A Visual Exploration of Melodic Relationships within Traditional Music Collections

Chris Walshaw
Dept Computing & Information Systems
University of Greenwich, London, UK
c.walshaw@gre.ac.uk

***Abstract*—The aim of this paper is to discuss a technique for visually exploring melodic relationships within traditional tune collections encoded in abc notation, a widely used text-based music representation system particularly popular for folk and traditional music. There are approximately ½ million melodies encoded in abc on the web and abcnotation.com provides a searchable index of the entire corpus with tools to view, download and listen to the scores.**

This paper stems from related work known as TuneGraph which uses a melodic similarity measure to derive a proximity graph representing relationships between tunes in the abc corpus, and which allows users of abcnotation.com to explore melodic similarity. As it stands TuneGraph only gives a localised view of the melodic relationships: this paper aims to look at exploring those relationships at a global (corpus-based) level via a prototype visualisation tool. Currently the tool is not interactive: in this paper the aim is to consider a proof-of-concept approach to explore where there is a useful visualisation possible; future work will look at user interactivity with the tool.

Keywords—melodic similarity, graph drawing

I. Introduction

The aim of this paper is to conduct a visual exploration of melodic relationships within traditional tune collections. It stems from related work into an online application known as TuneGraph, [1], which allows users of abcnotation.com to explore melodic similarity in folk music. TuneGraph uses a similarity measure to derive a proximity graph representing similarities within the abc notation corpus (~½ million tunes) which backs the search engine. From this a local graph is extracted for each vertex, aimed at indicating close variants of the tune represented by the vertex. Finally an interactive user interface displays each local graph on that tune's webpage, allowing the user to explore melodic similarities.

As it stands TuneGraph only gives a very localised view of the melodic relationships: this paper aims to explore those relationships at a global (corpus-based) level.

II. Methodology

To distinguish the work from that discussed in previous papers, here the similarity representation under study is referred to as a **corpus graph** and aims to display the melodic similarities within an entire tune collection (a corpus), as opposed to a **tune graph**, which just display near neighbours for a single tune. However the underlying technology is effectively the same.

The idea is that, given a collection of tunes and a melodic similarity measure which can compute pairwise similarity between tunes (e.g. [2]), it is possible to construct a complete proximity graph of the corpus. In this graph each vertex represents a tune and edge weights represent similarities between tunes: the greater the similarity the larger the edge weight. If a similarity threshold is applied so that an edge is only included in the graph if the two tunes it connects are sufficiently similar then a sparse proximity graph can be induced (the higher the threshold, the more sparse the graph).

In [1] the sparse proximity graph is sparsified further and processed to produce a collection of local tune graphs, one for each connected vertex in the graph. Here, since the interest is in visualising the corpus as a whole, no such processing need take place. However, as discussed in Section II(C), there is a need to connect all the vertices in the sparse graph.

A. Melodic similarity measure: multilevel recursive alignment

In this paper, the melodic similarity measure used is multilevel recursive sub-sequence alignment. Space precludes a full description but it is discussed in detail in [1]. Some additional enhancements, tested in [3], are also applied. The most important features and enhancements are:

- **Multilevel comparisons**. Each tune is coarsened by recursively removing non-stressed notes to create a hierarchy of tune representations. When two tunes are compared, the similarity is calculated at each level of the hierarchy and then aggregated across the levels. This means that tunes which may differ in minor details can still be considered sufficiently similar, in terms of stressed notes (i.e. at a coarser levels), to generate edges in the corpus graph. It also allows for significant improvements in computational cost as the similarity calculation can be abandoned for tunes which have little or no similarity at the coarse (computationally cheap) levels.
- **Recursive Sub-Sequence Alignment**. A popular class of melodic similarity measures use *local alignment* algorithms, e.g. [2], applied to sequences of pitches or intervals. In general such algorithms have some ability to match inexact sub-sequences but this is limited, particularly when the sequences are long. To extend their effectiveness, recursive sub-sequence alignment was introduced in [1]. That paper and the follow-on, [3], also concluded that an effective scheme is a recursive version of the longest common substring algorithm (a special case of local alignment which only produces exact matches but which is faster to compute and requires less memory) and so that algorithm is used here.
- **Biased Recursive Alignment Scoring.** An issue that became apparent when using recursive alignment, is that just adding all the scores together makes no distinction

between one long aligned sequence and several shorter ones, [1]. To address this, the similarity measure is biased towards longer aligned sub-sequences by taking the 2-norm (square root of the sum of squares) of the alignment scores found by the recursive local alignment. This was demonstrated in [3] to significantly improve the accuracy of the measure.

