Correlation Clustering for Crosslingual Link Detection

Jurgen Van Gael and Xiaojin Zhu
Computer Sciences Department
University of Wisconsin-Madison
Madison, WI 53706
{JVANGAEL, JERRYZHU@CS.WISC.EDU}

Abstract

The crosslingual link detection problem calls for identifying news articles in multiple languages that report on the same news event. This paper presents a novel approach based on constrained clustering. We discuss a general way for constrained clustering using a recent, graph-based clustering framework called correlation clustering. We introduce a correlation clustering implementation that features linear program chunking to allow processing larger datasets. We show how to apply the correlation clustering algorithm to the crosslingual link detection problem and present experimental results that show correlation clustering improves upon the hierarchical clustering approaches commonly used in link detection, and, hierarchical clustering approaches that take constraints into account.

1 Introduction

Crosslingual link detection is the problem of identifying news articles in multiple languages that report on the same news event. It is an important component in online information processing systems, with applications in security and information retrieval. Existing link detection systems are mostly monolingual, with a small number of bilingual link detection systems [Allan *et al.*, 2000; Chen and Chen, 2002; Spitters and Kraaij, 2002] and very few crosslingual link detection systems [Pouliquen *et al.*, 2004] that work across many languages. Like the latter, we assume monolingual link detection has been done, such that news articles on the same event in a single language already form a single group. This assumption is mild, as existing systems like Google News (`http://news.google.com`, the 'all n related' links) do just this. Our goal is thus to cluster these monolingual groups from different languages over a period of time, so that groups reporting on the same event are in the same cluster. One needs to take two things into consideration: 1. We would rather *not* cluster any monolingual groups from the same language together since we assume monolingual link detection has done a reasonable job. This is known as 'cannot-links' in constrained clustering as we will discuss later; 2. We in general do not know the number of clusters in advance.

In this paper we propose a principled approach to the crosslingual link detection task using *correlation clustering* [Bansal *et al.*, 2004; Demaine and Immorlica, 2003]. Correlation clustering is a recent graph-based clustering framework with interesting theoretical properties. It can be formulated to solve constrained clustering (also known as semi-supervised clustering), which we will use for crosslingual link detection. In constrained clustering, one performs clustering with additional constraints (or preferences) on the data points. Two typical constraints are must-link (where two items must be in the same cluster) and cannot-link (where two items cannot be in the same cluster). Constrained clustering has received considerable attention in machine learning [Bilenko *et al.*, 2004; Wagstaff *et al.*, 2001; Xing *et al.*, 2003]; we point to [Basu *et al.*, 2006] for further references. Solving the correlation clustering problem is hard but one natural way to approximate the best solution is to encode it in a linear programming optimization framework. We combine correlation clustering with a large-scale linear program solution technique known as 'chunking' in order to solve larger crosslingual link detection problems. The contribution of our paper is twofold:

1. we introduce a practical way for solving the complex correlation clustering algorithm in [Demaine and Immorlica, 2003];
2. we demonstrate good performance on crosslingual link detection using the correlation clustering approach.

In the rest of the paper, we start by reviewing correlation clustering and discuss how to implement it using linear programming chunking in section 2. We discuss related work in constrained clustering and crosslingual link detection in section 3. Finally we present experiments in section 4 where we improve upon existing crosslingual link detection systems.

2 Correlation Clustering

Consider the following problem: we are given a weighted graph for which we want to partition the nodes into clusters. If two nodes share an edge with positive weight, we prefer they be in the same cluster; if they share an edge with negative weight, we prefer they end up in different clusters. The goal of correlation clustering is to partition the graph into clusters to maximally satisfy these preferences.

