Constrained Text Clustering Using Word Trigrams

M. Eduardo Ares and Álvaro Barreiro

IRLab, Department of Computer Science, University of A Coruña, Spain
{maresb,barreiro}@udc.es

Abstract. In recent years there has emerged the field of Constrained Clustering, which proposes clustering algorithms which are able to accommodate domain information to obtain a better final grouping. This information is usually provided as pairwise constraints, whose acquisition from humans can be costly. In this paper we propose a novel method based on word n-grams to automatically extract positive constraints from text collections. Clustering experiments in text collections composed by different types of documents show that the constraints created with our method attain statistically significant improvements over the results obtained with constraints created using named entities and over the results of a high-performing non-constrained algorithm.

Keywords: constrained clustering, constraint extraction, word n-grams

1 Introduction

The ever-increasing amount of information available to individuals, corporations and governments has prompted a growing need for automated tools in order to explore and process this information, tools which with the popularisation of the Internet have gained an extraordinary importance. Traditionally, the answer given by Data Mining to this situation was divided in two approaches, classification and clustering. In clustering, an example of unsupervised learning, the system tries to detect groups of related elements in the data (the clusters), traditionally with the only help of the information contained in that data.

However, in recent years there has emerged a new fashion of semi-supervised algorithms, coined as *Constrained Clustering*[4]. These algorithms, which at the core still are clustering algorithms (that is, their aim is finding groups in the data), are nevertheless provided with some domain information with the purpose of guiding the process to a more accurate and meaningful (for a human) final grouping. This information is provided in the shape of a set of pairwise relations between data elements, called *constraints*, which express the preference (or even obligation, depending of the constrained clustering algorithm used) of whether the two data instances joined by each of these constraints should or should not be in the same cluster. Respectively, these two kinds or constraints are called positive, or Must-link, constraints, and negative, or Cannot-link, constraints.

Usually, in the experiments reported in the literature of constrained clustering the constraints are obtained with a perfect oracle which knows exactly the grouping used as golden truth, as the focus is on measuring how well the algorithms are able to use the information contained in the constraints. However, in a real world clustering problem obtaining the constraints would be a central issue. The simplest approach is using human experts, which, given two data instances, would decide if they should be placed in the same or different clusters. Even though approaches have been proposed in order to maximise the utility of each judgement [3], this process is likely to remain still expensive, as an important amount of constraints (relative to the size of the data collection) is usually needed to influence the clustering outcome. A less costly alternative is creating automated methods which, perhaps with training and/or some outside domain knowledge, are able to infer constraints from the data.

In this paper we propose a novel way of automatically obtaining positive constraints in textual documents using word n-grams. In the experiments, performed over reference collections, it is shown that the constraints created with this new method are of greater or similar quality to those obtained with a related existing approach which uses more complex and high level information (Named Entity Recognition, an approach suggested by [15]), while conveying mostly different information. As for the final clustering, using this new method obtains statistically significant improvements over the entity-based one. Using the constraints yielded by the presented approach also obtains statistically significant improvements over a high-performing non-constrained clustering algorithm such as Normalised Cut (NC), which is the non constrained version of the Constrained Normalised Cut (CNC) algorithm used with constraints.

The rest of this paper is organised as follows. In Section 2 we make a small introduction to constrained clustering. Afterwards, Section 3 introduces our method, followed by Sections 4 and 5, which explain the experimental setting and present and discuss the results, respectively. Section 6 surveys the related work. Finally, section 7 concludes the paper and gives an outlook of future work.

2 Constrained Clustering

Up to this date the research on Constrained Clustering has been mostly focused on creating new algorithms tailored to different scenarios. In this paper we are focusing on a complementary question, proposing a novel automated way to extract positive constraints from textual data. Given the fully automated nature of the method, it is naturally bound to yield a certain amount of inaccurate constraints, and so a suitable constrained clustering algorithm has to be chosen. Thus, in order to assess the quality of the constraints obtained with this new approach we have opted for using Constrained Normalised Cut (CNC)[11], as it is a high performing algorithm which has been shown to perform remarkably well under noisy conditions[1] and whose non-constrained counterpart, Normalised Cut (NC)[14] provides us with a very exigent regular clustering baseline.

