtering of common words. However, the results in the following sections are similar with no such filtering.

4.1. Text Sets and Representation

In our experiments we have used the following two rather different text sets:

20ng A part of the *20 Newsgroups* corpus[7], a collection of newsgroup documents in English (Lang, 1995).

DN A collection of newspaper articles from the Swedish newspaper *Dagens Nyheter*[8], from the larger collection *KTH News Corpus* (Hassel, 2001). These are categorized into the sections of the paper: Culture, Economy, Domestic, Foreign, and Sports.

We have removed stopwords, infrequent words, and information about the categories from both collections. Further, we have applied stemming (an implementation of the Porter stemmer (Porter, 1980)) on the 20ng text set. For DN we have lemmatized the words using the Granska Text Analyzer (Knutsson et al., 2003), and split compounds using the spell checking program Stava (Kann et al., 2001), since this kind of preprocessing improves clustering results for Swedish (Rosell, 2003). Table 1 contains some statistics for the text sets after this preprocessing.

The text sets are represented in the common vector space model of Information Retrieval. We construct a word-by-text-matrix with weights using a tf*idf-weighting scheme, and normalize the text vectors. For similarity between two texts, $sim(t_1, t_2)$, we use the dot product, which, as the texts are normalized, coincides with the common cosine measure.

4.2. Clustering Algorithm

We prefer fast clustering algorithms, as they lend themselves to interactive use. We have used the well-known *K-Means* algorithm[9], see for instance Manning et al. (2008), that represents each cluster by its centroid. It iteratively assigns all texts to their closest cluster and recomputes the centroids until no text changes cluster. We have, however, set a maximum of 20 iterations.

clusters in a clustering might benefit from having different numbers of words in their descriptions. We do not investigate this issue at all.

[5] We use the cosine measure, see Section 4.1.

[6] Related terms (synonyms, etc.) is always a problem in methods based on the term-document-matrix. Also, in the best case scenario the different meanings of homographs are captured by different clusters.

[7] From `people.csail.mit.edu/jrennie/20Newsgroups/`

[8] `www.dn.se/`

[9] We use the Infomat package, available at `www.csc.kth.se/tcs/projects/infomat/infomat/`

4.3. Evaluation

Clustering evaluation is hard. See Halkidi et al. (2001) for a discussion of several techniques. Evaluated methods should be analyzed with respect to the used quality measures to avoid pitfalls. We want to compare the original *exhaustive* clustering, containing all texts, to the trimmed *non-exhaustive* clusterings.

We evaluate the results by both an internal and an external quality measure to get a balanced view of the impact of our trimming methods. Internal quality measures use no external knowledge, but are based on what was available for the clustering algorithm. The internal *self similarity* uses the similarity measure to assess the clustering quality, see Section 4.3.1.

One type of external quality measures compare the clustering to another partition, such as a manual categorization. We use the *mutual information*, see Section 4.3.2. While the self similarity depends on the similarity definition, the mutual infomation depends on the quality of the categorization.

In the results of Section 4.4. we also give the number of texts (Texts) for all clusterings, as the methods presented here result in clusterings with different numbers of texts. Thus we need to know that the measures we use are not affected by this, or if they are we should be aware of how when interpreting the results. To this end we analyze the measures in Sections 4.3.1. and 4.3.2., and introduce two reference methods in Section 4.3.3.

As the K-Means algorithm is not deterministic we run it several times and calculate average values. As a rule of thumb we do not consider two results different if their standard deviations overlap.

4.3.1. Self Similarity

The centroid of each cluster is the average weight vector of the weight vectors of its texts. The *self similarity* of a cluster c_i, $sim(c_i, c_i)$, is the average similarity between all texts in the cluster. As we do not normalize centroids, it equals the dot product of the cluster centroid with itself. We define the average self similarity of the entire clustering:

$$\Phi(C) \quad = \quad \frac{1}{|C|} \sum_{c_i \in C} |c_i| \cdot sim(c_i, c_i).$$

Note that we do not change the representation in any method, so the similarities are comparable.

We use the average similarity so, in principle, the results should be comparable between clusterings of different numbers of texts. However, the similarities of texts with themselves (which is one) are included in the definition of $sim(c_i, c_i)$. This leads to higher values for smaller clusters, but the effect is negligible until the clusters are very small. We monitor it using the reference method RSz that we introduce in Section 4.3.3.

Internal evaluation using Φ assess clusterings based on the text similarity definition and how the texts are distributed over the clusters. As the K-Means algorithm uses the same information the evaluation is in some respects questionable.