We review the discussion in [Demaine and Immorlica, 2003] on how to formally describe correlation clustering as an integer program (IP). Let $G = (V, E)$ be a graph with weight w_e for every edge[1] $e \in E$. Let E^+ be the set of edges with positive weights, $E^+ = \{e \in E | w_e > 0\}$ and E^- be the set of edges with negative weight, $E^- = \{e \in E | w_e < 0\}$. We now associate a binary variable x_{uv} with every edge $(uv) \in E$ with the following interpretation: if $x_{uv} = 1$ then u, v are in different partitions, if $x_{uv} = 0$ then u, v are in the same partition. Intuitively x_{uv} is the binary indicator variable for whether we cut the edge or not. Correlation clustering minimizes the following objective

$$\sum_{e \in E^+} w_e x_e + \sum_{e \in E^-} -w_e(1 - x_e). \tag{1}$$

We want the variables to correspond to a valid partitioning: if u, v are in the same cluster and v, t are in the same cluster, then u, t must be so too. This can be achieved by the triangle inequality constraints $x_{uv} + x_{vt} \geq x_{ut}$ below. Simplifying the objective function we find the correlation clustering integer program:

$$\begin{array}{lll} \min_{\mathbf{x}} & \sum_{e \in E} w_e x_e & \\ \text{subject to} & x_e \in \{0, 1\}, & \forall e \in E \\ & x_{uv} + x_{vt} \geq x_{ut}, & \forall uv, vt, ut \in E \\ & x_{uv} = x_{vu}, & \forall u, v \in V \end{array} \tag{2}$$

The weights w are input to the algorithm, and can encode must-links and cannot-links besides similarities between data items. As formulated above, correlation clustering has two attractive properties that make it suitable for crosslingual link detection in particular and constrained clustering in general. First of all, one does not need to specify the number of clusters; the algorithm determines the optimal number of clusters automatically. Secondly, the graph edge weights can be arbitrary and do not need to satisfy any metric condition.

2.1 Linear Program Approximation

Unfortunately solving the correlation clustering IP in (2) exactly is NP-Hard. Recent theoretical results on approximation algorithms [Bansal *et al.*, 2004], in particular [Demaine and Immorlica, 2003], propose practical approaches to correlation clustering. We build on the work in [Demaine and Immorlica, 2003] where the authors describe an $O(\log n)$ approximation by relaxing the IP to a linear program (LP), and rounding the solution of the LP by a region growing technique. We replace constraint $x_e \in \{0, 1\}$ by $x_e \in [0, 1]$ in equation (2) to relax the IP to an LP. The solution to this LP might include fractional values which we will have to round. We point to [Demaine and Immorlica, 2003] for a detailed description and theoretical analysis of the rounding algorithm and limit ourselves to a qualitative description in this paper. One can interpret the value of the LP variables as distances: when a variable has value 0, the two adjacent nodes go in the same cluster as their distance is 0 while if a variable is 1, the two adjacent nodes go into different clusters. The rounding procedure now needs to decide on how to partition the graph given that some nodes are at fractional distances away from each other. Intuitively, the rounding algorithm will pick a node in the graph and gradually grow a ball centered around this node. While increasing the radius of the ball, all the nodes that are at a distance smaller than the radius away from the center of the ball will be included in the ball. The radius grows until some technical termination condition is met. All the nodes in the ball are then put into one cluster and removed from the graph. This procedure is repeated until there are no more nodes left in the graph. [Demaine and Immorlica, 2003] prove that the original objective function (equation (1)) of the LP relaxation will be bounded above by $O(\log n)$ times the objective function of the IP where n is the number of nodes in the graph.[2]

Unfortunately, the triangle inequalities could introduce up to $O(n^3)$ constraints in the LP, which puts a heavy burden on memory requirements. Next we discuss how we tradeoff memory for runtime so we can solve correlation clustering for larger problem sizes.

2.2 LP Chunking

Linear program chunking [Bradley and Mangasarian, 2000] is a technique to convert a large linear program into an iterative procedure on much smaller sub-problems, thus reducing the memory need. The iterative procedure produces the same solution and is guaranteed to terminate. It works as follows: one first breaks up all the constraints into *chunks* and solves the optimization problem using only the first chunk of constraints. The *active constraints* are those inequality constraints that achieve equality at the solution. Next, one keeps only the active constraints from the first chunk, adds all constraints from the second chunk, and solves the LP again. This procedure is repeated, looping through all chunks over and over until some convergence criterion is met. One can arbitrarily set the size of the chunks to reduce the memory load of the iterative procedure.