2.1 Normalised Cut and Constrained Normalised Cut

Normalised Cut is an exponent of the spectral clustering family of algorithms. The general objective of clustering algorithms is finding homogeneous groups in the data to cluster, a goal which could be summarised as putting the data in groups such that the data instances assigned to each one of them are similar between them and dissimilar to the ones in the other clusters. Spectral Clustering transforms this problem into a graph cutting problem. In it, data instances are the nodes of a weighted graph where the weight of each edge is related to the similarity between the vertices that it joins. Given this graph, the above-stated clustering objective can be expressed (if larger weights mean greater similarities) as finding a cut of the graph in subgraphs such that the weights of the edges cut are small and the edges between nodes in the same subgraphs have large weights. The advantage of this approach is that it enables us to tackle the clustering task using the extensive literature and results available on graph processing.

In the case of NC[14], Shi and Malik define a certain objective function over the affinity matrix of a graph and a cut of that graph, such that small values of the function would mean a "good" (in the terms stated in the previous paragraph) cut and hence a good grouping of the data. Unfortunately, finding the cut which minimises this function is NP-hard. In order to overcome this problem and make the algorithm computationally affordable an approximate solution is calculated by computing the first k eigenvectors of a Laplacian matrix of the graph (where k is the desired number of clusters), which form a projection of the original datapoints in a reduced k-dimensional space, and clustering these reduced representations with a technique such as k-Means. The contents of the resulting clusters are finally backtracked to the original points, yielding the final outcome of the NC algorithm.

Following the same general structure, Ji and Shu introduce in [11] the Constrained Normalised Cut algorithm, a variant of NC which supports positive constraints. In order to do so, they create a new objective function which measures both the NC objective and the observance of the constraints, which are encoded in the function in such a way that not complying with them would increment its value. Thus, a cut of the graph which is assigned a low value by the function would be at the same time a good clustering and one which substantially respects the constraints. The strength given to the constraints is controlled by a parameter, β, with larger values of β giving more weight to the constraints in the process. As minimising this new function is also NP-hard, the rest of the algorithm follows the same steps as the original NC algorithm (projection in a reduced space and clustering of the reduced representations).

Lastly, it should be noted that some experimental studies[1, 8] have found that in both NC and CNC using more than k eigenvectors in the projection step can improve (sometimes dramatically) the quality of the final clustering (which is still done into k clusters). Consequently, in the experiments reported in this paper we have tuned, apart from β when using Constrained Normalised Cut, the number of eigenvectors used in the projection stages of the two algorithms, a parameter which we have called d.

3 Constraint Extraction Method

In this paper we propose an automatic positive constraint extraction method which is based on the overlap of word n-grams, which are sequences of n contiguous words from a text. Figure 1 shows an example of unigrams, bigrams and trigrams which could be extracted from a sentence.

"As for me, all I know is that I know nothing"	
1-grams	as, for, me, all, i, know, is, that, i, know, nothing
2-grams	as for, for me, me all, all i, i know, know is, is that, that i, i know,...
3-grams	as for me, for me all, me all i, all i know, i know is, know is that,...

Fig. 1. Some possible word n-grams of a sentence

The vast majority of the similarity measures between documents used in text clustering are based on measuring the overlap of their vocabularies, with the intuition that two documents sharing words is a good indicator of their relatedness. Following the same logic, the method proposed in this paper uses the overlap between the word n-grams of documents to detect pairs of documents which are likely to be in the same cluster, producing a Must-link between them if they share a minimum number of n-grams (t). By using word n-grams we are taking advantage of the fact that the words in a text come in a certain natural order, obtaining more information than that obtained by considering each word by itself. For example, let us consider two documents which share a trigram. This means not only that they have (if none of the words is repeated) three words in common, but also that they appear next to each other and in the same order in the two documents, which, for instance, makes more likely that these words are being used with the same sense in both documents, which, in turn, makes more likely that these documents are related and hence belonging in the same cluster.

