情報実験第４
ビッグデータ

全体スケジュール

- 第１回 ガイダンス・環境構築
- 第２回 講師紹介,Python基礎
  - 文法、データ型
  - テーブルデータ操作(pandas),グラフ
- **第３回 テキストマイニング基礎①**
  - **形態素解析、係り受け解析**
- **第４回 テキストマイニング基礎② / 個人課題**
  - 文書類似度
- 第５回 個人課題
- 第６回 チーム作り / チーム課題設定
- 第７回～第９回 チーム課題
- 第１０回 チーム課題発表
- Extra(共起, 最新論文,word2vec）
- 第１2回 まとめ(課題フィードバック等）、環境クリーンアップ

前準備

- http://www.itpro.titech.ac.jp/exp4/からダウンロードした
  extraフォルダを
  書類/exp4bd/
  フォルダに格納

#フォルダのパス・名前は任意だが、以降の説明では上記パスを利用するので、適宜読み替えること

- グラフ可視化ツール「Gephi」を
  https://gephi.org
  からダウンロードしてインストール
- USBからentity_vectorフォルダを
  書類/exp4bd/
  フォルダにコピー

テキストマイニング
～共起 ～

文書の特徴把握

- 以前、単語の出現回数を調べることで
  文書の特徴を把握した
- 単語の出現回数だけでみるよりも、単語と単語の関連性も含めて見た方が内容の把握が簡単なのではないか？
  - 単語X,YのTFが高いとしても、XとYの関係を語っているのか、全く別の話題(例:1章がXに関する記述,2章がYに関する記述）なのかわからない
  - 何がどうした/なにがどうだ 等の名詞-動詞,名詞-形容詞の関連性

共起(Collocation)

- ある単語とある単語が同時に使われること
  例）「選挙」 という語 と
  「投票」「衆議院」「比例」などがよく一緒に使われる
- どこまでを「同時に使われた」とみなすか
  - N-gram:対象となるテキストの中で、連続するN個の表記単位(gram)の出現頻度
    例）「衆議院 選挙 の 投票 に 行った」 の「選挙」のN-Gram
    1-gram(unigram) ： 衆議院(1),の(1)
    2-gram(bigram) ： 衆議院(1),の(1),投票(1)
    3-gram(trigram) ： 衆議院(1),の(1),投票(1),に(1)
  - １文の中全て
    例）「衆議院 選挙 の 投票 に 行った」 の「選挙」の共起頻度
    衆議院(1),の(1),投票(1),に(1),行った(1)
  - 係り受けの係り元/係り先
    衆議院 - 選挙-の　　(衆議院,選挙)
    +投票に　　（選挙,投票）
    +行った　　（投票,行った）

※　実際は必要品詞の原型抽出後(衆議院/選挙/投票/行く)を元に計算することが多い

共起(Collocation)の利用例

Weblio**英語共起表現検索**
http://ejje.weblio.jp/concordance/

2語左の単語		1語左の単語		検索キーワード	1語右の単語		2語右の単語	
It	546	is	805	part	of	2365	the	1467
is	210	a	224		in	176	a	57
the	199	was	156		II	67	of	30
The	153	took	131		I	56	The	25
was	64	forms	106		is	42	part	24
it	51	became	90		One	31	his	22

unigram,
bigram
を表示

共起(Collocation)の利用例

共起語分析ツール （SEOツール）
https://coresysapp.net/c/cooccur-terms.php

共起語抽出ツール

「選挙」の共起語

共起語一覧　自社記事比較　共起ネットワーク

説明資料：共起語ツールを使ったコンテンツSEO

▼全ての共起語一覧

共起語	出現数	出現頻度	重要度
衆院選	14回	20%	3.4
当選	11回	15%	3
投票	11回	20%	2.7
日本	9回	15%	2.4
22日	6回	15%	1.6
10月	6回	15%	1.6
--非公開--	6回	15%	1.6

▼5位以内のページ40%以上が含む共起語

共起語	出現数	出現頻度	重要度
衆院選	14回	20%	3.4
当選	11回	15%	3
10月	6回	15%	2.7
22日	6回	15%	2.4
17年	6回	20%	1.6
開票	4回	15%	1.6