Let a general linear program be described as,

$$\min_{\mathbf{x}} \{c^\top x | Hx \geq b\}, \tag{3}$$

with $c \in \mathbb{R}^n, H \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$. Let the constraints $[H \quad b]$ be partitioned into l blocks, possibly of different sizes, as follows:

$$[H \quad b] = \begin{bmatrix} H^1 & b^1 \\ \vdots & \vdots \\ H^l & b^l \end{bmatrix} \tag{4}$$

At iteration j we compute x^j by solving the following linear program,

$$\min_{\mathbf{x}^j} \{c^\top x^j | H^{(j \mod l)} x^j \geq b^{(j \mod l)} \wedge \bar{H}^j x^j \geq \bar{b}^j\}, \tag{5}$$

[1] We will denote an edge both as $e \in E$ and as a pair of vertices $(uv) \in E$

[2] Although the theoretical analysis in [Demaine and Immorlica, 2003] requires the ball to grow continuously, this is not practical. By redefining the volume in [Demaine and Immorlica, 2003] to be the total volume of edges inside the ball *as well as the total volume inside the cut*, i.e., replace $p_{vw} \cdot x_{vw} \cdot (r - x_{uv})$ by $p_{vw} \cdot x_{vw}$ in their definition, and modifying the description of step 3 in their algorithm as: 'Grow r by $\min\{x_{uv} - r > 0, v \notin B(u, r)\}$ so that $B(u, r)$ includes *another entire edge*', the theoretical guarantee stays the same but we only need to check the radius r a finite number of times.

where $[\bar{H}^0 \quad \bar{b}^0]$ is empty and $[\bar{H}^j \quad \bar{b}^j]$ is the set of active constraint, i.e. all inequalities satisfied as equalities by x^j at iteration j. We stop iterating when $c^T x^j = c^T x^{j+\nu}$ for some pre-specified integer ν. We point to [Bradley and Mangasarian, 2000] for more details and proofs of the finite termination of this algorithm.

3 Related Work

Constrained or semi-supervised clustering has enjoyed some recent attention [Basu *et al.*, 2006; Bilenko *et al.*, 2004; Davidson and Ravi, 2005; Wagstaff *et al.*, 2001; Xing *et al.*, 2003]. In [Basu *et al.*, 2006], the authors categorize all semi-supervised methods into two classes: *constraint-based* and *distance-based* methods. The constraint-based methods, such as [Wagstaff *et al.*, 2001] and to which our approach belongs, rely on the must-link and cannot-link constraints to guide the clustering algorithm in finding a partitioning that does not violate the constraints. Distance-based methods, such as [Xing *et al.*, 2003], learn a metric using the constraint information and then apply existing clustering algorithms to the data points in the learned metric space. These approaches require specifying the number of clusters beforehand. One solution to this issue is to use variants of hierarchical clustering that take constraints into account, e.g. [Davidson and Ravi, 2005]. By changing where to cut the dendrogram, one can control the number of clusters. The main difference between hierarchical clustering with constraints and correlation clustering is that the former makes local, greedy decisions at every step while correlation clustering optimizes the clustering over the whole graph at once. One motivation for our work is the observation that the crosslingual link detection systems in [Pouliquen *et al.*, 2004; Allan *et al.*, 2000; Chen and Chen, 2002] do not use constrained clustering techniques.

So far, correlation clustering has not been applied to machine learning tasks very often. We are only aware of [McCallum and Wellner, 2005] who implement a more restricted version of correlation clustering in [Bansal *et al.*, 2004] for noun co-reference.