The trimming methods start from the clustering, so they also use this information. They continue by removing texts that are not similar to the description, which is not a text. Here they do not consider the whole centroid (and hence all texts) as K-Means. The evaluation therefore can be said to be unfair to the trimming methods.

4.3.2. Mutual Information

Consider a categorization K with κ categories of the same text set as the clustering C with γ clusters. The elements $m_i^{(j)}$ of a confusion matrix M count the number of texts that belong to cluster c_i and category $k^{(j)}$. The probability that a text picked at random belongs to cluster c_i and category $k^{(j)}$ is: $p_i^{(j)} = m_i^{(j)}/|C|$. The mutual information (see for instance Manning and Schütze (1999)) compares a clustering to a categorization:

$$MI(C, K) \quad = \quad \sum_i \sum_j p_i^{(i)} \log_2 (\frac{p_i^{(j)}}{p_i p^{(j)}}),$$

where $p_i = |c_i|/|C|$ and $p^{(j)} = |k^{(j)}|/|C|$.

For each trimmed clustering we construct a corresponding categorization by removing the same texts from the original categorization. This might lead to an “easier” categorization, which is desirable – if a trimming method makes it easier to follow the categorization it is successful. Successful in the external evaluation sense: the clustering divides the texts into groups that are similar to the categories. If we believe that the categorization is adequate, a clustering (a trimmed or an original) with a high mutual information indicates clusters of low diversity, that will be easier for a user to grasp.

The mutual information is only based on relative frequencies and thus is *not dependent on the number of texts in the clustering*. However, during the trimming and the corresponding reduction of the categorization entire categories may be removed[10]. The normalized mutual information (NMI) takes the distribution over the clusters and categories into account and makes it theoretically possible to compare results of clusterings and categorizations with different number of clusters and categories (Strehl and Ghosh, 2003):

$$NMI(C, K) \quad = \quad \frac{MI(C, K)}{\sqrt{H(C)H(K)}},$$

where $H(C) = -\sum_i p_i \log_2 p_i$ and $H(K) = -\sum_j p^{(j)} \log_2 p^{(j)}$.

4.3.3. Reference Methods

To get further understanding of the performance of the presented methods we introduce two reference methods. For every trimmed clustering $trim$ we also evaluate the baseline $RSz(trim)$, where each cluster of the original clustering C is reduced randomly to the same size as for $trim$. In theory NMI should not favor small clusters, and when the clusters are large enough Φ does not either. If $RSz(trim)$ perform equally to C it is also true for our experiments.

[10] This is very rare. Clusters always keep at least some texts with all the methods as the words used for the trimming appear in them. Entire categories are removed very seldom and only when the trimming is extreme.

Method	Text Set 20ng			Text Set DN		
	NMI	Φ	Texts	NMI	Φ	Texts
Clustering	0.57 (0.02)	0.037 (0.000)	7519 (0)	0.53 (0.02)	0.048 (0.001)	6877 (0)
Ce	0.69 (0.04)	0.061 (0.002)	3211 (128)	0.61 (0.02)	0.071 (0.003)	3693 (109)
$Sz(Ce)$	0.69 (0.04)	0.067 (0.002)	3211 (128)	0.66 (0.02)	0.077 (0.002)	3693 (109)
$RSz(Ce)$	0.57 (0.03)	0.039 (0.000)	3211 (128)	0.56 (0.02)	0.051 (0.001)	3693 (109)
$CeEnt$	0.76 (0.05)	0.072 (0.003)	2263 (144)	0.67 (0.03)	0.090 (0.004)	2652 (146)
$Sz(CeEnt)$	0.71 (0.05)	0.078 (0.003)	2263 (144)	0.70 (0.03)	0.095 (0.003)	2652 (146)
$RSz(CeEnt)$	0.58 (0.03)	0.041 (0.001)	2263 (144)	0.58 (0.03)	0.053 (0.002)	2652 (146)
F	0.67 (0.04)	0.057 (0.002)	3591 (136)	0.59 (0.02)	0.068 (0.003)	3781 (86)
$Sz(F)$	0.68 (0.05)	0.063 (0.002)	3591 (136)	0.65 (0.02)	0.076 (0.002)	3781 (86)
$RSz(F)$	0.57 (0.03)	0.039 (0.001)	3591 (136)	0.55 (0.02)	0.051 (0.001)	3781 (86)
FaP	0.72 (0.06)	0.067 (0.004)	2692 (226)	0.63 (0.03)	0.076 (0.003)	3304 (157)
$Sz(FaP)$	0.70 (0.05)	0.073 (0.004)	2692 (226)	0.67 (0.03)	0.082 (0.002)	3304 (157)
$RSz(FaP)$	0.57 (0.03)	0.040 (0.001)	2692 (226)	0.57 (0.03)	0.052 (0.001)	3304 (157)