However, one important aspect to consider is that not all n-grams shared between documents are informative about their relatedness. Namely, it is very likely that a considerable amount of documents share trigrams such as "a lot of", "for instance the" or "in order to", which are expressions which bear little or none information about the subject of a text, and, consequently, the fact of they being shared should not be treated as evidence to create a constraint. In order to reduce this "noise" we have opted for pruning from the set of n-grams obtained from a document all those which contain one or more stopwords. For instance, if we were using trigrams, from the sentence shown in Fig. 1 only "me all i", "all i know" and "i know nothing" would be considered when creating constraints, ignoring others such as "as for me", "for me all" or "is that i". Discarding these trigrams is a conservative solution from the point of view of the constraint creation, limiting quite aggressively their amount in the interests of improving the quality (accurateness) of the resulting set.

In summary, the constraint extraction algorithm proposed in this paper is composed of the next three steps:

I. Extract the n-grams of the documents to cluster
II. Discard the n-grams containing at least one stopword
III. Create a Must-link constraint between all pairs of documents that share at least t n-grams

4 Experimental Design and Methodology

In order to assess the goodness of the constraint extraction method proposed in the previous section we have performed an array of experiments, which were focused on three aspects:

1. Are the constraints generated by this method accurate?
2. Is the information supplied by the constraints extracted with this method different to the one obtained with an existent method?
3. Do these new constraints improve the clustering? If so, how does this improvement compare with the one attained with an existent method?

Obviously, this last question, of whether or not using the new constraints yields a better clustering, is the definitive measure of the quality of the constraints, as that is the final goal of using constrained clustering. However, the first two aspects are also important, as they show the suitability of mining the n-grams in order to obtain new information to feed the clustering algorithms. Also, the comparison of results of the three questions, and specially the discordances that might appear between them, can provide us with some insights about what constitutes a good and effective constraint.

4.1 Datasets and Document Representation

The experiments reported in this paper were conducted over two datasets[1]:

Dataset (i) is a subset of the Reuters RCV1 collection, which compiles a year of stories dispatched by that news agency starting in August of 1996 which were manually categorised according to different criteria: country, industry, and topic. Concretely, this dataset was created by choosing 1000 documents from each of the categories `GDIP` ("International relations"), `GVIO` ("War, Civil war"), `GPOL` ("Domestic politics") and `GCRIM` ("Crime, Law enforcement"), four of the more populated subcategories of `GCAT` ("Government/Social"), a wide reaching top-level category. The documents were randomly chosen from inside each category so as to minimise the chances of picking documents which were too related (corrections, follow-ups, two sides of the same story,...). This yielded a dataset

[1] The exact composition of the datasets can be obtained at: `www.dc.fi.udc.es/~edu/AresBarreiroCERI12.gtruth.tar.gz`

composed by 4000 documents uniformly distributed by topic into four equally-sized clusters.

Dataset (ii) is taken from the 20 Newsgroup collection, a collection of 18828 newsgroups posts. Specifically, we have used the posts of the subset of groups related to religion, that is, the topics `talk.religion.misc`, `alt.atheism` and `soc.religion.christian`. Hence, this dataset is composed by 2424 documents distributed into three clusters, according to the group in which the document was posted.

By using these two datasets we aim to have a wide picture of the performance of the algorithms, since, as it stems from the descriptions, the character of each dataset is quite different: while in the first we will be dealing with neat texts composed by professional journalists, written conforming to a particular style book, in the second the texts were written by regular Internet users in the midst of an on-line discussion, and hence they often show a quite anarchic structure and are affected by typos, anacoluthons,...

In the clustering experiments we have used Mutual Information to represent the documents, as it has been shown to perform better than other $tf \cdot idf$ approaches [13]. Therefore, each document was represented by a vector $[mi_k]_{k=1..m}$, where each mi_k is the pointwise mutual information of the document and each of the m terms in the collection. The similarity between two documents was the cosine distance between their representations.

4.2 Baselines

To have a measure of how our algorithm compares with an existing constraint extraction method we have used the one introduced by Song et al. in [15]. The algorithm, which is based on the detection of named entities, has two steps: in the first, a named entity recognition algorithm is applied to the documents (namely, the Stanford NER, detecting classes "Location", "Person" and "Organization"). In the second stage a positive constraint is created between the documents which share a minimum number of those named entities. The intuition behind this approach is clear: named entities convey a great deal of the meaning of a text, and, consequently, that two documents share a given amount of these entities suggests a relation between the topics covered in them.