The only crosslingual link detection system that covers a large set of languages we are aware of is described in [Pouliquen *et al.*, 2004]. The authors describe a system which performs crosslingual link detection as well as monolingual news tracking, i.e. the identification of related news over time in one particular language. Their approach uses a very rich article representation based on extracting named entities, keywords and geographical names. In addition, the articles are mapped onto the multilingual thesaurus EUROVOC [Steinberger *et al.*, 2002] which categorizes the articles in several of 6000 hierarchically organized subjects. Our system, on the other hand, uses machine translation tools to represent articles in a uniform way. This is a common [Diab and Resnik, 2001] way of working with multilingual corpora. Our experiments show that although the translation is noisy, it does not significantly affect performance. Our crosslingual link detection task is also related to the work in [Diaz and Metzler, 2007], where the authors introduce a framework for aligning documents in parallel corpora based on topical correspondence.

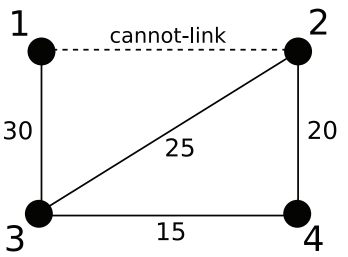


Figure 2: Toy dataset

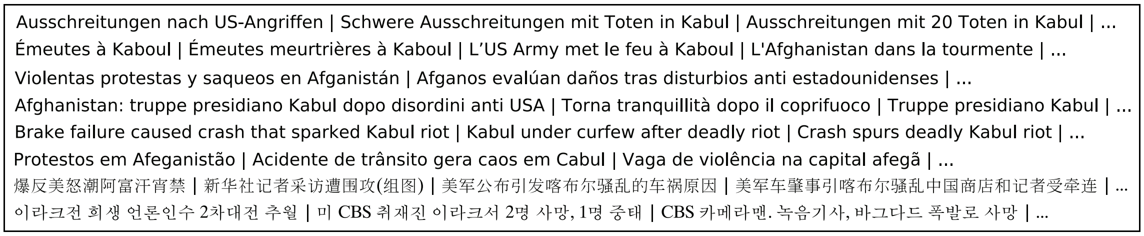
4 Experiments

In this section, we first illustrate correlation clustering on a toy dataset. We then discuss how to solve the crosslingual link detection problem using a correlation clustering based constrained clustering approach and show how this improves upon existing hierarchical clustering approaches.

4.1 Correlation Clustering on a Toy Dataset

It is straightforward to adapt correlation clustering for constrained clustering. Say we are given a set of items $U = \{u_1, u_2, \cdots, u_l\}$, a pairwise similarity measure $S : U \times U \rightarrow \mathbb{R}$, a set $C_M \subset U \times U$ of must-link constraints and a set $C_C \subset U \times U$ of cannot-link constraints. We build a graph G where the set of vertices is U. As a first step, we add an edge for all pairs of nodes not in $C_M \cup C_C$ and set the edge weight according to the similarity measure S. Let M be a constant that is sufficiently larger than the sum of the absolute values of all weights in the graph so far. In the second step, for all the pairs in C_M and C_C, we add either hard or soft preferences: if we assume that the constraints are hard, we add an edge for every must-link constraints with weight M and an edge for every cannot-link constraint with weight $-M$. If we want soft preferences, we can use values smaller than M according to the strength of the preferences.

Figure 2 shows a toy dataset consisting of four nodes with a cannot-link constraint between nodes 1 and 2. The weights are specified in the figure. The edge not shown in the figure has a similarity of zero. We use -1000 for the cannot-link constraint edge weight. The objective function for this dataset is to minimize $-1000x_{(1,2)} + 30x_{(1,3)} + 25x_{(2,3)} + 20x_{(2,4)} + 15x_{(3,4)}$ subject to the triangle inequality constraints. Solving the IP exactly would give us a solution that assigns 1 to all variables except $x_{(2,3)} = x_{(2,4)} = x_{(3,4)} = 0$; this corresponds to the clustering $\{1\}, \{2, 3, 4\}$. Although nodes 1 and 3 have the highest similarity, the cannot-link constraint guides the correlation clustering algorithm to not take node 1 into the cluster with 2 and 3. Note how a hierarchical clustering algorithm would start off wrong as it merges nodes 1 and 3 together and thus fails to find the best clustering. Even a hierarchical clustering algorithm that takes constraints into account will not find the best clustering as it will greedily merge nodes 1 and 3 together.