Table 2: Original clustering results and trimming results for the four different methods using five word descriptions, where at least two of these need to be in each text. For each method also the corresponding reference methods Sz(.) and RSz(.). The normalized mutual information (NMI), the self similarity (Φ), and the number of texts. Average results for 20 clusterings to 10 clusters each, of text sets 20ng and DN, standard deviations within parenthesis. For a result to be considered better than an other the standard deviations, as a rule of thumb, must not overlap.

Our second reference method, $Sz(trim)$, reduces the original clustering to the same size as $trim$ by removing the texts in each cluster that have the lowest similarity to the centroid. Normally this leads to improved internal quality. To compare the trimming methods to Sz using the average self similarity Φ is unfair, as the latter uses the full centroid (and hence all texts), while the trimming methods use the much shorter descriptions.

For a method $trim$ to be considered valuable it has to outperform $RSz(trim)$. Otherwise we can just remove any texts and get better results. If it outperforms $Sz(trim)$ it is definitely useful, if it does not it can still be valuable. Without the trimmingmethod we would not know how many texts to remove with $Sz(trim)$. Further, we would not have the advantage of the descriptions that fit the clusters.

4.4. Results and Discussion

We summarize the results of our experiments here, focusing on the quality of the trimmed clusterings. This is, however, as we already have stressed, only one of the benefits of the methods – the other being that the descriptions are more accurate for the trimmed clusters.

4.4.1. Main Results

Table 2 gives an overview of some of the experiments we have performed using descriptions of 5 words, where at least two has to be in each text. It presents average results for 20 clusterings with 10 clusters each, of the 20ng (left part) and the DN (right part) text sets. The four extraction-trimming methods are presented together with their corresponding reference methods. We obtained results similar in tendency for other numbers of clusters and words in the descriptions.

Our main finding is that for all trimming methods there are significant improvements in both self similarity and mutual information compared to the original clustering.

All trimming methods also outperform the random size reduction method RSz for both measures. In fact, as expected the RSz methods perform equally to the original clustering in NMI and just slightly better in Φ. *Hence, we know that improvements achieved through trimming is not explained by the size of the clusters.*

All trimming methods perform comparably in NMI and Φ to the centroid ordered size reduction Sz. This is a good result considering that the number of texts in the trimmed clusters are determined automatically. As mentioned in Section 4.3.3., the Sz method is dependent on the trimming methods to know how many texts to remove. It is remarkable that removing texts based on similarity to the short description leads to as good results as when based on the entire cluster centroid (i.e. Sz)[11].

4.4.2. Number of Words in the Descriptions

In Figure 1 we give the results for trimming with different numbers of words in the descriptions for the 20ng text set. The results for the DN text set are similar in tendency.

All values are still the average for 20 clusterings to ten clusters, but the standard deviation is only presented for the original clustering (dotted lines in the plots). It is always in the same order of magnitude for the trimmed clusterings.

The left side of Figure 1 (plots a through c) shows the results for trimming where each text has to contain at least one of the words in the description. The right side (plots d through f) shows results where each text has to contain at least five of the words in the description, or as many as possible for descriptions with fewer words. For descriptions with one to five words plot f shows a very steep reduction of the number of words (as should be expected).

The results in quality (NMI and Φ) seems to be explained entirely by the number of texts in the results; the

[11] It can be compared to the good clustering results when using truncated centroids to represent clusters in Schütze and Silverstein (1997). However, they keep significantly more words.