We have chosen this constraint extraction method due to two main reasons. First, its structure is similar to the one proposed in this paper (which was independently developed). Secondly, named entities are words by themselves or sequences of consecutive words, as are the n-grams used in our approach. These two circumstances facilitate the comparisons between the two algorithms, and enable us to evaluate the contingent improvement in the results which could be attained by using higher level information, such as the one used to tell named entities apart. This use of high level information (compared to the one used in our approach, i.e. that words appear together and do not include a stopword) by Song et al.'s algorithm ensures that their algorithm is a demanding baseline.

Finally, we have used, as stated in section 2, Normalised Cut as the non-constrained baseline in the clustering experiments. It was chosen due to being

high-performing and also because it is the non-constrained counterpart to the Constrained Normalised Cut algorithm used with constraints.

4.3 Clustering Metrics and Statistical Significance

In the clustering experiments we have used Adjusted Rand Index (ARI)[10] to evaluate the goodness of the outcomes of the algorithms, comparing them with the reference grouping. This metric takes into account the good pairwise decisions made by the algorithm, with a maximum value of 1 when the partitions compared are equal and a value of 0 when comparing two random partitions. Higher values of ARI mean more similarity with the reference grouping and consequently better results.

The statistical significance of the clustering results was assessed using a single-tailed Sign Test[7], where the results (ARI) for each initialisation of the seeds (please see next section) of the algorithms being compared are the pairs of observations. An improvement was considered significant if p-value≤ 0.05.

4.4 Parameters

Both our method and the one proposed by Song et. al. have just one parameter, t, the number of minimum n-grams or named entities (respectively) which must be shared between documents to create a constraint between them. When a distinction must be made between the t of each method they will be named t_{tri} and t_{ent}. In our method the size of the n-grams could be treated as a parameter, but in the experiments we have chosen to set it to 3 as preliminary tests showed that trigrams had a good and consistent performance.

As for the clustering algorithms, we have considered that the number of clusters of each collection is known, and consequently we have set the wanted number of cluster to that value. Therefore, CNC is left with two parameters, β (the strength of the constraints) and d (the number of eigenvectors used in the projection phase), being this last parameter the only one for NC. Since the focus is not on the clustering algorithms, but on the constraints, an array of values was tested in order to show the best results which can be attained with the given constraints. Finally, k-Means was used to cluster the reduced representations of the documents. As the outcome of this algorithm is dependent on the original seeds, ten random initialisations of these seeds were tested (the same seeds were used for NC and CNC with each of the constraint extraction methods). In the next section we report the average ARI of these initialisations.

5 Results and Discussion

Table 1 compares the amount of constraints created and the percentage of them which are accurate for our method, Trigrams, and the entity based one in the two datasets. There is a perceptible difference in the behaviour of the methods in each dataset: while in (i) the amount of constraints created using entities is

larger, in (ii) it is mostly the other way around. We think that this is explained by the differences between the documents in the datasets. On the one hand, the news stories in (i) are full of named entities indicating locations, organisations and persons, most of which only span one of two words. Consequently, if they are surrounded by stopwords (something which is very likely) they will be pruned by our method, while Song et al.'s will use them. On the other hand, in (ii), which is composed by newsgroup posts, their method finds it difficult to find those named entities, whereas ours makes the most of the quotations that users make of other posts.

Table 1. Constraints and percentage of them accurate for each constraint generation method

Dataset (i)

	Trigrams		Entities	
t	const.	accurate	const.	accurate
1	244,738	42.57%	1,524,140	33.73%
2	58,295	57.02%	507,282	39.55%
3	24,350	65.67%	210,218	48.28%
4	12,136	74.13%	96,554	56.98%
5	7,120	79.86%	48,836	63.30%

Dataset (ii)

	Trigrams		Entities	
t	const.	accurate	const.	accurate
1	141,579	54.12%	214,929	51.96%
2	53,330	73.87%	33,229	65.49%
3	31,732	80.01%	8,485	74.64%
4	24,891	79.33%	3,268	80.23%
5	19,194	77.72%	1,692	85.28%

All the same, the comparison between constraints sets of similar size, $t_{3gr} = 1, t_{ent} = 3$ in (i) and $t_{3gr} = 3, t_{ent} = 2$ in (ii), shows that our method respectively slightly underperforms and clearly outperforms Song et al.'s regarding the ratio of accurate constraints created. That is, the accuracy of this new method, which uses lower level information, appears to be comparable or better than that of the entity based one.