共起(Collocation)の利用例

共起語分析ツール （SEOツール）
https://coresysapp.net/c/cooccur-terms.php

共起語抽出ツール

「選挙」の共起語

共起語一覧　自社記事比較　共起ネットワーク

説明資料：共起語ツールを使ったコンテンツSEO

▼全ての共起語一覧

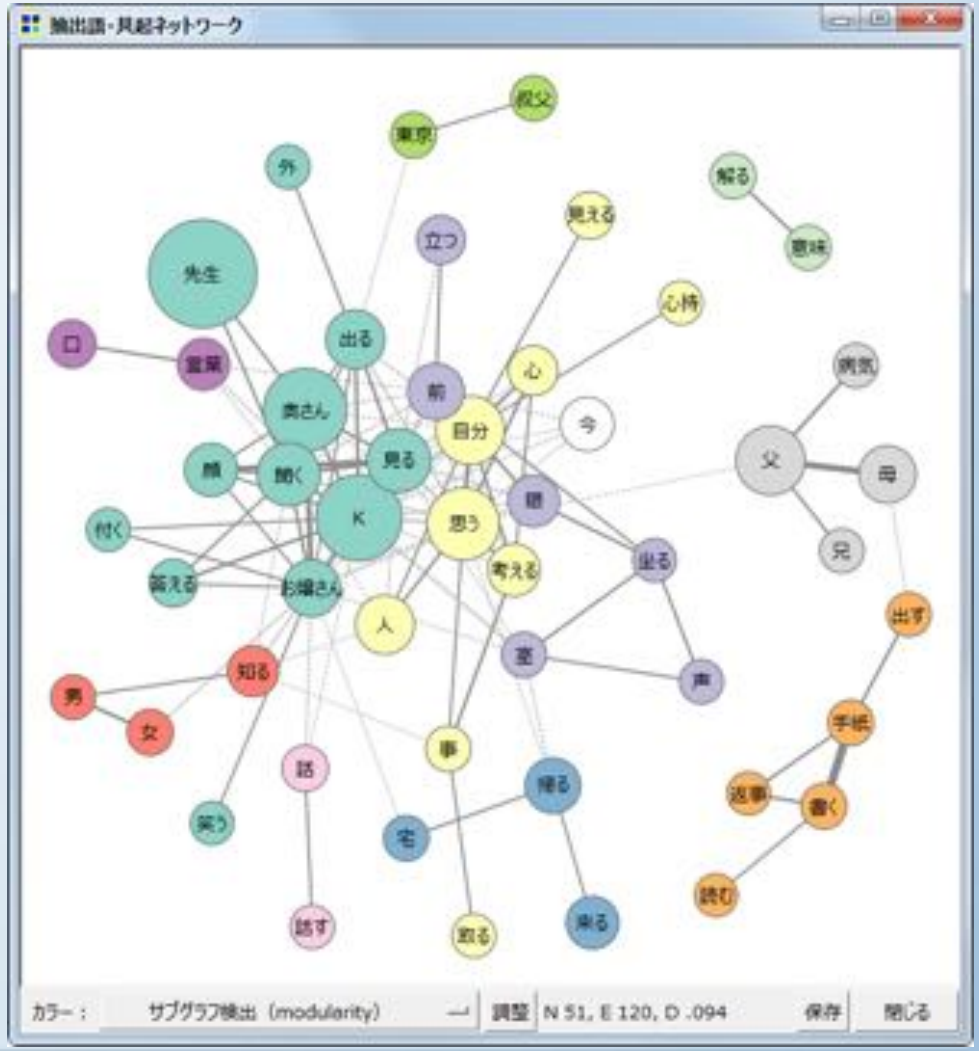
共起語	出現数	出現頻度	重要度
衆院選	14回	20%	3.4
当選	11回	15%	3
投票	11回	20%	2.7
日本	9回	15%	2.4
22日	6回	15%	1.6
10月	6回	15%	1.6
--非公開--	6回	15%	1.6

▼5位以内のページ40%以上が含む共起語

共起語	出現数	出現頻度	重要度
衆院選	14回	20%	3.4
当選	11回	15%	3
10月	6回	15%	2.7
22日	6回	15%	2.4
17年	6回	20%	1.6
開票	4回	15%	1.6

共起ネットワーク

- 単語をノードとして、共起語をエッジで結んで、ネットワーク図（graph）で表したもの



共起ネットワークをわかりやすくする工夫

- 単語の出現頻度（TF）等の単語特徴量によってノードの大きさやラベル文字サイズを大きくする
- 共起頻度の大きさによって、エッジの太さを太くしたり、色を濃くしたりする
- バネモデルを用いてレイアウトする
  （共起頻度が高い語ほど近くに配置）
- ネットワーク(graph)をクラスタリングし、クラスタ毎に色分け（→話題毎に色分け）

Let's try it

前回までに利用した形態素解析用の関数
（ getAozoraText ,parse2df）では、
「。」毎に改行を挿入（ getAozoraText ）
改行毎に形態素解析し、
'文番号' を付加した形態素解析結果の
DataFrameを返却
していた

→ 1文中の同時利用を共起とみなした共起頻度を
GroupBy,
itertools.combinations(),
collections.Counter
などを使えば簡単に算出できる

