AKADEMIA NAUK STOSOWANYCH W TARNOWIE

Wizualizacja formuł logicznych

Prowadzący: dr hab. inż. Radosław Klimek

Damian Ciura,
Stanisław Mendel,
Wiktor Warmuz

AKADEMIA NAUK STOSOWANYCH W TARNOWIE

Spis treści

Wstęp

Celem projektu była wizualizacja formuł logicznych, czyli reprezentacja graficzna struktury logicznych wyrażeń matematycznych lub symbolicznych, umożliwiająca łatwiejsze zrozumienie ich skomplikowanych relacji i interakcji.

Naszymi formułami były pliki typu CNF opisane poniżej, które zawierały formuły logiczne w postaci tzw. Koniunkcyjnej Normalnej Formy (CNF), co umożliwiało efektywne reprezentowanie złożonych wyrażeń logicznych za pomocą koniunkcji i dysjunkcji klauzul.

CNF to katalog danych, który zawiera przykłady plików przechowywanych przy użyciu formatu pliku CNF DIMACS. Ten format służy do definiowania wyrażenia boolowskiego, napisane w spójnej formie normalnej, którą można wykorzystać jako przykład problemu satysfakcji.

Problem satysfakcji dotyczy przypadku, w którym N boolean zmienne są używane do utworzenia wyrażenia boolowskiego obejmującego negację (**NIE**), połączenie (**I**) i rozłączenie (**LUB**). Problem w tym w celu ustalenia, czy istnieje przypisanie wartości do wartości logicznej zmienne, które sprawiają, że formuła jest prawdziwa. To coś w rodzaju próby przełącz kilka przełączników, aby znaleźć ustawienie, które włącza żarówkę.

Formuły te należało „przeliczyć” jedną z metod embedding, tj. przekształcić je w reprezentację numeryczną przy użyciu algorytmu embedding, takiego jak DeepWalk. Algorytm ten konwertuje struktury grafów, w tym formuły logiczne w postaci CNF, na wektory numeryczne zwane embeddingami. W efekcie każda formuła logiczna jest reprezentowana w przestrzeni o

mniejszej wymiarowości, co ułatwia analizę i wizualizację złożonych struktur logicznych. Proces ten pozwala na bardziej efektywną pracę z formułami logicznymi w kontekście analizy danych za pomocą technik embeddingu.

Metody embedding są używane do reprezentacji obiektów, danych lub informacji w przestrzeni o mniejszej wymiarowości, zwanej przestrzenią embeddingu, w taki sposób, aby zachować pewne właściwości tych obiektów. Główne cele korzystania z metod embeddingu obejmują:

- Redukcję Wymiarów:
  - Embedding pozwala na reprezentację danych w przestrzeni o mniejszej liczbie wymiarów niż oryginalne dane. To ułatwia analizę, wizualizację i manipulację dużymi i złożonymi zbiorami danych.
- Utrzymanie Relacji:
  - W przypadku metod embeddingu grafu, takich jak DeepWalk, celem jest zachowanie struktury grafu i relacji między wierzchołkami w przestrzeni embeddingu. W rezultacie podobne obiekty w oryginalnej przestrzeni są blisko siebie w przestrzeni embeddingu.
- Efektywne Uczenie:
  - Embedding ułatwia uczenie maszynowe, ponieważ reprezentacja numeryczna obiektów może być bardziej efektywnie przetwarzana przez modele uczenia maszynowego. Umożliwia to lepsze wykorzystanie algorytmów uczenia maszynowego i głębokiego uczenia.
- Wizualizacja:
  - Przestrzeń embeddingu umożliwia wizualizację złożonych struktur danych. Możemy łatwo przedstawiać i analizować relacje między obiektami, co jest trudne do osiągnięcia w oryginalnej, wysokowymiarowej przestrzeni danych.

- Podobieństwo i Klasyfikacja:
  - Embedding pozwala na określenie podobieństwa między obiektami na podstawie odległości w przestrzeni embeddingu. Może to być wykorzystywane do zadań klasyfikacyjnych, gdzie obiekty o podobnej reprezentacji są często przypisywane do tych samych klas.
- Skalowalność:
  - Metody embedding są często skalowalne i mogą być stosowane do dużych zbiorów danych, co jest istotne w przypadku analizy danych na dużą skalę.

W przypadku formuł logicznych, embedding pozwala na przekształcenie złożonych struktur logicznych na bardziej zrozumiałe i efektywne reprezentacje numeryczne, co ułatwia ich analizę i manipulację w kontekście różnych zastosowań.