Aside: It should be stressed that the ideas discussed here are generic: in principle, any melodic similarity measure, S(X,Y), which measures the similarity between a pair of tunes X and Y, can be used. However, obviously the results and visualisations will depend on the quality and features of the measure.

B. Matching threshold

Neglecting the multilevel framework for now, this similarity measure, S(X,Y), induces a complete weighted graph on a dataset, where the edge weight between each pair of melodies is given by the similarity. Subsequently, when the graphs are displayed, edge thickness is shown in proportion to the weight with very similar vertices joined by thick edges and not so similar ones by thin edges.

As mentioned, since most edges in the graph will have very small weights it makes sense to restrict the graph to include only edges for tunes which are reasonably close matches. This restricted graph is referred to henceforth as the **corpus graph**, (also referred to as the fundamental proximity graph in [1]).

This restriction could be achieved in a variety of ways but here it is assessed by a **matching threshold**, T, and edges between melodies X and Y are only included if they match across at least some proportion T of their length. Specifically an edge between vertices V_x and V_y is only included if

$$S(X,Y) \geq \text{average}(\text{length}(X), \text{length}(Y)) * T$$

Following [1] values for T in the experiments are 1/4, 1/6 and 1/8; the former value considerably restricts the number of edges whilst the latter is fairly inclusive and can allow a lot of false positive matches.

Note: the use of *biased* alignment scoring, mentioned above, does obscure what these fractions imply exactly, as it is no longer a case of adding up all the recursively aligned scores. To explore this further consider that a large proportion of melodies in the dataset are 32-bar tunes in an AABB format. This is very typical in western European folk music and usually means that the tune is written as 16 measures, as A and B sections, with repeat markers at the end of each section. For a reel in common time this would be quantised as 8 eighth notes per measure or a total of 16 x 8 = 128 notes (strictly speaking 127 intervals).

If T is set to 1/4 then, for an edge to be included between them, two tunes would need to match exactly across one a quarter of the tune (4 measures or 32 notes). Alternatively, they could match across four segments, each two measures (16 notes) long (in this case $S(X,Y) = \sqrt{16^2 + 16^2 + 16^2 + 16^2} = \sqrt{1{,}024} = 32$); in other words in this case they need to match over a total of 64 notes or half the tune, even though T = 1/4.

C. Connecting the corpus graph

As will be seen in the results section, it became apparent that the choice of similarity threshold can have an impact on whether it is even possible to visualise the corpus graph. If the threshold is too low, e.g. Fig. 5(l), then the graph will not be sparse enough to produce a successful visualisation as, typically, too many edges obscure any valuable information that could be gained from such a visualisation, [4].

However, higher values of the threshold typically generate a corpus graph with many *disconnected components* (subgraphs that are not connected by any edges) and often large numbers of *isolated vertices* (vertices with no incident edges – i.e. tunes that are not sufficiently similar to any other tune in the corpus to generate an edge).

This presents a problem for the investigation discussed here. One option is simply to visualise the largest connected component (i.e. the one with the most vertices) but in many cases this component may represent only a very small portion of the dataset. Alternatively, there are published schemes to lay out disconnected components: for example Monserrat *et al.* discuss an algorithm which regards each component as a super-vertex, connects all the super-vertices together in a chain-like structure and then uses a force-directed placement algorithm to create layouts both at the macro level (of the components) and at the micro level (within each component), [5].

Unfortunately, in the context of a proximity graph, such a scheme potentially discards a lot of valuable information: although two vertices may not be connected because the corresponding tunes are not sufficiently similar, nonetheless it is possible to measure the similarity between them and use that to inform the visualisation.