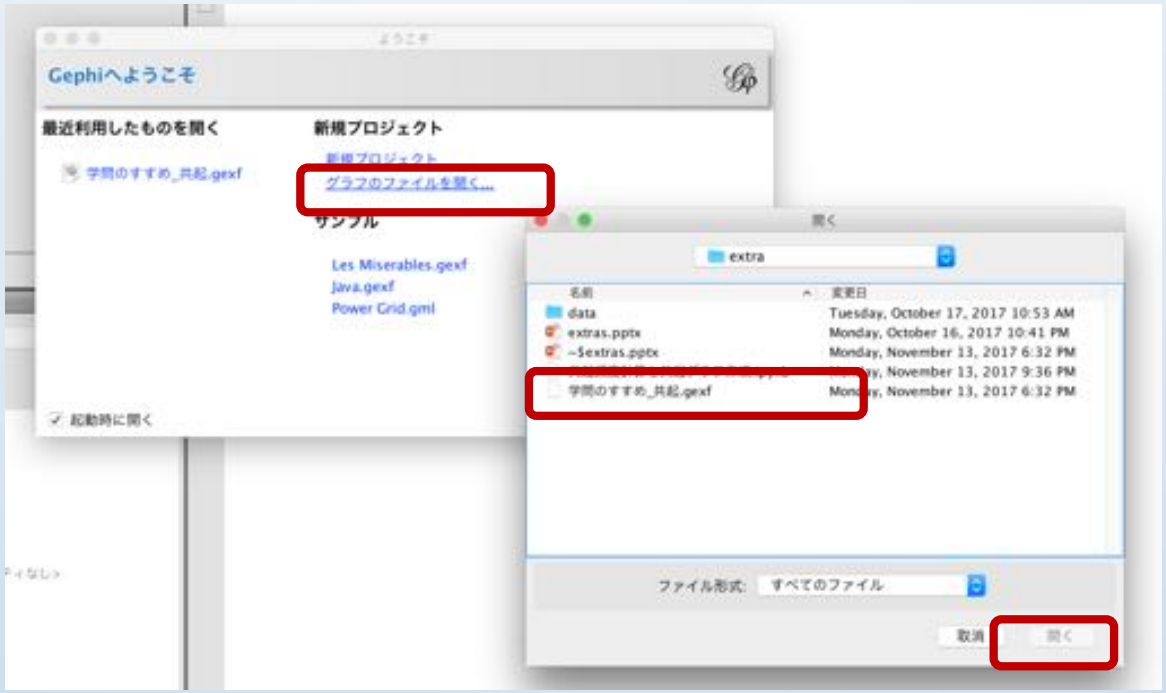
→ Graph構造の作成には
networkxライブラリを用いる

→詳細は
jupyter notebook で説明

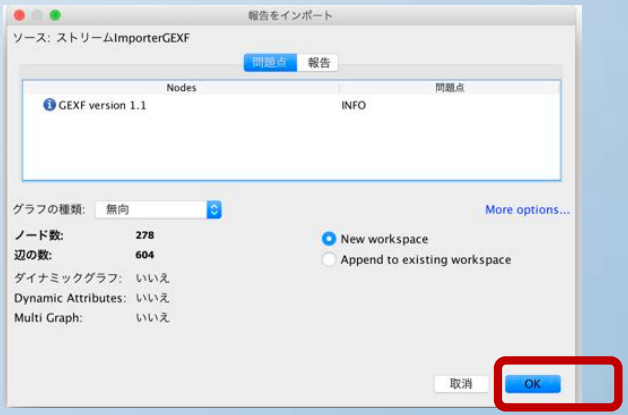
Gephiによるグラフ可視化

Let's try it

作成したGraph(gexf)ファイルをGephiで読み込む

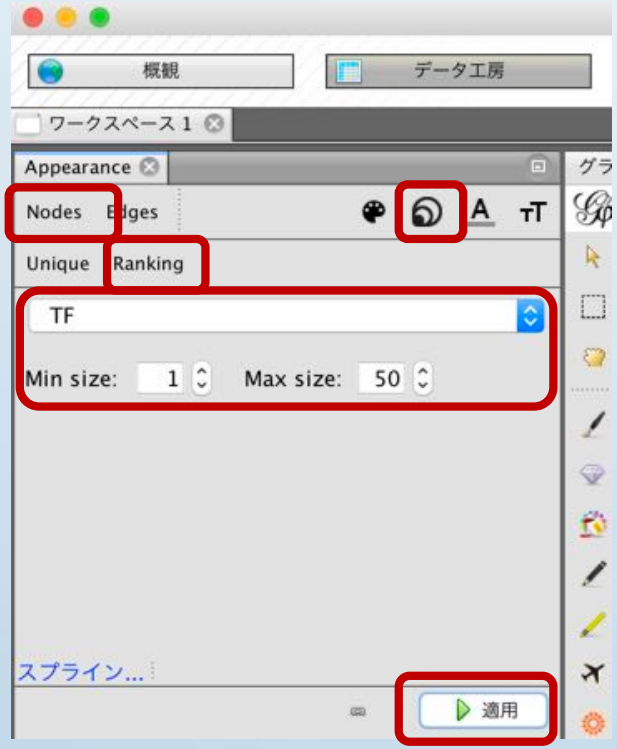


そのまま[OK]