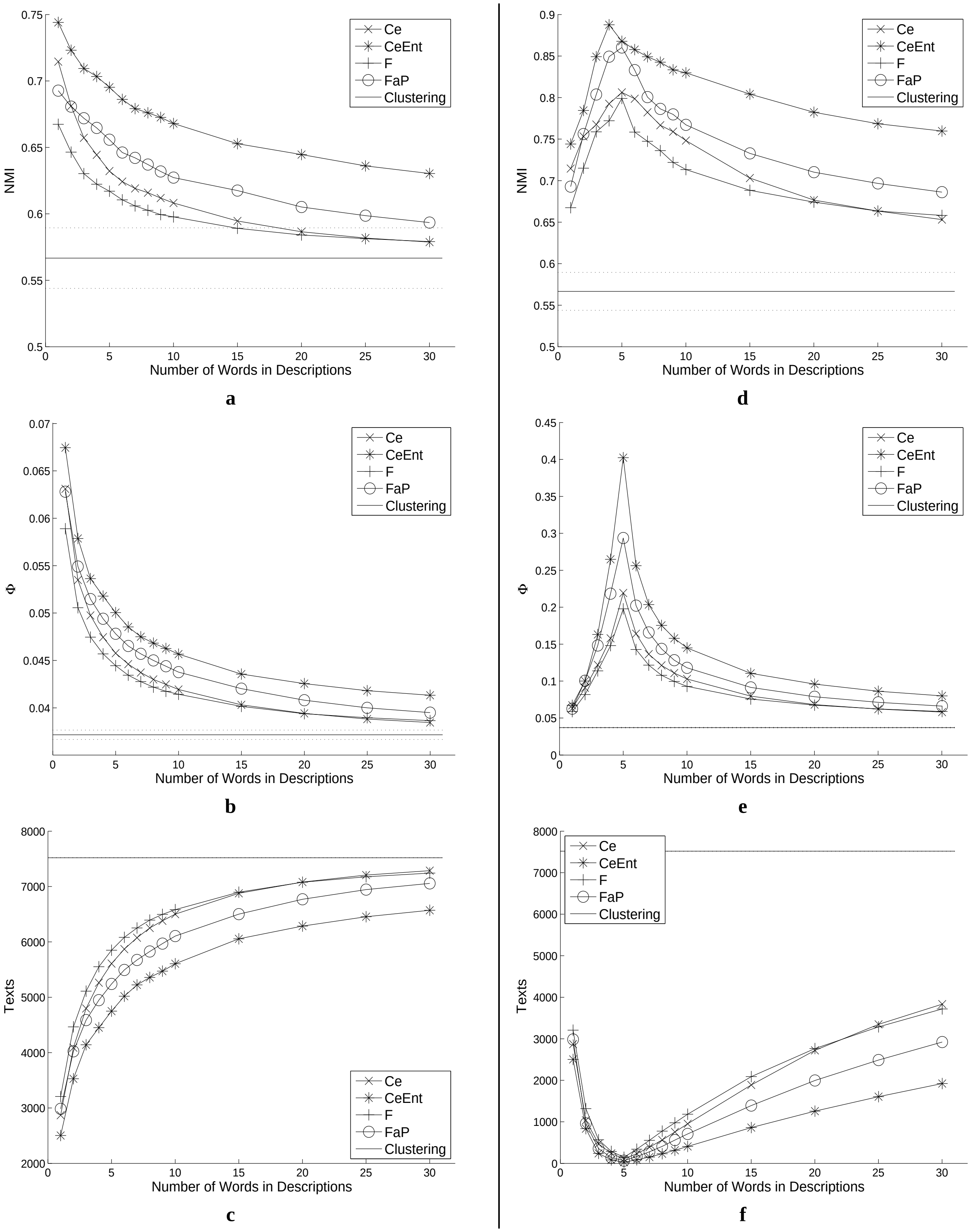

Figure 1: Average results for different number of words in the descriptions for text set 20ng. Each result is from 20 clusterings and the dotted lines indicate standard deviation for the original clustering. The normalized mutual information (NMI), the self similarity (Φ), and the number of texts. Left column (a-c) texts with at least one word in the description. Right column (d-f) texts with at least five words in the descriptions (except for descriptions of 1-5 words, where all words in the descriptions has to be in each text). For the corresponding RSz methods see Figure 2.