Accordingly the following algorithm has been implemented:

1. The corpus graph G(V, E) is constructed as discussed, with edges in E only included between pairs of vertices if the corresponding tunes are sufficiently similar.
2. The connected components are evaluated and all vertices are marked with a flag indicating which component they belong to.
3. A list is compiled of all those excluded edges which run between pairs of vertices in different components (i.e. with different flags).
4. The list is sorted in order of decreasing edge weight (i.e. decreasing similarity) and then the first edge in the list is added to the graph, but with its weight set to zero.
5. The process repeats from step 2 until the graph is fully connected. After the first iteration the edge list is already compiled so step 3 just involves removing edges between vertices in the same (newly connected) component.

It is easy to see that this scheme adds nothing to the total edge weight in the graph, but increases the number of edges by $|C| - 1$ where $|C|$ is the number of disconnected components in the original proximity graph.

The sorting of the edge list prioritises how the disconnected components are connected up. Obviously a disconnected component may have many vertices which are not quite similar enough to be connected to other components in the original

proximity graph and probably only one of these will end up being connected. However, since the visualisation is primarily about visualising similar tunes the inclusion of these additional edges is really just to enable the layout of the disconnected components in a sensible fashion.

Setting zero edge weights is important for visualisation: since edge weights influence vertex placement, a zero weight edge will have minimal impact on the graph layout but will mean that the two insufficiently similar vertices that it connects are positioned as close together as possible. Furthermore zero-weight edges are not displayed in the visualisations.

Note: connecting the graph up this way is computationally expensive. For a graph with thousands of components there can be millions of inter-component edges which need to be assessed. As a consequence the experiments are limited to graphs with a maximum of ~ 5,000 vertices by restricting the number of tunes.

D. Visualisation: multilevel force-directed placement

Having constructed the connected corpus graph, it can be visualised using multilevel force-directed placement (FDP), a heuristic method for drawing graphs which uses a multilevel framework combined with an FDP algorithm. FDP is a common technique for computing a layout of graph vertices, e.g. [5], which models edges as springs (that push / pull vertices apart) and includes global repulsive forces to untangle the graph.

FDP is unable to untangle large-scale structures and limited to graphs of a few hundred vertices. Hence multilevel schemes were introduced both to accelerate the computation and, more importantly, impart a global quality to the drawing, [6]. The multilevel technique matches and coalesces pairs of adjacent vertices to define a new graph and is repeated recursively to create a hierarchy of increasingly coarse graphs. The coarsest graph is then given an initial random layout and the layout is refined iteratively with FDP and recursively extended to all the graphs starting with the coarsest and ending with the original.

III. RESULTS

A. Annotated datasets

The initial investigation explores two small datasets known to contain many related tunes and which have been annotated manually to indicate similar melodies, specifically those belonging to the same tune family.

The first of these datasets is the Annotated Corpus of the Meertens Tune Collection [1] (MTC), version 2.0.1, [7]. This contains 360 Dutch folk melodies, each identified by experts as belonging to one of 26 tune families.

The second dataset (Morris) contains English morris dance tunes taken from the Morris Ring website[2]. Since morris music has many (approx. 35) related traditions, each typically associated with a village, there are several tunes found in more than one tradition, but each tradition typically has a different variant of the tune. This dataset therefore contains 368 tunes which have been manually identified as belonging one of 113 tune families (in many cases the tune family can be derived from the tune title or song words)[3].

Table I shows the corpus graph characteristics for both datasets using three different values for the matching threshold, specifically T = 1/4, 1/6 and 1/8. The table shows |V|, the number of vertices (fixed), |E| the number of edges and |C| the number of disconnected components. It also shows the maximum, d_{max}, and average, d_{avg}, degree of the vertices. The minimum degree is not shown as in all cases there are isolated vertices (often many of them) of degree 0.

Characteristics for the connected corpus graph, G′(V, E′), are not shown but most can be derived. The number of components |C′| is 1; the number of edges, |E′|, is just |E| + |C| – 1, so the average degree, d'_{avg}, is given by 2 (|E| + |C| – 1) / |V|.