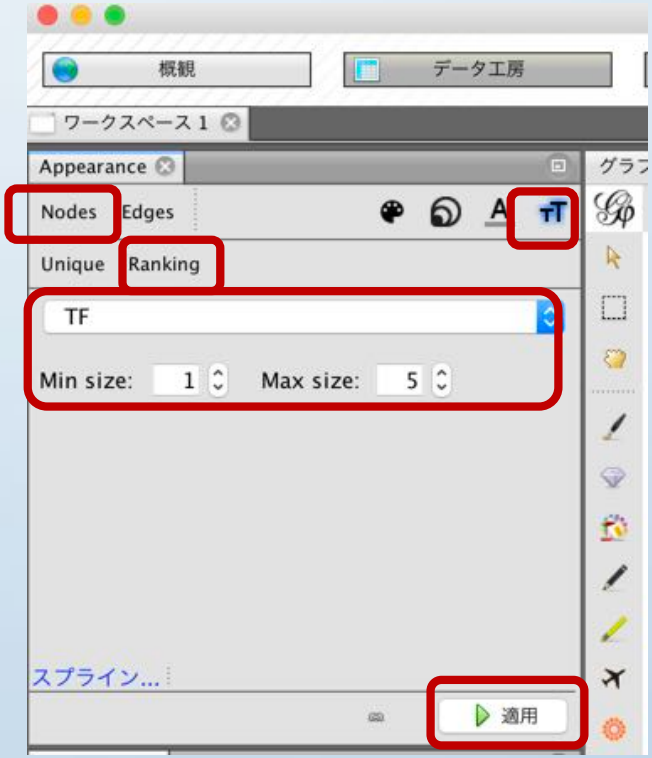


- 単語の出現頻度（TF）等の単語特徴量によってノードの大きさやラベル文字サイズを大きくする

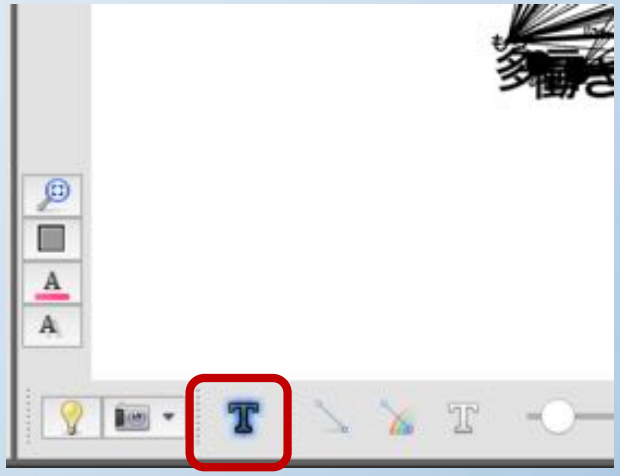
ノードサイズ



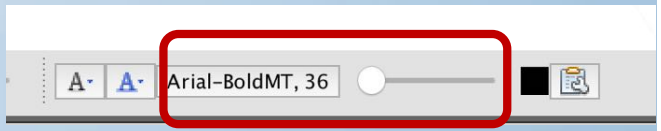
ラベルサイズ



ラベル表示有無

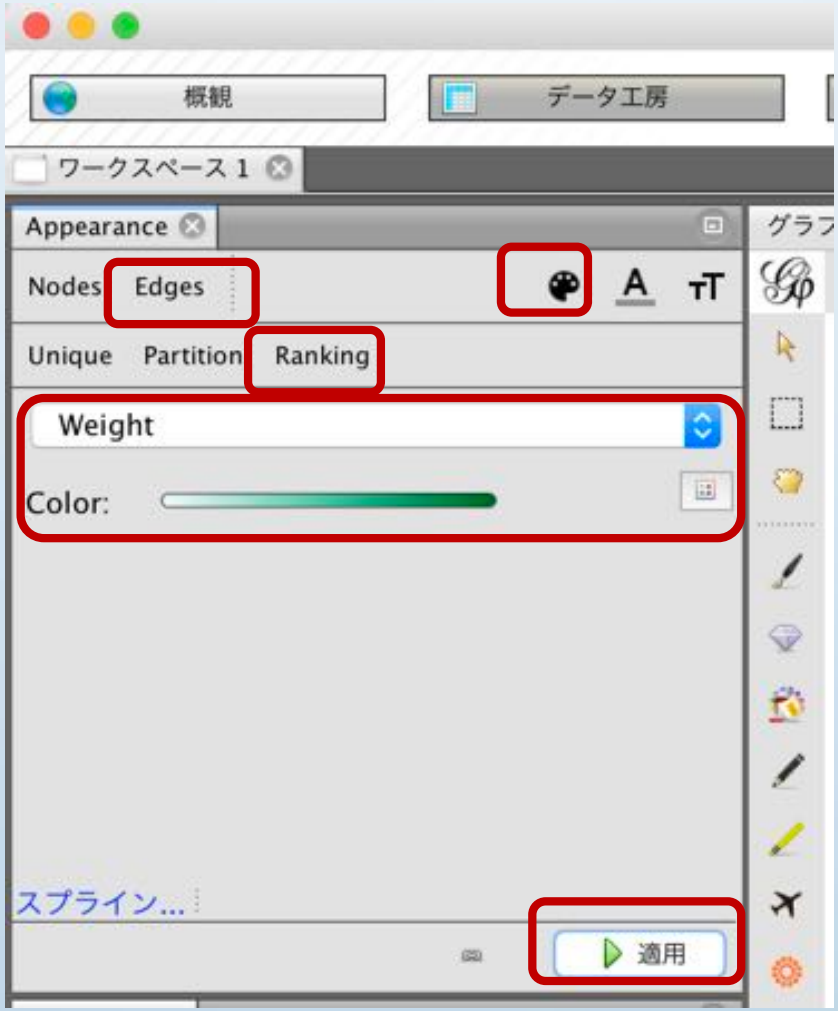


基本ラベルサイズ



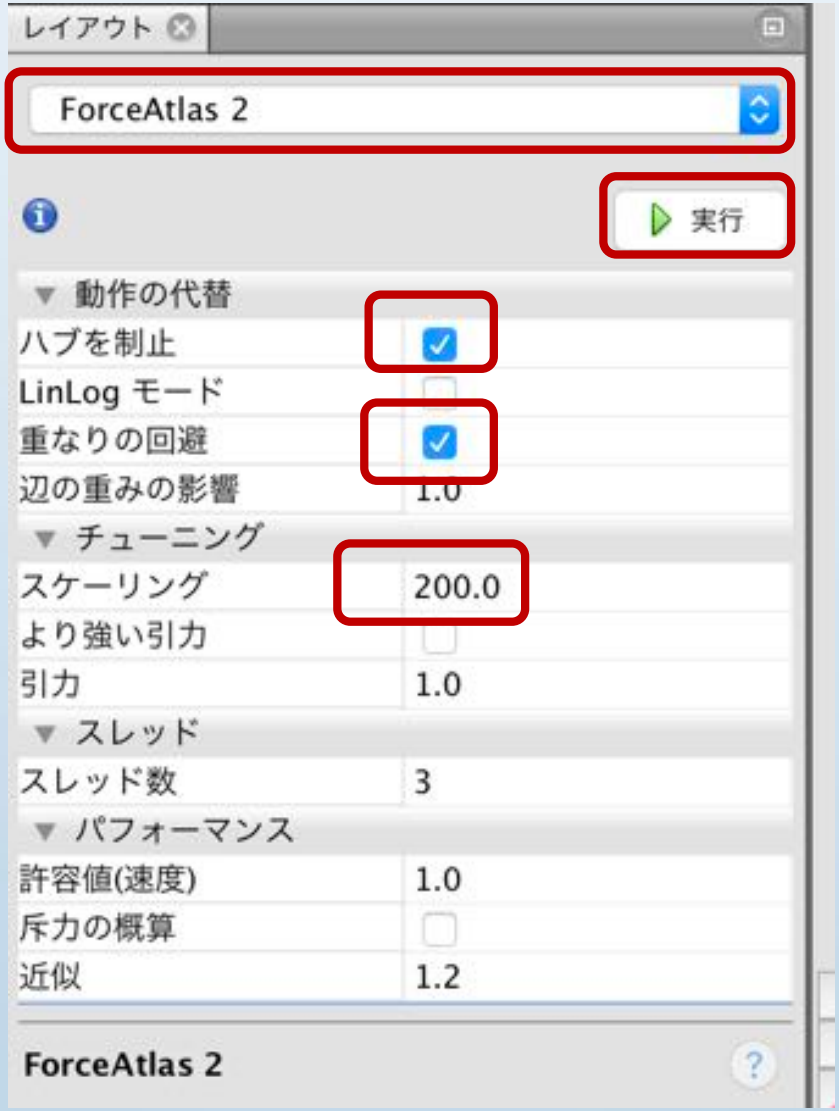