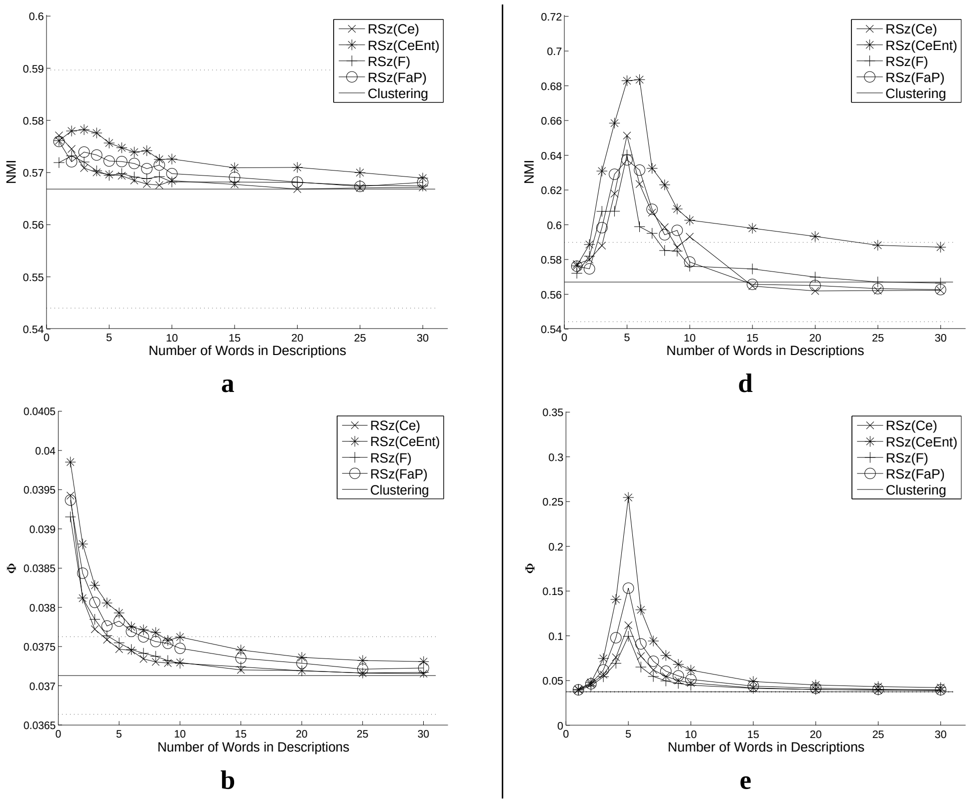

Figure 2: Results for the reference method RSz for the same setting as in Figure 1. The number of texts are the same. The measures are only affected when the trimming is extreme (note the huge difference in scale in the left column). Hence the positive results in Figure 1 can not be explained simply by the reduction of the number of texts in the trimmed clusterings.

fewer texts, the higher quality. However, the number of texts that are retained is decided by the trimming method.

The effects of the trimming methods are quite obvious up to 30 word descriptions, but such long descriptions may not be practical. The balance between the number of texts and quality is probably differing between applications. The number of words in the descriptions should primarily be chosen for the convenience of the users. Looking at these results there is no other reason to have longer descriptions than that shorter ones lead to fewer texts. However, when the trimming is extreme the evaluation measures can be affected.

Figure 2 shows the results for the corresponding RSz methods. The number of texts are the same as in Figure 1. It is only when the trimming is extreme (very few texts left) that the results of the trimming methods can be explained in part by the fewer texts. Otherwise the trimming methods perform significantly better than the RSz methods. We do not present the results for the Sz methods here. They perform similar to the trimming methods.

In almost all our results the methods have the same order quality wise, although they mostly overlap in standard deviations. The methods that extract words that are both representative and discriminating (i.e. CeEnt and FaP) retain less texts but perform better than the methods that only considers representative words. Using only the discriminating factors (the pseudo information gain and the second factor of FaP) leads to too specific words and almost no texts in the resulting clusters.

5. Conclusions and Future Work

We have presented methods for trimming text clusters to fit their descriptions. Our evaluation gave results that were similar in tendency on two different text sets in different languages, using both an internal and an external quality measure.

The result of the trimming methods are clusterings with fewer texts, better descriptions, and of higher quality. We believe that such a trimmed clustering therefore will be advantageous for most applications, but in particular when clustering results are presented to humans in an interactive manner, as in the Scatter/Gather system. Smaller coherent clusters with accurate descriptions are more useful than a clustering covering all texts at the expense of clarity.

A further extension of an interactive clustering tool would be to allow the user to generate different descriptions for a cluster and to remove or add words to these. The system would respond by presenting the texts that fit the new descriptions.

There is much that could be done using the methods presented here and to continue and extend this work. Trimmed clusters are less diverse than the originals, but they can still contain differing themes. It would be interesting to try to build as coherent descriptions as possible using for instance some kind of word relation resource, such as a thesaurus or an automatic method that extracts the relations from text.

By trimming a diverse cluster with different descriptions we may be able to find different interesting themes. This could potentially, in combination with a fast naïve clustering algorithm, produce well described clusters of high quality fast. It might also be important to try to find the most suitable number of words in the description for each cluster.