Ausschreitungen nach US-Angriffen | Schwere Ausschreitungen mit Toten in Kabul | Ausschreitungen mit 20 Toten in Kabul | ...
Émeutes à Kaboul | Émeutes meurtrières à Kaboul | L'US Army met le feu à Kaboul | L'Afghanistan dans la tourmente | ...
Violentas protestas y saqueos en Afganistán | Afganos evalúan daños tras disturbios anti estadounidenses | ...
Afghanistan: truppe presidiano Kabul dopo disordini anti USA | Torna tranquillità dopo il coprifuoco | Truppe presidiano Kabul | ...
Brake failure caused crash that sparked Kabul riot | Kabul under curfew after deadly riot | Crash spurs deadly Kabul riot | ...
Protestos em Afeganistão | Acidente de trânsito gera caos em Cabul | Vaga de violência na capital afegã | ...
爆反美怒潮阿富汗宵禁 | 新华社记者采访遭围攻(组图) | 美军公布引发喀布尔骚乱的车祸原因 | 美军车肇事引喀布尔骚乱中国商店和记者受牵连 | ...
이라크전 희생 언론인수 2차대전 추월 | 미 CBS 취재진 이라크서 2명 사망, 1명 중태 | CBS 카메라맨. 녹음기사, 바그다드 폭발로 사망 | ...

Figure 1: Samples from the large dataset.

4.2 Crosslingual link detection

We generated five datasets by crawling Google News. We specifically focused our experiments on news articles which Google categorized as 'world news' as we assume this is the category where the most interesting cross-lingual links can be made. The first four datasets each consist of 60 monolingual 'world news' article groups from three languages: English, German and French. Each of these four datasets was generated one week apart by selecting the top 20 article groups for each language in April 2006. This results in a total of 60 article groups in each dataset. In May 2006, we generated the fifth dataset which is larger and consists of 160 article groups from the 'world news' category in eight different languages: English, German, Italian, French, Portuguese, Spanish, Korean and Chinese. Figure 1 shows a sample from the larger dataset. For all five datasets, we manually created a ground truth clustering[3].

For correlation clustering, we construct a fully-connected graph where each node is a monolingual article group. We create cannot-links between all pairs of article groups from the same language and choose -10^8 as the weight for these cannot-link edges. We compute similarity values between article groups from different languages with the following procedure: first we concatenate all the article titles in a monolingual group to form a 'document representation' for the group. We then use Google machine translation to automatically translate the 'document' into English, and remove stopwords from the translation. Therefore monolingual groups in different languages are represented by their corresponding (noisy) English translation, providing a way to compute their similarities. Empirically we found no difference in performance using different machine translation tools such as Babelfish and Wordlingo. Next, for each monolingual group, we convert the translated document into a TF.IDF vector $\bar{w} = (w_1 w_2 \cdots w_{|V|})$, with $w_i = n_i \cdot \log(|D|/|D_i|)$, where n_i is the number of times word i appears in the document representing the article group, D represents the set of article groups in the dataset and D_i represents the set of article groups that include word w_i. We compute the similarity s_{wv} between any two TF.IDF vectors $\bar{w}, \bar{v}$ as their inner product,