Metody Embedding: DeepWalk

DeepWalk to metoda używana głównie do reprezentowania węzłów w grafach za pomocą wektorów, co może być przydatne w różnych zadaniach analizy grafów. Jeśli masz plik CNF (Conjunctive Normal Form), co sugeruje, że mamy do czynienia z problemem związany z logiką boolowską, to zastosowanie DeepWalk może być interesujące.

Poniżej przedstawiamy kilka pomysłów, jak metoda DeepWalk mogłaby być użyta w kontekście pliku CNF:

- Reprezentacja zmiennych:
  - Każda zmienna w formule CNF może być traktowana jako węzeł w grafie. DeepWalk może pomóc w stworzeniu reprezentacji wektorowej dla każdej zmiennej, co może być użyte w dalszych analizach.

- Badanie relacji między zmiennymi:
  - Możemy użyć DeepWalk do odkrywania podobieństw między zmiennymi poprzez analizę ich reprezentacji wektorowej. To może pomóc zidentyfikować zależności lub wzorce w formule CNF.

- Rozpoznawanie wzorców klauzul:
  - Klauzule w formule CNF mogą być traktowane jako połączenia między zmiennymi. DeepWalk może pomóc w analizie tych połączeń, co może prowadzić do odkrywania istotnych wzorców w strukturze formuły.

- Predykcja wartości zmiennych:
  - Jeśli związane są zmiennymi wartości (np. prawda/fałsz), to DeepWalk może pomóc w predykcji tych wartości na podstawie ich reprezentacji wektorowej i relacji między nimi.

- Klastrowanie zmiennych:
  - Możemy użyć DeepWalk do klastrowania zmiennych w formule CNF na podstawie ich podobieństwa, co może pomóc w identyfikacji grup zmiennych o podobnym zachowaniu.

Podobne Metody Embedding

- Node2Vec Node2Vec jest rozszerzeniem DeepWalk, pozwalającym na elastyczną kontrolę nad eksploracją i eksploatacją grafu podczas generowania embeddingów.

- LINE (Large-scale Information Network Embedding) LINE koncentruje się na zachowaniu struktury grafu poprzez minimalizację funkcji straty reprezentacji pierwszego i drugiego rzędu.

- Grafowe Konwolucyjne Sieci Neuronowe (GCN) GCN to podejście oparte na warstwach konwolucyjnych, stosowane do analizy struktury grafów, co pozwala na bardziej zaawansowaną ekstrakcję cech.

- Doc2Vec Doc2Vec to technika embeddingu stosowana w analizie dokumentów tekstowych, w której każdy dokument jest reprezentowany jako wektor.

- Graph2Vec Graph2Vec stosuje podejście podobne do Doc2Vec, ale dla grafów, generując embeddingi grafów na podstawie ich struktury.

Prezentacja Danych DeepWalk

1. **Implementacja DeepWalk**

   W implementacji algorytmu DeepWalk, zastosowano podejście do budowy sekwencji losowych spacerów po grafie, a następnie wykorzystano Word2Vec do nauki reprezentacji wierzchołków.

   Word2Vec to algorytm używany do uczenia reprezentacji słów w przestrzeni o mniejszej wymiarowości. Pomimo że został pierwotnie opracowany do pracy z danymi tekstowymi, takimi jak zdania czy dokumenty, może być także zastosowany do innych rodzajów danych, w tym do reprezentacji wierzchołków w grafach, co jest często wykorzystywane w przypadku metod embeddingu grafu, takich jak DeepWalk.

   W kontekście implementacji DeepWalk, wykorzystanie Word2Vec odnosi się do sposobu uczenia reprezentacji wierzchołków na podstawie sekwencji spacerów po grafie. Proces ten można opisać w kilku krokach:

   1) Generowanie Sekwencji Spacerów:
      Algorytm DeepWalk rozpoczyna od generowania sekwencji losowych spacerów po grafie. Spacer ten jest realizowany poprzez poruszanie się po wierzchołkach grafu zgodnie z pewnymi regułami, na przykład przy użyciu losowego błądzenia.

2) Przekształcanie Sekwencji na Konteksty i Słowa:
Następnie, każda sekwencja spacerów jest przekształcana na pary "kontekst - słowo", gdzie wierzchołki są traktowane jak słowa. Kontekstem dla danego słowa może być na przykład kilka wierzchołków, które występują w tej samej sekwencji spacerów.