TABLE I. ANNOTATED DATASET CHARACTERISTICS

Dataset	T	|V|	|E|	|C|	d_{max}	d_{avg}
MTC	1/8	360	981	85	19	5.45
MTC	1/6	360	833	117	16	4.63
MTC	1/4	360	501	157	13	2.78
Morris	1/8	368	667	124	18	3.62
Morris	1/6	368	450	184	17	2.45
Morris	1/4	368	199	268	15	1.08

The graph layouts are calculated using the multilevel force-directed placement algorithm discussed in section II but, since the datasets are annotated, a more informative visualisation is possible by assigning each tune family to a different colour.

Fig. 1 and Fig. 2(a) show the three graphs produced for the MTC dataset. As can be seen in Fig. 1(l) for T = 1/8, the methodology has done well at disambiguating the tune families: generally vertices of the same colour (in the same family) are clustered closely together even if no visible edge connects them (recall that zero-weight edges, which are not shown, are used to connect the graph and connect tunes to their most similar neighbour even though the two tunes are not considered sufficiently similar to have that similarity represented in the graph). Furthermore there are few edges between tunes in different families (i.e. "cut" edges between vertices of a different colour). However, there are some confusing edge crossings (centre left of the image) and so raising the threshold to exclude the more tenuous similarities can be helpful.

Fig. 1(r) for T = 1/4 demonstrates this. With a threshold this high the graph is much sparser (501 edges as compared with 981 previously) and the tune families are generally very well-spaced. However, this graph contains rather a lot of isolated vertices and some tune families appear to have no inter-family edges at all.

The compromise threshold of T = 1/6, shown in Fig. 2(l), resolves both difficulties: generally the tune families are well-spaced but there are not too many isolated vertices (although note that isolated vertices are an inevitable consequence of diversity within the dataset – they just express the fact that some tunes are not closely related to any others).

Fig. 2(r) shows the Morris dataset at the same threshold of 1/6. This is a rather more complicated picture as there are many

[1] http://www.liederenbank.nl/mtc/index.php

[2] https://themorrisring.org/music/handbook-morris-dances

[3] The allocation of tunes to families can be downloaded from the author's website http://chriswalshaw.co.uk/research/

more tune families (113 rather than the 26 of MTC) and some of those "families" only have 1 member. It is hard to find 113 distinct colours and so most are duplicated (currently the graph drawing software provides a maximum of 64 colours). Nonetheless the methodology has achieved good separation between tune family clusters, without too many isolated vertices and hence a graph layout that is relatively easy to navigate.

B. *Large datasets*

The next results consider larger datasets from 4 locations:

- The Session[4] (TheSession), a community site which hosts a large collection of Irish traditional tunes (~30,000).
- The Traditional Tune Archive[5] (TuneArch), a carefully curated archive of "North American, British and Irish traditional instrumental music" (~32,000 tunes).
- The Village Music Project[6] (VMP), another community project containing "traditional social dance music of England" mostly transcribed from 18th & 19th century manuscripts. At the time of writing the site hosts around 7,000 tunes but here a fixed subset of 5,638 used in previous papers, e.g. [1], is tested.
- An abc version of the Essen Folk Song Database[7] (Essen), containing "8000 European and Chinese folk songs".

Table II shows the corpus graph characteristics for all 4 datasets in the same format as Table I. As mentioned in section II.C, because of limitations in the graph connecting algorithm the maximum size of graph that can be easily constructed is limited to around 5,000 vertices and so, with the exception of the VMP collection, the datasets have all been restricted to this size by using only the first 5,000 tunes.

TABLE II. LARGE DATASET CHARACTERISTICS

Dataset	T	|V|	|E|	|C|	d_{max}	d_{avg}
TheSession	1/8	5,000	5,121	2,649	94	2.05
TheSession	1/6	5,000	2,847	3,312	12	1.14
TheSession	1/4	5,000	1,645	3,812	9	0.66
TuneArch	1/8	5,000	5,407	2,927	103	2.16
TuneArch	1/6	5,000	1,833	3,792	15	0.73
TuneArch	1/4	5,000	1,100	4,209	10	0.44
VMP	1/8	5,638	20,361	2,053	202	7.22
VMP	1/6	5,638	5,317	3,496	41	1.89
VMP	1/4	5,638	3,140	4,199	16	1.11
Essen	1/8	5,000	190,768	1,621	905	76.31
Essen	1/6	5,000	30,881	2,062	349	12.35
Essen	1/4	5,000	1,936	3,779	27	0.77

What is immediately clear is that there is a wide range of values for d_{max} & d_{avg}, the maximum & average degree of the vertices. This suggests that some datasets contain significantly more similarities than others. The first two datasets, TheSession & TuneArch, are relatively similar, but VMP is denser and Essen significantly more so.