$$s_{\bar{w}\bar{v}} = \sum_{i=1}^{|V|} w_i \cdot v_i. \tag{6}$$

Note that even with stop-word removal, two unrelated article groups often have a small but positive similarity due to common words. If we use the similarity (6) directly as graph edge weights for correlation clustering, many irrelevant groups will be clustered together. For the problem of link detection, this is clearly not desirable. We therefore subtract a bias constant t from all similarity values so that $w_{uv} = s_{\bar{u}\bar{v}} - t$. Intuitively, too small a similarity (6) between two article groups is in fact evidence that they should *not* be in the same cluster. By changing the bias t we change the resulting clustering, which is how we generate precision-recall curves. For all the experiments presented below, we chose our bias values as follows: we started with a bias such that only one edge in the graph remains positively weighted. Next, we steadily increase the bias such that another 0.1% of the edges becomes positively weighted. On the small datasets, we repeated the experiments until 75% of the edges are positively weighted while on the larger datasets we repeat the experiments until 10% of the edges are positively weighted. We compute precision and recall values relative to our manually labeled ground truth. We count an edge as true positive (TP), if its two article groups appear in the same cluster in both ground truth and our results, false positive (FP) if they do not appear in the same cluster in ground truth but do appear together in our results, and so on. Precision and recall is a better measure than accuracy for our task, since the baseline of classifying every edge as 'not in same cluster' would have high accuracy because of the large number of true negatives. We used CPLEX 9.0 on a 3.0 GHz machine with 2GB RAM to solve the linear programs.

Our first round of experiments are designed to illustrate how taking constraints into account improves performance on the crosslingual link detection problem. We compare our correlation clustering algorithm to the hierarchical clustering approach which has commonly been used for the crosslingual link detection problem, [Chen and Chen, 2002; Pouliquen *et al.*, 2004], and constrained hierarchical clustering such as [Davidson and Ravi, 2005]. Hierarchical clustering is done by choosing a bias value and adding edges to the graph in descending order according to their weight until the edge weights become smaller than the bias. We then output the connected components as the resulting clusters. Constrained hierarchical clustering is similar, except that at every step we only add an edge if it does not introduce a path between two nodes in a cannot-link constraint. Again, we output the connected components as the resulting clusters. The left plot in Figure 3 shows the average precision-recall over our four small datasets. If we keep the number of positively weighted edges small (large bias) then both types of hierar-

[3] The datasets are available at `http://www.cs.wisc.edu/~jvangael/newsdata/`.

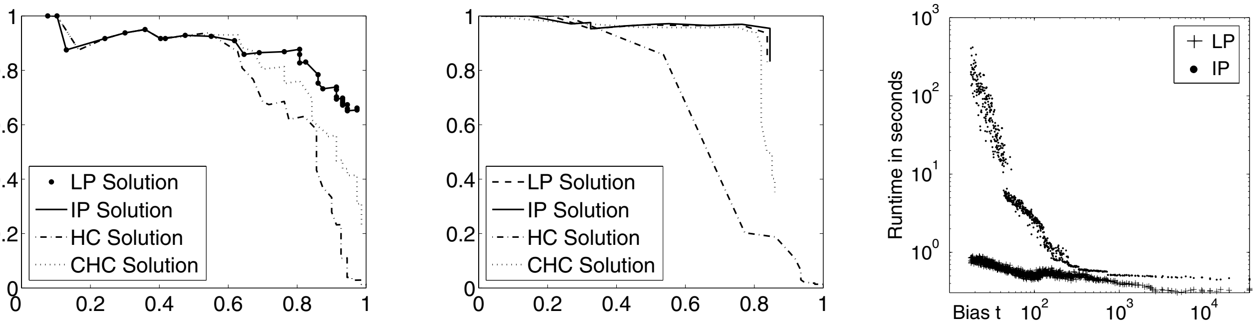


Figure 3: Left: average precision-recall over four small datasets. Middle: precision-recall for the large dataset. Right: average runtime over four small datasets.