3) Uczenie Word2Vec:
Otrzymane pary "kontekst - słowo" są używane do trenowania modelu Word2Vec. Model ten nauczy się reprezentacji numerycznych dla każdego wierzchołka w grafie na podstawie kontekstów, w jakich występuje.

4) Uzyskanie Embeddingów Wierzchołków:
Po zakończeniu procesu uczenia, otrzymujemy embeddingi wierzchołków, które są reprezentacjami numerycznymi wierzchołków grafu w przestrzeni o mniejszej wymiarowości.

Te embeddingi wierzchołków mogą być używane do różnych celów, takich jak analiza struktury grafu, rekomendacje, czy też klasyfikacja wierzchołków. Wyniki Word2Vec w przypadku DeepWalk pozwalają uzyskać semantyczne reprezentacje wierzchołków, co umożliwia lepsze zrozumienie ich roli i relacji w analizowanym grafie.

2. **Analiza Embeddingów za pomocą T-SNE**

Do wizualizacji uzyskanych embeddingów użyto algorytmu T-SNE, umożliwiającego przedstawienie wierzchołków grafu w przestrzeni o mniejszej wymiarowości, z zachowaniem ich wzajemnych odległości.

T-SNE, czyli t-distributed stochastic neighbor embedding, to algorytm do wizualizacji danych w przestrzeni o mniejszej wymiarowości. Jego głównym celem jest przeniesienie punktów z oryginalnej, wysokowymiarowej przestrzeni danych do przestrzeni o niższym wymiarze, zachowując przy tym istotne struktury odległości między punktami.

Oto kilka kluczowych cech algorytmu T-SNE:

I. Rozkład Studenta t-distribution:
   T-SNE używa rozkładu Studenta (t-distribution) w celu modelowania odległości między punktami w oryginalnej przestrzeni i przestrzeni docelowej. Ten rozkład ma właściwość szerokiego ogona, co pozwala na skupienie się na utrzymaniu odległości między odległymi punktami w oryginalnych danych.

II. Dwuetapowy Proces:
   Algorytm działa w dwóch etapach: konstrukcji macierzy podobieństwa dla oryginalnych danych i dla danych docelowych. W obu przypadkach stosuje się różnice między odległościami do konstrukcji macierzy podobieństwa.

III. Minimalizacja Funkcji Kosztu:
   T-SNE minimalizuje funkcję kosztu, która mierzy różnice między macierzami podobieństwa w oryginalnej i docelowej przestrzeni. W rezultacie punkty, które były blisko siebie w oryginalnych danych, są bardziej prawdopodobne, aby pozostać blisko siebie po transformacji do przestrzeni o mniejszej wymiarowości.

IV. Utrzymywanie Lokalnych Struktur:
T-SNE jest znane z utrzymania lokalnych struktur w danych, co oznacza, że sąsiadujące punkty w oryginalnej przestrzeni są bardziej skłonne do zachowania tej relacji w przestrzeni docelowej.

V. Wrażliwość na Parametr Perplexity:
Parametr perplexity w T-SNE wpływa na to, ile sąsiadów brane jest pod uwagę podczas konstrukcji macierzy podobieństwa. Różne wartości perplexity mogą prowadzić do różnych rezultatów w wizualizacji.

Algorytm T-SNE jest często używany do wizualizacji danych w obszarach takich jak analiza embeddingów wierzchołków grafów. W kontekście projektu, w którym analizowane są embeddingi wierzchołków uzyskane za pomocą DeepWalk, T-SNE pozwala na przedstawienie tych embeddingów w przestrzeni o niższym wymiarze, co ułatwia zrozumienie ich struktury i wzajemnych relacji.

Wyniki i Interpretacja

Wynikiem dla samego algorytmu DeepWalk dla plików CNF (Conjunctive Normal Form) powinny być numeryczne reprezentacje wierzchołków grafu logicznego, uzyskane poprzez proces embeddingu. Każdy wierzchołek grafu, reprezentujący formułę logiczną, zostanie przekształcony w wektor numeryczny, który zachowuje istotne relacje i strukturę logiczną.