Following manual investigation of a small number of matches a possible reason for the density of the Essen corpus graphs is that the tunes (mostly songs) tend to be very short in length (often just 8 measures). Therefore it is much easier to find a match across 1/4 of the tune (particularly at the end of a phrase where a single note may be held for the whole measure). This is backed up by the observation that the density of the graph tails off rapidly as the threshold is increased – much more rapidly than for the other datasets.

Fig. 3 uses the compromise threshold of 1/6 suggested above to visualise TheSession (l) and TuneArch (r) corpus graphs. Since the datasets are not annotated it is not possible to colour the vertices to indicate tune families. Nonetheless many small clusters are visible and indicate likely families.

Other identifiable structures include **super-connectors** (isolated vertices surrounded by sunflower like structures of other isolated vertices) and **weak linkage** (long, lightly-weighted edges which indicate loose connections between different subgraphs).

The super-connectors represent tunes which, although not sufficiently similar to any other tune to be connected by an edge, are passingly similar to many tunes. Investigation of some examples have suggested that this is because the super-connectors are very long in length.

The weak linkage is just a product of the layout algorithm: lightly weighted edges which join relatively coherent subgraphs tend to become very stretched when using force-directed placement because the attractive spring in the edge is not sufficient to overcome the global repulsive forces between the vertices in each subgraph which push the subgraphs apart.

Fig. 4(l) shows a similar structures for the VMP corpus graph although, as the graph is denser, there are more clusters and a tangle (centre-left) which the methodology has not managed to resolve fully.

Although none of these datasets are annotated, it is still possible to partition the graph in a number of ways and Fig. 4(r) shows one such attempt. Here the colours represent meters (4/4, 2/4, 6/8, etc) and clearly shows that although some tunes in different meters can be regarded as similar to each other (particularly using the multilevel similarity measure which may remove enough of the non-stressed notes to reveal related coarse structures), in some subgraphs particular meters dominate. For example most of the vertices in the upper left subgraph are "salmon" coloured (in this case indicating 6/8 jigs) whilst those top right are either "blue" (2/2) or "mediumvoiletred" (4/4).

Finally Fig. 5(l) shows the visualisation of the Essen dataset for T = 1/6. Here it is clear to see that the graph is almost certainly too dense for a layout to be computed for reasons discussed above. Fig. 5(r) therefore shows the graph for T = 1/4, revealing a few clusters, some super-connectors and weak linkage, although there is still a tangle, upper right, suggesting a large group of tunes that have some similarities between them.

IV. DISCUSSION, CONCLUSIONS AND FURTHER WORK

The results indicate that there are structures that become apparent when visualised using the corpus graph methodology discussed here. How much these structures reveal is debateable,

[4] https://thesession.org/

[5] http://www.tunearch.org/wiki/TTA

[6] http://www.village-music-project.org.uk/

[7] https://ifdo.ca/~seymour/runabc/esac/esacdatabase.html

but the differences between, say, the Essen corpus graph with T = 1/4 (Fig. 5(r)) and the TheSession and TuneArch graphs with T = 1/6 (Fig. 3), demonstrate very clear variances in the similarities within each dataset, despite comparable average vertex degrees (0.77, 1.14 and 0.73 respectively).

However, the results do suggest that further work into developing an *interactive* visual exploration tool, with user-driven zoom features and including score rendering and MIDI playing facilities, could be worthwhile. Moreover, as suggested in [1], the threshold, T, should be a variable parameter which the user could adjust according to their needs.

ACKNOWLEDGMENT

The author would like to thank Bob Sturm & Oded Ben-Tal for stimulating discussions in the development of this paper.