- 共起頻度の大きさによって、エッジの太さを太くしたり、色を濃くしたりする

エッジの濃さ



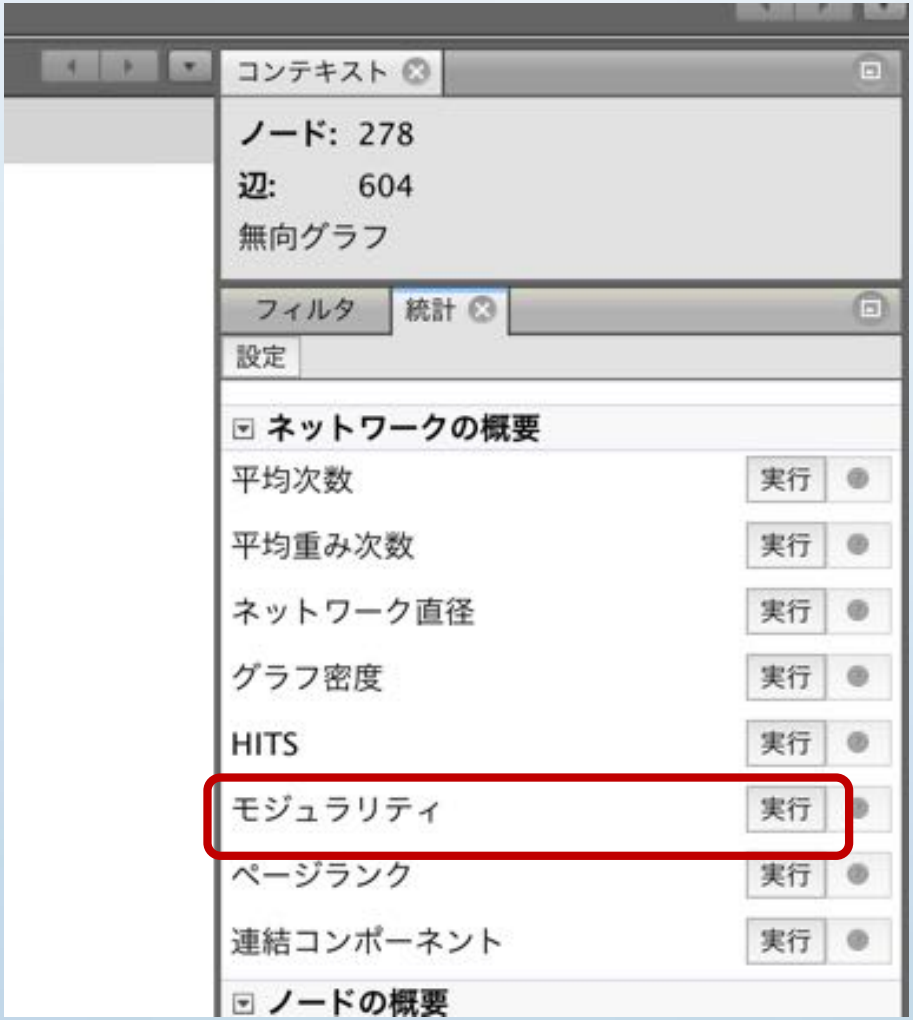
※　エッジの太さは
weight値で自動で決まる

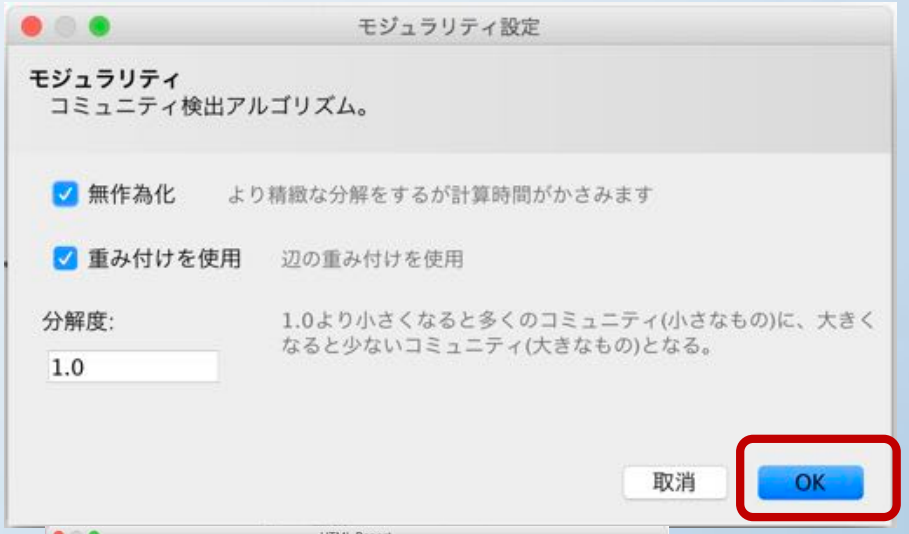
- バネモデルを用いてレイアウトする
  （共起頻度が高い語ほど近くに配置）

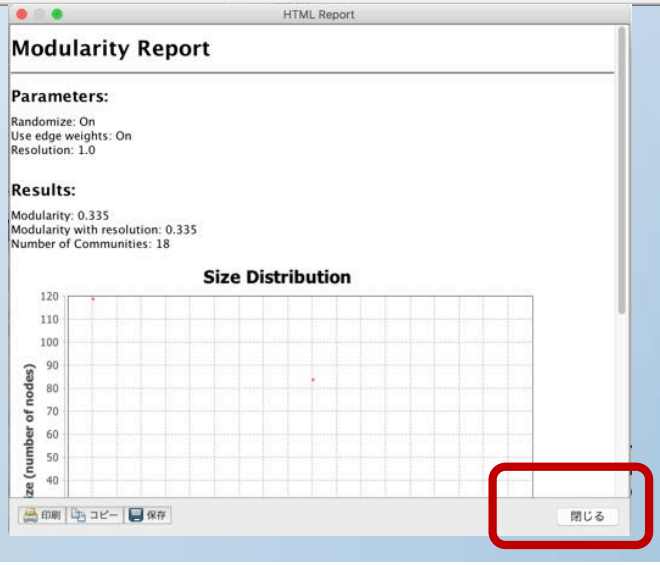


グラフの配置がある程度安定したら[停止]で処理を止める

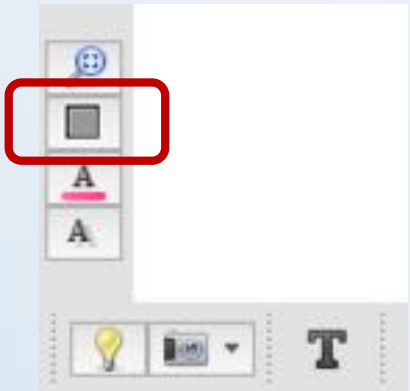