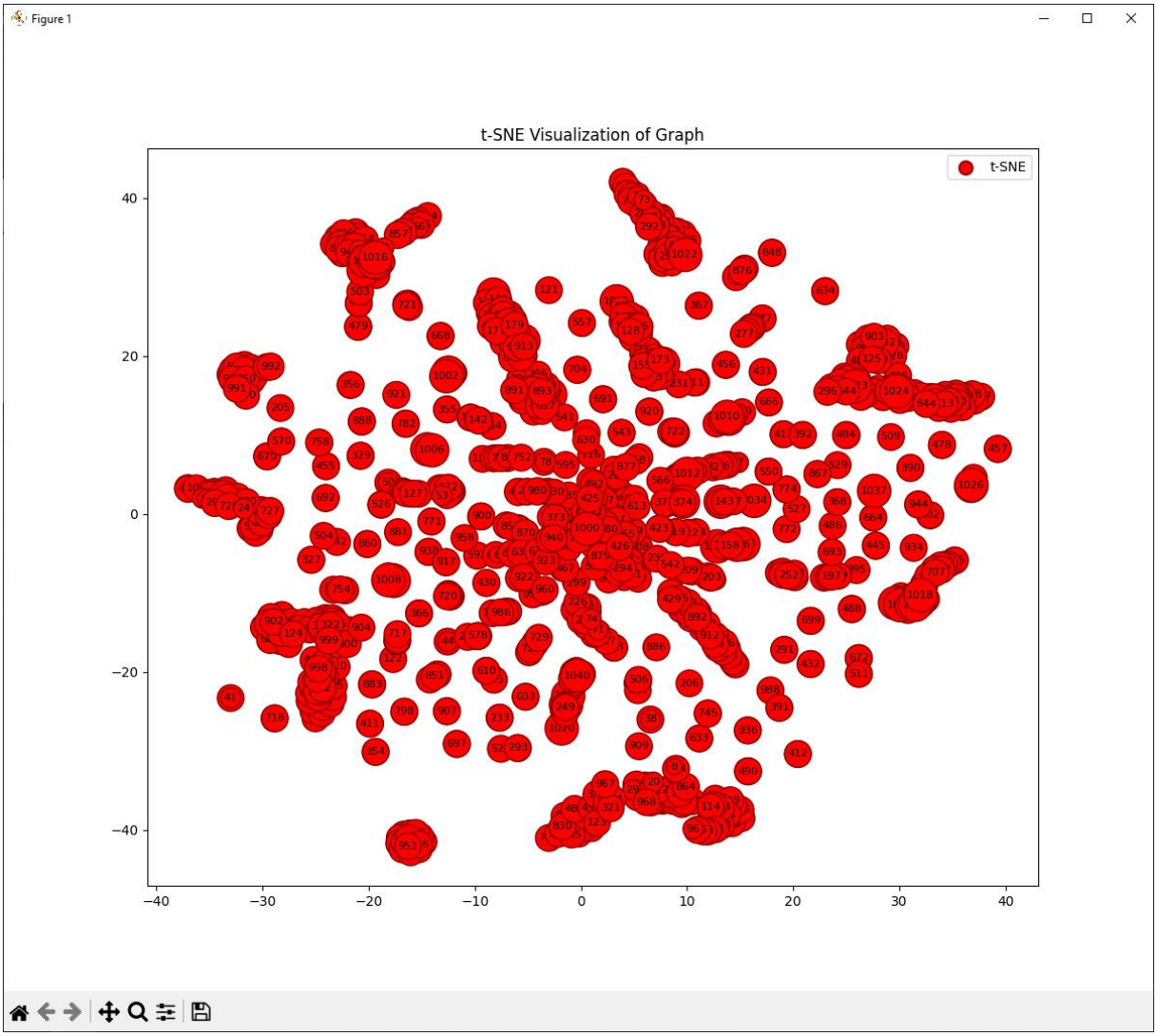
Przykład:

```
Wyniki przetwarzania formu◆y CNF:
Node 700 representation: [-0.05015413  0.05179213  0.00329128  0.04021302  0.04523034  0.05052309
 -0.02521885  0.01960666 -0.02480915 -0.0439726  -0.04933899 -0.01168919
 -0.04952234 -0.00561014 -0.01004201 -0.00064804]
Node 985 representation: [ 0.09726603 -0.14443228 -0.00117936 -0.01728959  0.22276004 -0.17958786
  0.28214926  0.2616957  -0.20635062 -0.03166614 -0.27441826 -0.21693802
  0.24692212  0.01430595 -0.39435783  0.22477786]
Node 1039 representation: [ 2.5293496e-01 -3.0934867e-02  1.2338757e-01  2.0051093e-01
  2.2498429e-01  7.0184492e-02  2.8267816e-01 -7.4799825e-03
  9.3429536e-03  5.0689980e-02 -1.2700604e-01 -1.1479657e-04
  2.0210367e-01 -2.9862061e-01 -3.6894313e-01  4.7876537e-01]
Node 701 representation: [ 0.7238183  -0.60250133  0.11953508 -0.6251324   0.7139104  -1.1070457
  1.1038647   1.2977906  -1.681222   -0.48052526 -1.0632093  -1.0773604
  1.0316055   0.1734838  -1.9310565   0.58037454]
Node 699 representation: [ 0.54610384 -0.7055759   0.1987048  -0.584439    0.64590615 -1.0131243
  1.2380309   1.3627675  -1.639042   -0.2855018  -1.1520172  -1.2292703
  0.93555284  0.01132892 -1.9691519   0.5117908 ]
Node 663 representation: [ 0.21455382  0.6210269   0.6630194   0.2155244   1.238077   -0.6995828
  0.9507906   1.2001173  -0.81963116 -0.8396864   0.08233946  0.5259742
  0.6139535  -1.4534348  -0.81068295  1.3006175 ]
Node 664 representation: [ 0.40308058  0.47596067  0.81700635  0.1511032   1.2653393  -0.7238499
  0.8719227   1.0533531  -0.75016963 -0.94849694  0.23166607  0.59105563