REFERENCES

[1] C. Walshaw, "Constructing Proximity Graphs To Explore Similarities in Large-Scale Melodic Datasets," in *6th Intl Workshop on Folk Music Analysis*, 2016.

[2] B. Janssen, P. van Kranenburg, and A. Volk, "Finding occurrences of melodic segments in folk songs employing symbolic similarity measures," *J. New Music Res.*, p. (to appear), 2017.

[3] C. Walshaw, "Tune Classification using Multilevel Recursive Local Alignment Algorithms," in *7th Intl Workshop on Folk Music Analysis*, 2017.

[4] C. Walshaw, "A Multilevel Algorithm for Force-Directed Graph-Drawing," *J. Graph Algorithms Appl.*, vol. 7, no. 3, pp. 253–285, 2003.

[5] T. Monserrat, J. Pabico, and E. lbacea, "A Hybrid Graph-drawing Algorithm for Large, Naturally-clustered, Disconnected Graphs," *Asia Pacific J. Multidiscip. Res.*, vol. 2, no. 4, pp. 119–126, 2015.

[6] C. Walshaw, "A multilevel algorithm for force-directed graph-drawing," *J. Graph Algorithms Appl.*, vol. 7, no. 3, 2003.

[7] P. van Kranenburg, B. Janssen, and A. Volk, "The Meertens Tune Collections : The Annotated Corpus (MTC-ANN) Versions 1.1 and 2.0.1," 2016.

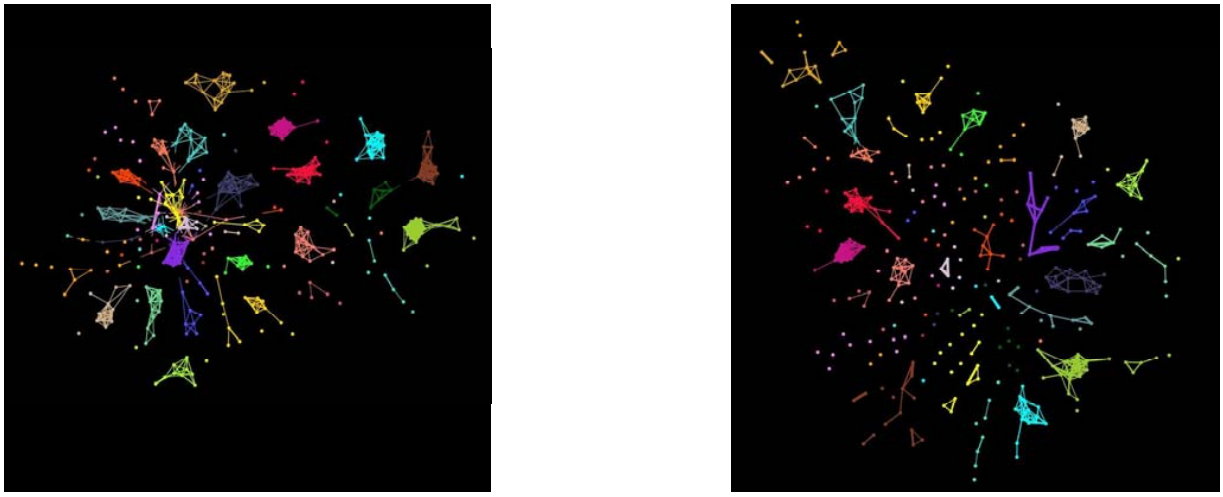

Fig. 1. Corpus graphs for the MTC dataset with T = 1/8 (left) and T = 1/4 (right)