Table 2 shows the overlap between the constraints created with both methods. In the two datasets, and both for the total amount of possible constraints created with either method (i.e. $t_{3gr} = 1, t_{ent} = 1$) and for the sets indicated in the previous paragraph the portion of shared constraints is under 60%. This is specially significant when comparing the whole sets of possible constraints extracted from (i). Even though Song et al.'s method creates above five times more constraints, the overlap between it and our method is only of a 58%, descending to a 28% when considering only the accurate constraints (where the ratio is close to 1:5). This result shows that, despite the similarities between both methods noted in Sect. 4.2, the information contained in n-grams is still quite different and can be exploited to create original constraints.

Finally, Table 3 shows the results of the clustering experiments over the two datasets when using the constraints extracted with the two methods compared. In both datasets for each t the best results when using the constraints created with our method improve both the best unconstrained baseline (NC) and the best results obtained with the constraints extracted using named en-

Table 2. Overlap between the constraints created with both methods for selected values of t. Values (amount of constraints shared and % over those created with each method) are shown for all the constraints and for only the accurate ones

Dataset (i)

t		shared		
3gr.	ent.	amount	% 3gr.	% ent.
1	1	140,983	57.61%	9.25%
(accurate)		68,807	28.11%	13.38%
1	3	80,732	32.99%	38.40%
(accurate)		43,774	42.02%	43.13%

Dataset (ii)

t		shared		
3gr.	ent.	amount	% 3gr.	% ent.
1	1	58,763	41.51%	27.34%
(accurate)		42,184	55.05%	37.77%
3	2	12,980	40.91%	39.06%
(accurate)		10,302	40.58%	47.34%

tities, improvements which are in most cases statistically significant. Moreover, the constraints created using the approach presented in this paper were never harmful. That is, for all values of t_{3gr} there are several values of β for which the resulting clustering improves the unconstrained baseline. This is not the case when using Song et al.'s approach: in dataset (i) with $t_{ent} = 2$ all experiments performed worse than the baseline, with values for best result (obtained with β set to 1.25E-4) being even statistically significantly worse. Overall, the results using the constraints created with n-grams show a more stable behaviour with respect to β, and, as for t, for almost any combination of t_{3gr} and t_{ent} the best result with trigrams is better than the best one using entities. Finally, the good results of our approach with $t_{tri} = 1$ suggest that it can be successfully used as a parameter-free method.

On a general note, the results of the clustering experiments show how, although informative, the trends found when studying in abstract terms numbers of constraints and accuracy ratios are not necessarily translated to the final clustering results. For instance, in Table 1 we can see how setting t_{tri} to 1 and t_{ent} to 2 in (i) or setting t_{tri} and t_{ent} to 1 in (ii) yields sets of constraints of similar accuracy but with a noticeable larger amount of entity-based constraints. As usually more accuracy comes at the cost of tighter policies when creating constraints, which would mean less constraints, this would suggest that the entity-based constraints could perform better, as we are able to attain the same accuracy in a larger set of constraints. However, Table 3 shows how in the first example the difference in the average ARI is negligible, while in the second the trigram-based constraints yield markedly better results. Something similar happens when comparing the effects of setting t_{tri} to 1 and t_{ent} to 3 in (i): although the resulting sets of constraints are similar in terms of size and accuracy (and hence we could expect a similar effect on clustering) the clustering experiments show again that entities perform appreciably better. This should be taken into account when investigating new constraint creation methods.

Table 3. Best average ARI of the results of CNC with the given β over ten random initialisations of the seeds using the constraints yielded by each method with the given t. The value of d for which the best value was obtained is between parentheses. The "Baseline" value is the best average ARI of NC. **Bold**=Best for method and t. **Bold & enlarged**=Best in dataset. †=Stat. sign. improvement over unconstrained. ‡=Stat. sign. improvement over unconstrained and the other method with same t and best β.