- ネットワーク(graph)をクラスタリング



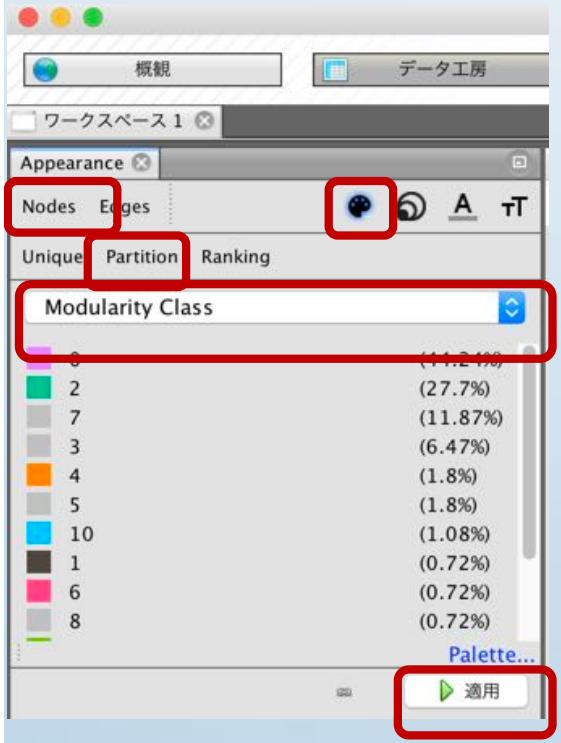




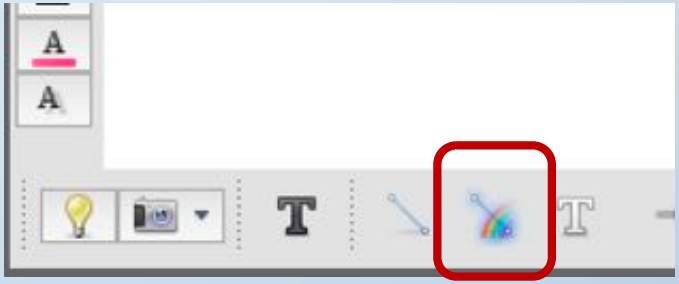
- クラスタ毎に色分け（→話題毎に色分け）



一旦、カラー(エッジに設定した色）をリセット

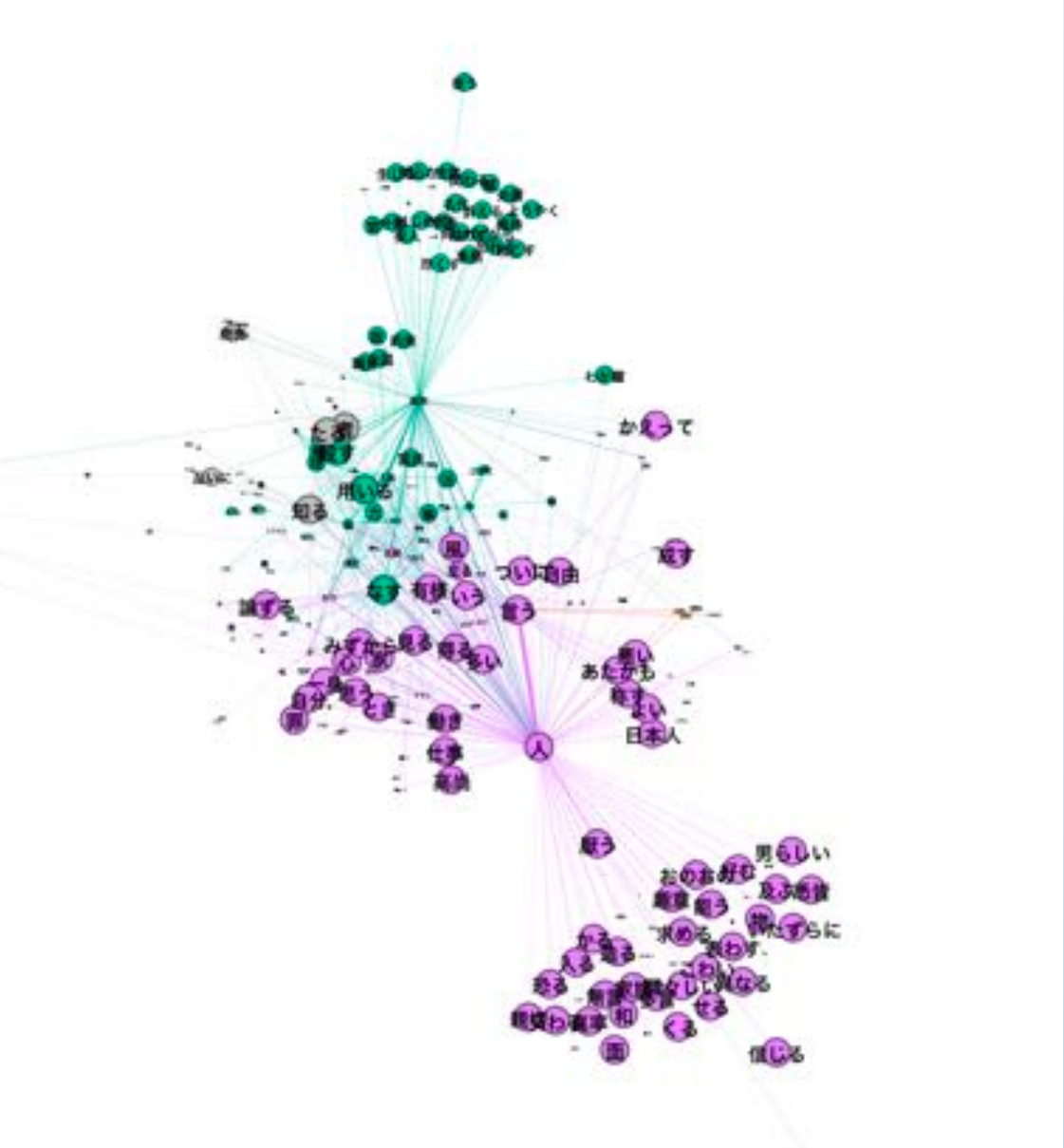


一旦、カラー(エッジに設定した色）をリセット



エッジをクラスタの色に合わせる

「学問のススメ」の共起グラフ