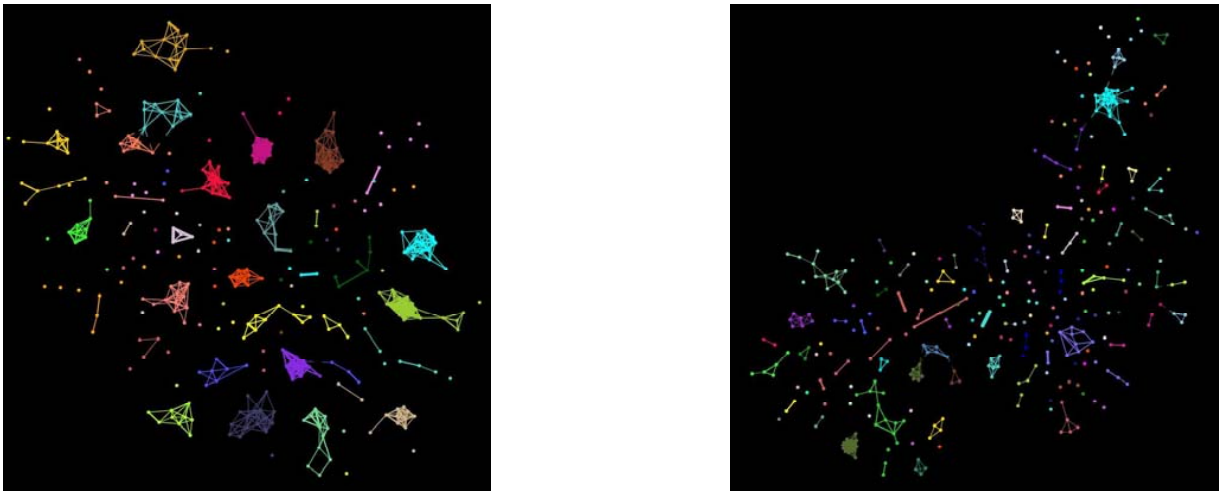

Fig. 2. Corpus graphs for MTC (left) & Morris (right) datasets with T = 1/6

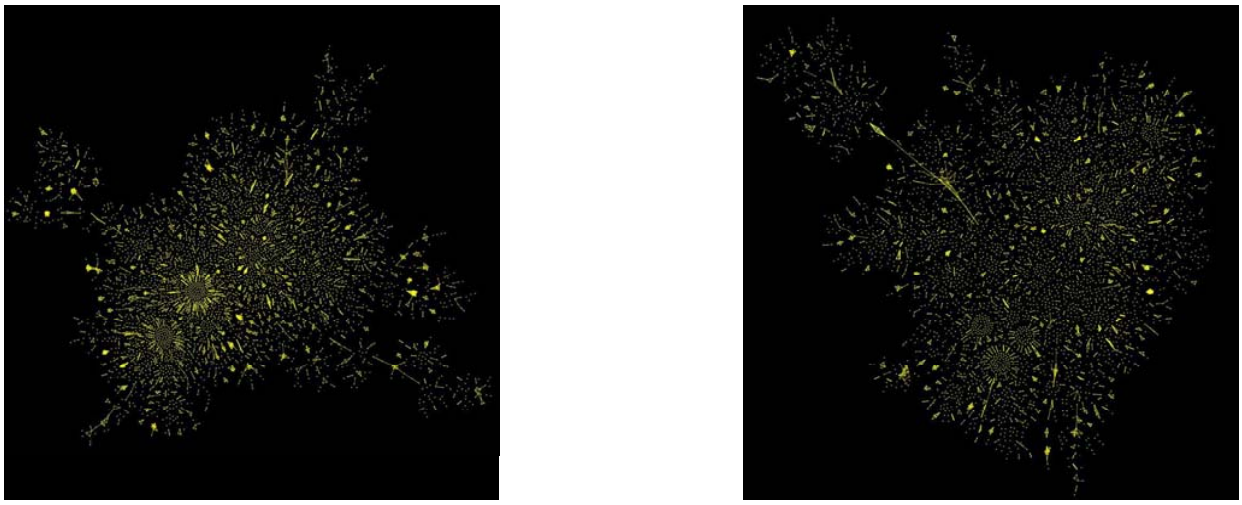

Fig. 3. Corpus graphs for TheSession (left) & TuneArch (right) datasets with T = 1/6