chical clustering perform as well as correlation clustering. Inspecting the datasets, this behavior can be explained by the fact that there are a number of news events for which the articles use a subset of the vocabulary that is not commonly used in other articles. Our similarity measure assigns large weights among article groups in different languages on these events and very small weight between these article groups and article groups on a different topic. In a sense these are 'easy' instances which both hierarchical clustering approaches as well as correlation clustering get right. If we increase the number of positive edges (small bias) then the simple hierarchical clustering algorithm performs much worse than correlation clustering. As a simple hierarchical clustering approach has no notion of cannot-link constraints, it will cluster groups from the same language together. Usually, crosslingual link detection systems choose to leave these clusters out, but this decision comes at the price of lower recall. Constrained hierarchical clustering performs a little better as it takes our assumption about the correctness of the monolingual groups into account. Nonetheless, Figure 3 also shows that correlation clustering, which takes the whole graph into account instead of making local greedy decisions can still outperform constrained hierarchical clustering. We attempted to compare our approach to the constrained clustering in [Bilenko *et al.*, 2004] using their UTWeka implementation. The implementation ended up returning many empty clusters, resulting in low precision and recall.

The middle plot in Figure 3 shows the precision-recall for the large dataset; it indicates the trend we observed with the smaller datasets: taking into account constraints can still improve the performance of crosslingual link detection.

Next, let us consider the solution found by the approximation algorithm and the exact integer solution. Figure 3 shows that on the small datasets the two solution are exactly equal. Inspecting the LP solutions, we find that in the high bias regime, almost no rounding is necessary as the LP solution is the exact IP solution. Only in the low bias regime, when more edges are positively weighted, rounding becomes necessary. On the large dataset, Figure 3 shows that although there is a small difference between the two solutions, the LP relaxation with rounding does well to find a good approximation to the integer solution. We observed rather unexpected behavior from the rounding algorithm that influences the precision-recall curves: at very low bias, due to the symmetry of the graph, the optimal LP solution has a number of variables with 0.5 values. From the theoretical analysis of the rounding algorithm, we know that a radius cannot grow to be 0.5. As a result of these properties, in the low bias regime a large number of nodes will end up as singleton clusters. This prohibits recall from increasing to 1.0 and we observe the precision-recall curve loop back towards lower recall and higher precision. Because the curve essentially follows the first path 'in the opposite direction' we did not include this in Figure 3 for clarity.

Our next experiment was designed to evaluate how much the LP approximation algorithm improves the runtime over solving the IP exactly. The rightmost plot in Figure 3 shows the average runtime over the four small datasets of solving the exact IP compared to solving the approximation algorithm. Every dot in the graph represents the time required to solve either the IP or the LP with rounding for a specific bias. It is clear from this figure that the LP approximation algorithm for correlation clustering is significantly faster than solving the IP directly. However, even on the larger dataset the main bottleneck is not so much the runtime but rather the memory requirements. On this large dataset, the underlying graph has 160 nodes which results in over $2,000,000$ constraints for both the IP and LP. This is about as large a correlation clustering instance we can solve without using chunking on our machine with $2GB$ RAM.

Our last experiment shows the results of applying chunking to the LP for correlation clustering. Our experimental setup is the following: we create instances of the correlation clustering with random edge weights, distributed roughly according to the instances of interest to crosslingual link detection. We chose our chunk size to be as large as possible while still having some workspace memory for the processing in between iterations: this resulted in 10^6 constraints per chunk. Finally we use a value of $\nu = 4$ as our stop condition. Table 1 shows the runtime for chunking versus solving the whole LP at once. Correlation clustering instances of size 128 are the first instances where the number of constraints is larger than the chunk size. At this size, the runtime overhead for chunking is mostly due to the stop condition. Starting from graphs with around 200 nodes we cannot fit the whole LP in memory anymore and we must apply chunking to tradeoff memory for runtime. Table 1 shows that chunking is useful for scaling up the size of solvable correlation clustering problem but has its limitations too. First of all, the runtime increases fast: this is due to the fact that doubling the size of the graph roughly corresponds to an eight-fold increase in the number of constraints and equivalently an eight-fold increase in the number of chunks. Another problem that arises is that the set of active constraints ($\bar{H}$) can become larger than the chunk size and exhaust available memory. We believe these problems are inherent to correlation clustering approximations based on integer programming.

