分散処理システム特論２　最終課題

17x3043 小菅一磨

1. **使用したデータセットおよび概要**

QiitaのサイトのWebAPIから取得した投稿情報。投稿についたタグを投稿毎にリストにして、txtファイルに保存したものをソースコードのファイルに含めてあります。QiitaのAPIでは様々な情報を提供していて、qiitaの投稿から、その投稿についたいいねの数やコメントも取得することができ、また何より、非常に多くのITエンジニアの方によってサポートされた掲示板なので、非常に有用性が高いと考え使用することにしました。

Qiitaにて投稿された最新の投稿を合計3000記事取得し、その投稿のタグ情報を取得しました。そして同じ投稿に含まれるタグに対してエッジを設け、その回数で重み付けを行いどのようなタグが同じ投稿に含まれるかをnetworkxのライブラリを用いて分析しました。興味があるネットワーク分析でnetworkxを使いたかったので、rdf情報ではないのですが、ネットワークのようなものを可視化できないかと思い、qiitaのwebAPIは元々知っていたので、その中からタグ情報を使い、I T分野のタグとなる単語の関係を分析しました。

1. **プログラムの説明**

以下にプログラムの簡単なアルゴリズムを示します。

➀Qiitaサイトから最新の投稿のタグ情報を抽出(3000記事分)

すでに出現していたタグの場合、そのタグの出現回数も計算。グラフには上位50のタグを使用するため回数によってフィルタリングを行いました。

↓

②同じ投稿に含まれていたタグの組み合わせに対して、エッジを追加

例：[“Python”,”機械学習”,”SVM”]の場合

(Python,機械学習),(Python,SVM),(機械学習,SVM)のエッジを作成

\*有効グラフではありません\*

↓

③この②を繰り返している間にすでにグラフオブジェクトに含まれているエッジの場合はそのエッジにつく重みを増やす。

↓

④グラフネットワークをmatplotlibで表示

プログラムに使用した主なライブラリ

requests(API取得のためのライブラリ)

json(json関係のライブラリ、APIからのデータはjsonだったのでそのjsonを整形するために使用)

collenctions(リスト内の出現回数を計算するためのものでリストや辞書の構造体の中身をいじるのに使用できる)

itertools(イテレータに関するライブラリでこちらもリスト内の出現回数を計算するために使用)

network(ネットワーク分析用のライブラリ)

matplotlib(networkxのグラフを描画するため)

プログラムに関して苦労した点がいくつかありました。

まず、APIデータの量の多さです。取得するのに非常に時間がかかりました。そのため一度実行して終わったと思ったらプログラムのエラーなどで最初からになることが何回かありました。またqiitaのAPIはアクセス制限があるので、しっかりとコードを書いてから実行しないと一回失敗すると次の実行まで時間を空けないといけないのが少し辛かったです。Qiitaから取得できるテキストがjsonだったので、それを整形するプログラムを書くのが結構苦労しました。

そしてあまりmatplotlibを使わないのでわからなかったのですが、描画するときにラベルが日本語の場合文字化けしてしまいそれを解消するのにかなり時間がかかって辛かったです。

1. **結果と考察**

以下に結果の一例を示します。

これはソースコードのフォルダにも入っているのですが、説明のために表示しました。

まずノードの大きさは、その単語の出現回数を表していて投稿についている回数が多いタグほどノードも大きくなっています。次にエッジの太さですが、同じ投稿に含まれている組み合わせほど太くなっています。例えば上部のSwiftとiOsのノードを繋いでいるエッジはiOsとflutterノードを繋ぐエッジよりも太くなっています。これは、SwiftとiOsの方が同じ投稿にタグとしてつけられた回数が多いことがわかります。そのように見てみると、妥当だと言えるのがpython分野で、pythonタグは、機械学習タグや、raspberrypiとのエッジの重みが強く、OS関連のmacやwindows、ubuntuに繋がりがあり、初心者タグは多くのタグとの繋がりが多いですが、特にフロントエンドのHTMLや開発で多く使われている、python、rubyとの繋がりが高いとみることができます。また、ノードの大きさから、現在の流行りのプログラミング言語はpythonと言えることがわかります。何回か実行してみたのですが、pythonのノードはどの実行でも同じような大きさが表示されたので妥当性も高いと思いました。

グラフ

自動的に生成された説明

また単語間の類似度を計算するのなどは文章からベクトルを作成してそのベクトルの類似度を計算するものが多いですが、それは文章が存在しないと学習できないのと比べて、単語の関連性であったら、こういったタグ情報だけでも、どういった単語が関連があるのかがわかるので、文書や単語の分析とは違った結果が得られたと思い非常に良い分析となりました。今回のこの、分析を応用すれば、短文を大量に集めてわかちがきを行い、文章ごとの単語の組み合わせにエッジをつければどういった単語に関連性があるのかが可視化できるのではないかと考えました。短文からの分析は自分の中では難しいと思っているので、こういった手法が使えるのではないかと思いました。ただ、その短文を集める方法が思いつかなかったのと、時間がなかったため今回は実装できませんでした。