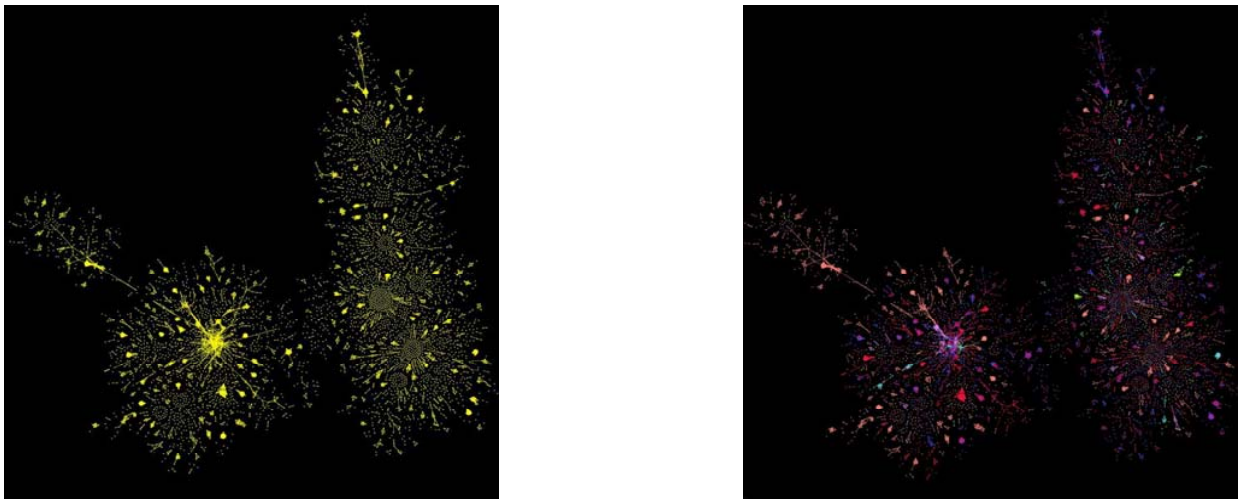

Fig. 4. Corpus graphs for the VMP dataset with T = 1/6 showing it unpartitioned (left) and partitioned by meter (right)

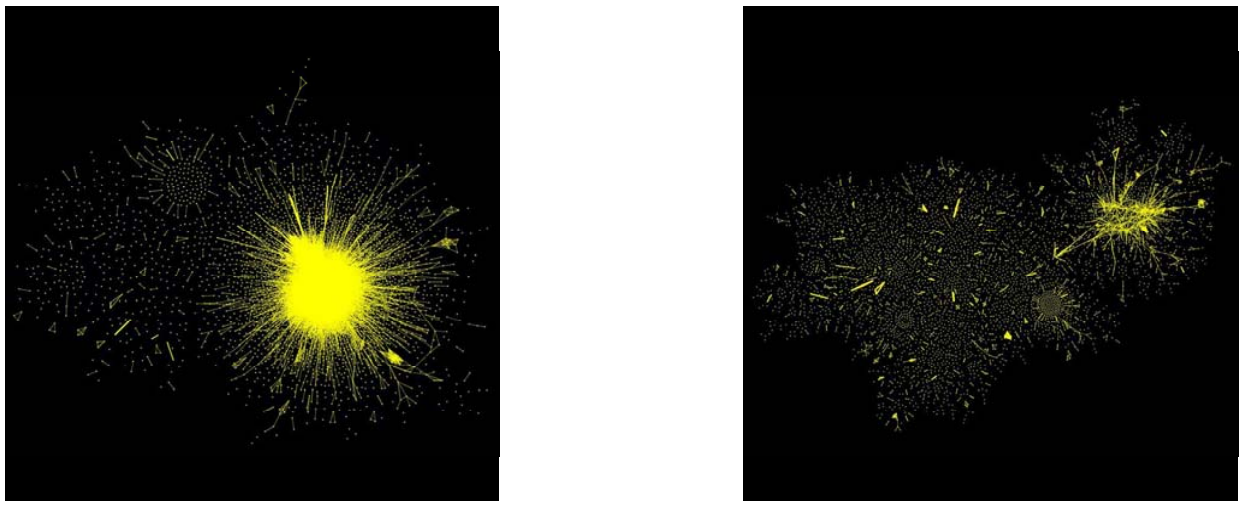

Fig. 5. Corpus graphs for the Essen dataset with T = 1/6 (left) and T = 1/4 (right)