Dataset (i)

	$t=1$		$t=2$		$t=3$	
β	Trigrams	Entities	Trigrams	Entities	Trigrams	Entities
1.25E-4	0.504 (4)	0.501 (4)	0.504 (4)	**0.502 (4)**	0.504 (4)	0.503 (4)
2.5E-4	0.504 (4)	0.501 (4)	0.504 (4)	0.502 (4)	0.504 (4)	0.503 (4)
5.0E-4	0.504 (4)	0.505 (4)	0.505 (4)	0.500 (4)	0.504 (4)	0.503 (4)
6.25E-4	0.505 (4)	0.506 (4)	0.505 (4)	0.501 (4)	0.504 (4)	0.504 (4)
0.00125	0.508 (4)	0.507 (4)	0.507 (4)	0.501 (4)	0.505 (4)	**0.506†(4)**
0.0025	0.509 (4)	**0.517†(4)**	0.512 (4)	0.497 (4)	0.508 (4)	0.502 (4)
0.0050	0.511 (4)	0.501 (4)	0.512 (4)	0.497 (4)	0.512 (4)	0.503 (4)
0.00625	0.506 (4)	0.478 (4)	**0.515‡(4)**	0.495 (4)	0.513 (4)	0.502 (4)
0.0125	**0.538‡(5)**	0.448 (11)	0.514 (4)	0.419 (11)	0.515 (4)	0.457 (12)
0.025	0.537 (5)	0.117 (5)	0.513 (8)	0.419 (11)	0.513 (4)	0.475 (12)
0.05	0.532 (49)	0.041 (4)	0.483 (4)	0.339 (50)	0.509 (6)	0.419 (17)
0.0625	0.443 (14)	0.005 (4)	0.512 (5)	0.316 (49)	0.504 (4)	0.412 (16)
0.125	0.331 (48)	0.001 (91)	0.485 (5)	0.114 (60)	**0.541 (5)**	0.337 (4)
0.25	0.098 (5)	0.001 (97)	0.432 (26)	0.012 (5)	0.538 (5)	0.285 (66)
0.5	0.005 (4)	0.001 (100)	0.379 (37)	0.005 (4)	0.495 (6)	0.126 (54)
Baseline	0.504 (4)					

Dataset (ii)

	$t=1$		$t=2$		$t=3$	
β	Trigrams	Entities	Trigrams	Entities	Trigrams	Entities
1.25E-4	0.284 (15)	0.283 (15)	0.284 (15)	0.283 (15)	0.283 (15)	0.283 (15)
2.5E-4	0.285 (15)	0.286 (18)	0.284 (15)	0.283 (15)	0.284 (15)	0.283 (15)
5.0E-4	0.285 (15)	0.282 (15)	0.284 (15)	0.284 (15)	0.285 (15)	0.283 (15)
6.25E-4	0.286 (15)	0.290 (18)	0.285 (15)	0.285 (15)	0.285 (15)	0.283 (15)
0.00125	0.286 (15)	0.290 (18)	0.286 (18)	0.286 (15)	0.291 (15)	0.282 (15)
0.0025	0.294 (18)	0.283 (17)	0.288 (15)	0.292 (18)	0.287 (18)	0.283 (18)
0.0050	0.294 (22)	0.292 (20)	0.292 (18)	0.284 (18)	0.288 (15)	0.289 (18)
0.00625	0.290 (16)	0.286 (20)	0.288 (18)	0.298 (15)	0.280 (18)	0.288 (18)
0.0125	0.296 (22)	0.277 (27)	0.292 (7)	0.280 (15)	0.292 (16)	0.278 (16)
0.025	**0.356‡(17)**	0.281 (51)	0.368 (8)	0.295 (19)	0.309 (9)	0.285 (12)
0.05	0.300 (36)	**0.306 (21)**	**0.397‡(15)**	0.310 (24)	0.385 (10)	0.305 (12)
0.0625	0.151 (90)	0.297 (39)	0.378 (15)	**0.327†(24)**	**0.413‡(10)**	0.321 (12)
0.125	0.003 (15)	0.020 (4)	0.278 (30)	0.324 (36)	0.346 (28)	**0.327†(12)**
0.25	0.000 (9)	0.001 (3)	0.002 (4)	0.274 (43)	0.272 (99)	0.322 (12)
0.5	0.002 (93)	0.001 (6)	0.000 (3)	0.071 (73)	0.001 (3)	0.310 (38)
Baseline	0.283 (18)					

6 Related Work

As for automatic constraint extraction, the method most related to ours is Song et al.'s [15], which was explained in detail in Sect. 4.2 as it was used as baseline in this paper. Other notable method, albeit with a different approach, is [9], which makes preliminary clusterings of the data in order to detect related data instances. These are examples of methods which use internal information of the data to cluster to extract constraints, whereas other methods use external information; a recent example is [2], which uses del.icio.us tags to create positive constraints between web pages.