The combination of a clustering and a trimming method results in a non-exhaustive clustering. If it is important that all texts are in the clustering they are easily included. On the other hand, it would be interesting to compare this method to non-exhaustive clustering algorithms. Another approach could be to remove entire clusters that do not fit their descriptions well enough.

In addition to a description a cluster digest contains the most representative texts of the cluster. We have not investigated the difference in quality of these texts for different trimming methods (and Sz methods).

In summary, the combination of cluster description extraction and cluster trimming has two advantages: it results in clusters with better descriptions *and* of improved quality.

6. References

F. Beil, M. Ester, and X. Xu. 2002. Frequent term-based text clustering. In *KDD '02: Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pages 436–442, New York, NY, USA.

D. R. Cutting, J. O. Pedersen, D. Karger, and J. W. Tukey. 1992. Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proc. 15th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*.

I. S. Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD '01: Proc. 7th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pages 269–274, New York, NY, USA.

B. J. Fogg and H. Tseng. 1999. The elements of computer credibility. In *CHI '99: Proc. of the SIGCHI Conf. on Human factors in computing systems*, pages 80–87, New York, NY, USA.

W. B. Frakes and R. Baeza-Yates. 1992. *Information Retrieval Data Structures & Algorithms*. Prentice Hall.

L. A. Garcia-Escudero, A. Gordaliza, C. Matran, and A. Mayo-Iscar. 2008. A general trimming approach to robust cluster analysis. *Annals Of Statistics*, 36:1324.

M. Halkidi, Y. Batistakis, and M. Vazirgiannis. 2001. On clustering validation techniques. *J. of Intelligent Information Systems*, 17(2-3):107–145.

M. Hassel. 2001. Automatic construction of a Swedish news corpus. In *Proc. 13th Nordic Conf. on Comp. Ling. – NODALIDA '01*.

A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

F. Janssens, W. Glänzel, and B. De Moor. 2007. Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 360–369, New York, NY, USA.

V. Kann, R. Domeij, J. Hollman, and M. Tillenius. 2001. Implementation aspects and applications of a spelling correction algorithm. *L. Uhlirova, G. Wimmer, G. Altmann, R. Koehler (eds.), Text as a Linguistic Paradigm: Levels, Constituents, Constructs. Festschrift in honour of Ludek Hrebicek, vol 60 of Quantitative Linguistics*, pages 108–123.

O. Knutsson, J. Bigert, and V. Kann. 2003. A robust shallow parser for Swedish. In *Proc. 14th Nordic Conf. on Comp. Ling. – NODALIDA '03*.

A. Kulkarni and T. Pedersen. 2005. SenseClusters: unsupervised clustering and labeling of similar contexts. In *ACL '05: Proc. of the ACL 2005 on Interactive poster and demonstration sessions*, pages 105–108, Morristown, NJ, USA.

K. Lang. 1995. Newsweeder: Learning to filter netnews. In *Proc. of the Twelfth Int. Conf. on Machine Learning*, pages 331–339.

C. D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Q. Mei, X. Shen, and C. Zhai. 2007. Automatic labeling of multinomial topic models. In *KDD '07: Proc. of the 13th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pages 490–499, New York, NY, USA.

A. Popescul and L. Ungar. 2000. Automatic labeling of document clusters. Unpublished manuscript, U. Pennsylvania.

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

M. Rosell and S. Velupillai. 2008. Revealing relations between open and closed answers in questionnaires through text clustering evaluation. In *Proc. of the Sixth Int. Conf. on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

M. Rosell. 2003. Improving clustering of Swedish newspaper articles using stemming and compound splitting. In *Proc. 14th Nordic Conf. on Comp. Ling. – NODALIDA '03*.

H. Schütze and C. Silverstein. 1997. Projections for efficient document clustering. In *Proc. 20th Annual Int. ACM SIGIR Conf. on Research and development in information retrieval*, pages 74–81, New York, NY, USA.

A. Strehl and J. Ghosh. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617.

P. Treeratpituk and J. Callan. 2006. Automatically labeling hierarchical clusters. In *Proc. of the 2006 Int. Conf. on Digital government research*, pages 167–176, New York, NY, USA.

O. Zamir, O. Etzioni, O. Madani, and R. M. Karp. 1997. Fast and intuitive clustering of web documents. In *KDD '97: Proc. of the 3rd ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pages 287–290, New Port Beach, CA, USA.