  0.43542978 -1.4476738  -0.9661762   1.3678856 ]
Node 667 representation: [ 1.2727288  -1.4514563   0.74958456  1.7939088   0.5870417  -1.717783
  0.7202004   0.7318968   0.11862318 -1.0289006   1.6870208  -0.520833
  1.7890025  -1.4541829  -1.9661658   0.9526932 ]
Node 706 representation: [ 1.2466276  -1.3069011   0.66997385  1.8280346   0.6212764  -1.6653123
  0.82931936  0.7699604   0.20064013 -1.076987    1.7441151  -0.48196524
  1.7959546  -1.4784567  -1.8875012   0.9543292 ]
Node 665 representation: [ 1.1130502  -1.0590398   0.5885066   1.2983825   0.3792159  -1.3455886
  0.49189875  0.6613765   0.02250505 -0.7354265   1.3322873  -0.40026483
  1.4356292  -1.0035871  -1.4224795   0.75436676]
Node 669 representation: [ 0.3297131  -0.24084918  0.13877094  0.2725846   0.14950834 -0.29692763
  0.13580911  0.12406757  0.07837853 -0.15117082  0.3300263  -0.12628919
  0.26908967 -0.25795865 -0.2820511   0.14199461]
Node 39 representation: [ 0.8302078  -0.9936511  -0.39876065  1.0368955   0.5646379  -0.07498803
  0.62199926  0.9925494   0.621702   -0.5524645  -0.81924564  0.09891495
  0.93386716  0.13121803 -1.650889    1.4015808 ]
Node 988 representation: [ 0.72999054 -0.9504298  -0.42375833  1.0364094   0.7069928  -0.13071787
  0.64630103  0.981804    0.6467885  -0.6009929  -0.87555367  0.02959346
  1.0145164   0.09452669 -1.6555277   1.3899088 ]
Node 37 representation: [ 0.24477287 -0.44762444 -0.2592663   0.27755287 -0.09387704 -0.15540302
 -0.08731195  0.36001965  0.3141345  -0.3428102  -0.3675788   0.05629604
  0.40122417  0.1997936  -0.50916624  0.22079672]
```