On the other hand, word n-grams have been widely used in Information Retrieval and Data Mining. This dates back to works such as [12], which shows that indexing word bigrams (*statistic phrases*) works comparably well as indexing syntactic phrases, and has given interesting results such as [6], where n-grams are used to detect near-duplicates. Far from being confined to theoretical works, large companies have taken interest in practical applications of n-grams, as is the case of Google, which has released a n-gram corpus[5] to encourage research in this field.

7 Conclusion and Future Work

In this paper we have presented a novel approach based on word n-grams to automatically extract positive clustering constraints from textual collections. The experiments performed, where we have compared this new method with an existing similar technique based on higher level information, have shown that our new approach creates constraints with similar accuracy, which convey mostly new information and yield statistically significantly better results in a constrained clustering task. It also attains a greater and again statistically significant improvement over a high performing non-constrained baseline. These results hold in two different datasets, composed by documents of different nature selected from reference clustering collections.

In the future, we would like to look into ways to improve the filtering of n-grams done in the second step of the algorithm, as well as into techniques to combine constraints obtained with different approaches. Also, further experiments should be conducted in order to assess the feasibility of using the proposed approach as a parameter-free method.

Acknowledgement The first author wants to acknowledge the support of Ministerio de Educación from the Spanish Government under the FPU Grant AP2007-02476.

References

1. Ares, M.E., Parapar, J., Álvaro Barreiro: An experimental study of constrained clustering effectiveness in presence of erroneous constraints. Information Processing & Management 48, 537–551 (2012)

2. Ares, M.E., Parapar, J., Barreiro, A.: Improving text clustering with social tagging. In: Proceedings of the Fifth International Conference on Weblogs and Social Media, ICWSM 2011, Barcelona, Spain, July 17-21, 2011. pp. 430–433. The AAAI Press (2011)
3. Basu, S., Banjeree, A., Mooney, E., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: In Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04). pp. 333–344 (2004)
4. Basu, S., Davidson, I., Wagstaff, K.: Constrained Clustering: Advances in Algorithms, Theory, and Applications. Chapman & Hall/CRC (2008)
5. Brants, T., Franz, A.: Google research blog: All our n-gram are belong to you (2006), http://googleresearch.blogspot.com.es/2006/08/all-our-n-gram-are-belong-to-you.html, [Online; accessed 11/05/2012]
6. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G.: Syntactic clustering of the web. In: Selected papers from the sixth international conference on World Wide Web. pp. 1157–1166. Elsevier Science Publishers Ltd., Essex, UK (1997)
7. Conover, W.J.: Practical nonparametric statistics. John Wiley & Sons, New York, third edn. (1971)
8. Ding, C.: A tutorial on spectral clustering. Tutorial presented at ICML 2004: 21st International Conference on Machine Learning (2004), http://ranger.uta.edu/~chqding/Spectral/
9. Greene, D., Cunningham, P.: Constraint selection by committee: An ensemble approach to identifying informative constraints for semi-supervised clustering. In: Proceedings of the 18th European conference on Machine Learning. pp. 140–151. ECML '07, Springer-Verlag, Berlin, Heidelberg (2007)
10. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2, 193–218 (1985)
11. Ji, X., Xu, W., Zhu, S.: Document clustering with prior knowledge. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 405–412. ACM (2006)
12. Mitra, M., Buckley, C., Singhal, A., Cardie, C.: An analysis of statistical and syntactic phrases. In: Proceedings of RIAO-97. pp. 200–214 (1997)
13. Pantel, P., Lin, D.: Document clustering with committees. In: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 199–206. ACM Press (2002)
14. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 888–905 (2000)
15. Song, Y., Pan, S., Liu, S., Wei, F., Zhou, M.X., Qian, W.: Constrained coclustering for textual documents. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010. The AAAI Press (2010)