# nodes	# constraints	whole LP	chunking LP
64	1×10^5	48	72
128	1×10^6	203	1065
192	3×10^6	out of memory	1402
256	8×10^6	out of memory	2708
320	1×10^7	out of memory	5070
384	2×10^7	out of memory	17298
448	4×10^7	out of memory	52803
512	6×10^7	out of memory	out of memory

Table 1: Runtime in Seconds

5 Conclusion

In this paper we introduce an implementation for correlation clustering using linear program chunking that scales beyond the implementation of the algorithm in [Demaine and Immorlica, 2003]. However, we find that even our chunking method which can trade off memory for runtime has its limits due to the growth ($O(n^3)$) of the linear program size. Nonetheless, we believe that for constrained clustering problems of limited size (a few hundred data points) correlation clustering is worth pursuing. Moreover, our experiments on the crosslingual link detection task show that correlation clustering outperforms both hierarchical clustering and hierarchical clustering with constraints.

In future work, we plan to investigate whether other algorithms for correlation clustering have smaller time and space complexity. Also, we believe it would be interesting to combine correlation clustering and our machine translation based representation with the rich document representation from [Pouliquen *et al.*, 2004] to improve performance of crosslingual link detection even more.

Acknowledgments

We thank Shuchi Chawla for helpful comments on correlation clustering, Michael Thompson and Ted Wild for linear program chunking.

References

[Allan *et al.*, 2000] J. Allan, V. Lavrenko, D. Frey, and V. Khandelwal. UMass at TDT 2000. *Proceedings of Topic Detection and Tracking Workshop*, 2000.

[Bansal *et al.*, 2004] N. Bansal, A. Blum, and S. Chawla. Correlation Clustering. *Machine Learning*, 56(1):89–113, 2004.

[Basu *et al.*, 2006] Sugato Basu, Mikhail Bilenko, Arindam Banerjee, and Raymond J. Mooney. Probabilistic semi-supervised clustering with constraints. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 71–98. MIT Press, 2006.

[Bilenko *et al.*, 2004] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, 2004.

[Bradley and Mangasarian, 2000] PS Bradley and OL Mangasarian. Massive data discrimination via linear support vector machines. *Optimization Methods and Software*, 13(1):1–10, 2000.

[Chen and Chen, 2002] Y.J. Chen and H.H. Chen. NLP and IR approaches to monolingual and multilingual link detection. *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7, 2002.

[Davidson and Ravi, 2005] I. Davidson and S.S. Ravi. Agglomerative Hierarchical Clustering with Constraints: Theoretical and Empirical Results. *Lecture notes in computer science*, pages 59–70, 2005.

[Demaine and Immorlica, 2003] E. Demaine and N. Immorlica. Correlation clustering with partial information. *Proc. of 6th APPROX*, pages 1–13, 2003.

[Diab and Resnik, 2001] M. Diab and P. Resnik. An unsupervised method for word sense tagging using parallel corpora. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 255–262, 2001.

[Diaz and Metzler, 2007] F. Diaz and D. Metzler. Pseudo-Aligned Multilingual Corpora. *Proceedings of the 20th IJCAI*, 2007.

[McCallum and Wellner, 2005] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. *Advances in NIPS*, 17, 2005.

[Pouliquen *et al.*, 2004] B. Pouliquen, R. Steinberger, C. Ignat, E. Käsper, and I. Temnikova. Multilingual and cross-lingual news topic tracking. 20:23–27, 2004.

[Spitters and Kraaij, 2002] M. Spitters and W. Kraaij. Unsupervised event clustering in multilingual news streams. *Proceedings of the LREC2002 Workshop on Event Modeling for Multilingual Document Linking*, pages 42–46, 2002.

[Steinberger *et al.*, 2002] Ralf Steinberger, Bruno Pouliquen, and Johan Hagman. Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc. In *CICLing*, page 415, 2002.

[Wagstaff *et al.*, 2001] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. *Proceedings of the 18th ICML*, pages 577–584, 2001.

[Xing *et al.*, 2003] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. *Advances in NIPS*, 15:505–512, 2003.