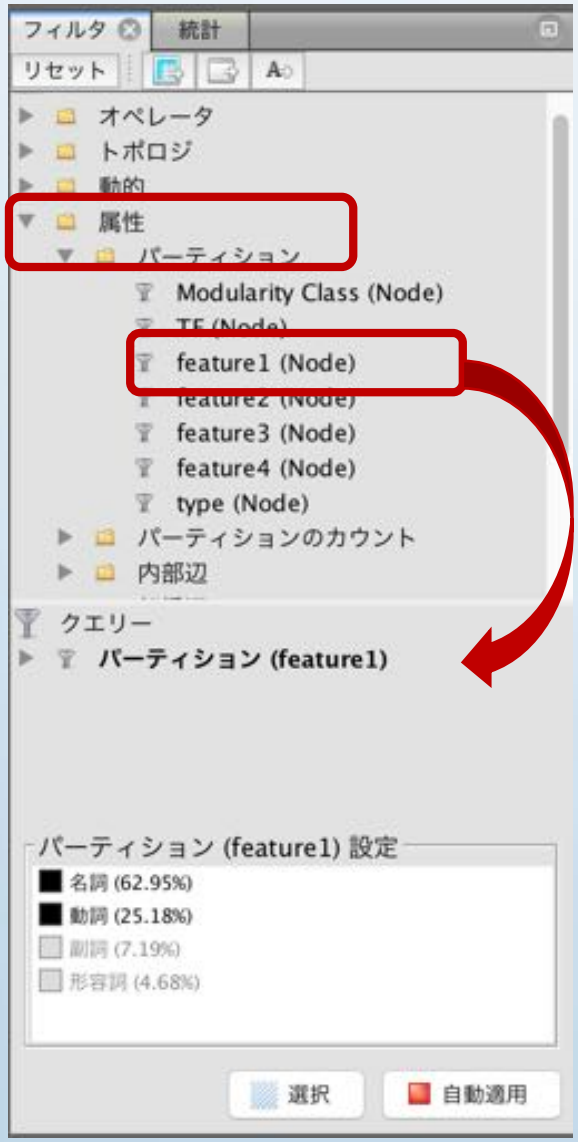


主な話題は二つ。
それぞれ「人」「政府」が中心
→「人」と「政府」について述べている

グラフ描画エリアでの操作（Mac）

- 拡大縮小： スクロール操作
- 視点移動： controlキーを押しながらDrag操作

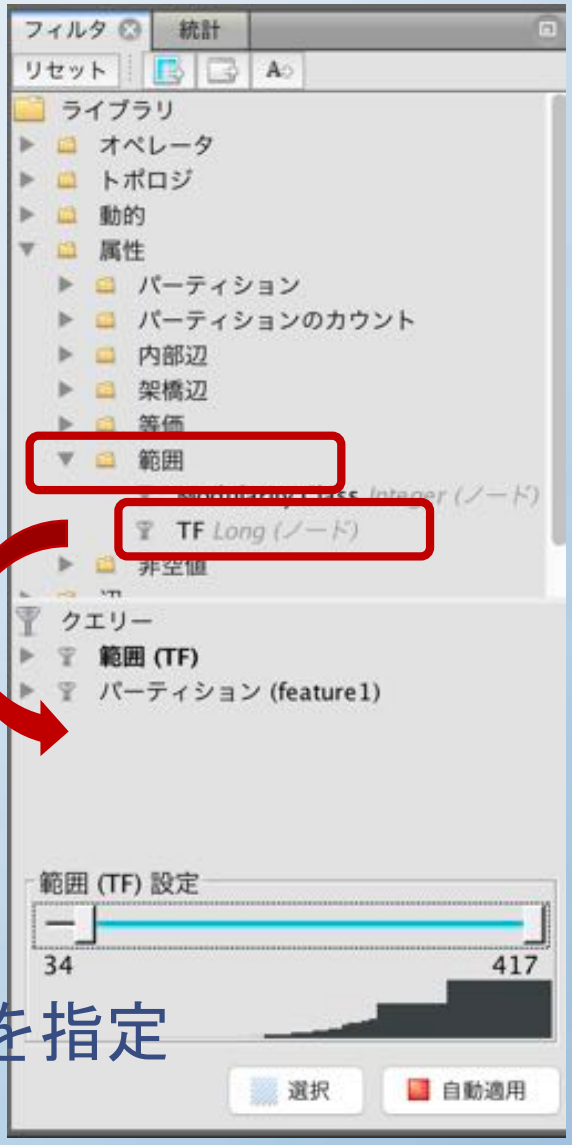
フィルタ



Drag&Drop

表示する
品詞を選択

Drag&Drop



表示する
TFの範囲を指定

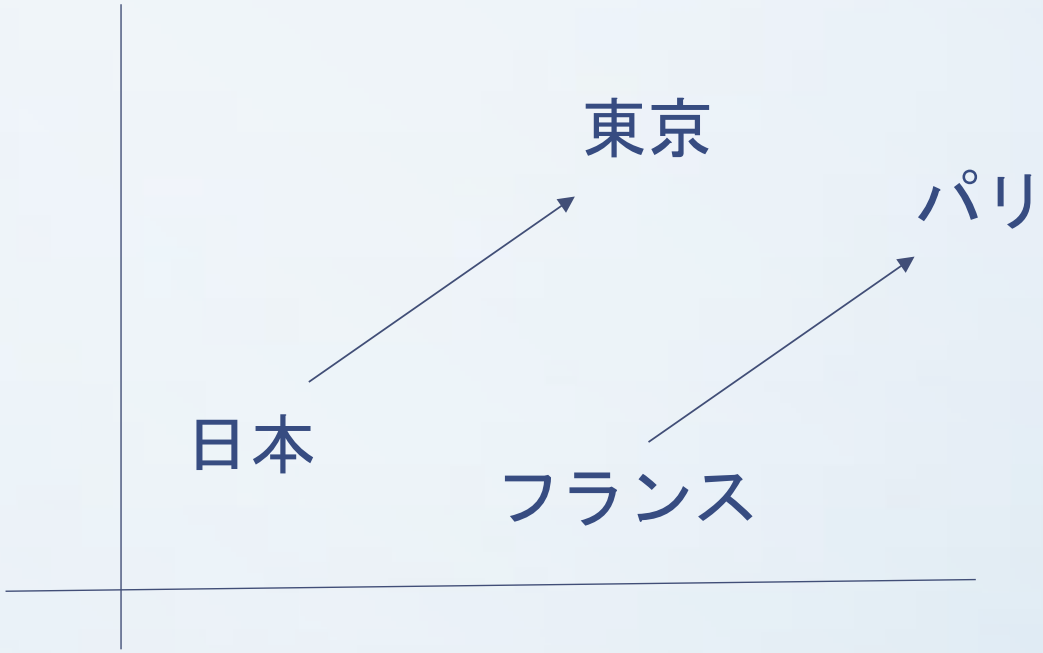
テキストマイニング
～word2vec ～

word2vec

- ざっくり言うと・・・
  共起関係をニューラルネットワークで学習したもの
- 単語の特徴量を200次元程度で表現
  （抽象的な意味を表現するようなベクトルになる）

- 意味の近い単語は、似た特徴ベクトルになる
- 単語 A,Bの位置関係(ベクトルの差）と
  単語C,Dの位置関係(ベクトルの差）が同じような意味を持つ

Let's try it



上記のようなベクトルになっていれば、
　「パリ」-「フランス」＝「東京」-「日本」
が成り立つ
→
　「パリ」-「フランス」+「日本」
を計算すると「東京」が導けるはず

→ jupyter notebook で試す