Wynikiem prezentacji danych za pomocą metody wizualizacji T-SNE są zbiorowiska wierzchołków, im bliżej siebie są wierzchołki na grafie, tym bardziej podobne są do siebie w oryginalnym, wysokowymiarowym zbiorze danych. Algorytm T-SNE jest zaprojektowany tak, aby zachować struktury podobieństw między wierzchołkami, co oznacza, że w przestrzeni o niższej wymiarowości zachowuje się odległości między wierzchołkami, które były blisko siebie w oryginalnej przestrzeni danych.

W rezultacie zbiorowiska wierzchołków, które obserwujemy po zastosowaniu T-SNE, odzwierciedlają ich podobieństwo w oryginalnym grafie. Wierzchołki reprezentujące podobne formuły logiczne są skupione razem, tworząc klastry lub grupy. To ułatwia analizę i zrozumienie struktury danych logicznych, ponieważ wierzchołki, które są ze sobą podobne, są reprezentowane jako bliskie sobie punkty w przestrzeni wizualizacji T-SNE.

Warto jednak zauważyć, że w przypadku T-SNE, odległości między klastrami niekoniecznie odzwierciedlają rzeczywiste odległości między nimi w oryginalnych danych. T-SNE ma tendencję do skupiania się na utrzymaniu lokalnych struktur, więc odległości między klastrami mogą być zniekształcone. Dlatego interpretacja wyników T-SNE wymaga uwzględnienia ograniczeń tego algorytmu i zrozumienia, że odległości w przestrzeni T-SNE są bardziej związane z relacjami lokalnymi niż globalnymi w oryginalnych danych.



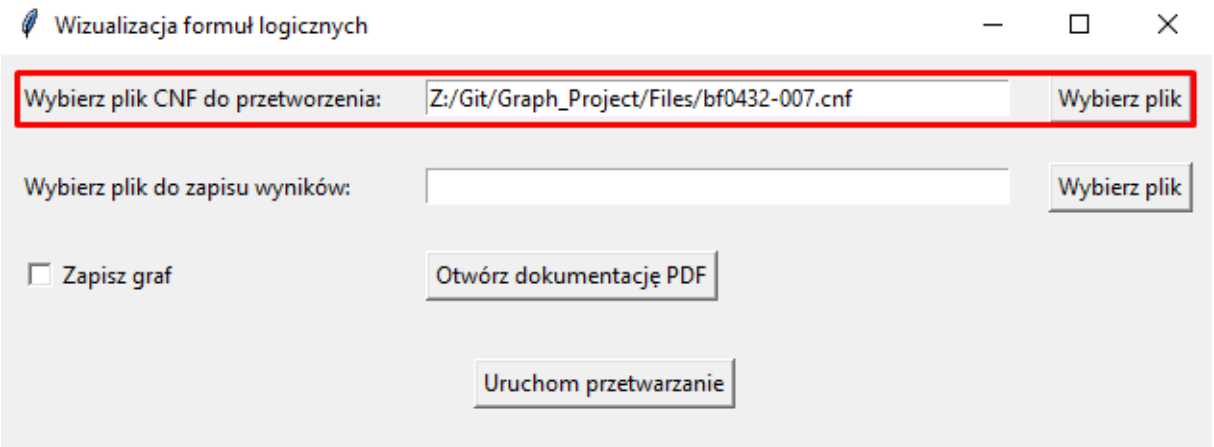
Aplikacja

Wymagania:

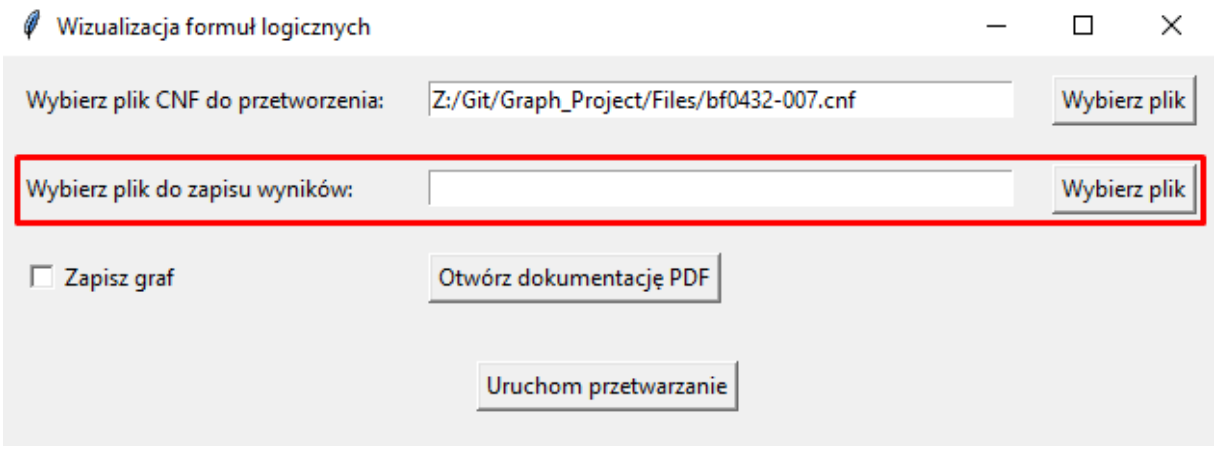
- Wymagania minimalne:
  - Wersja Pythona 3.x

- Zainstalowane biblioteki języka Python:
  - Tkinter
  - Networkx
  - Matplotlib
  - Webbrowser
  - Os

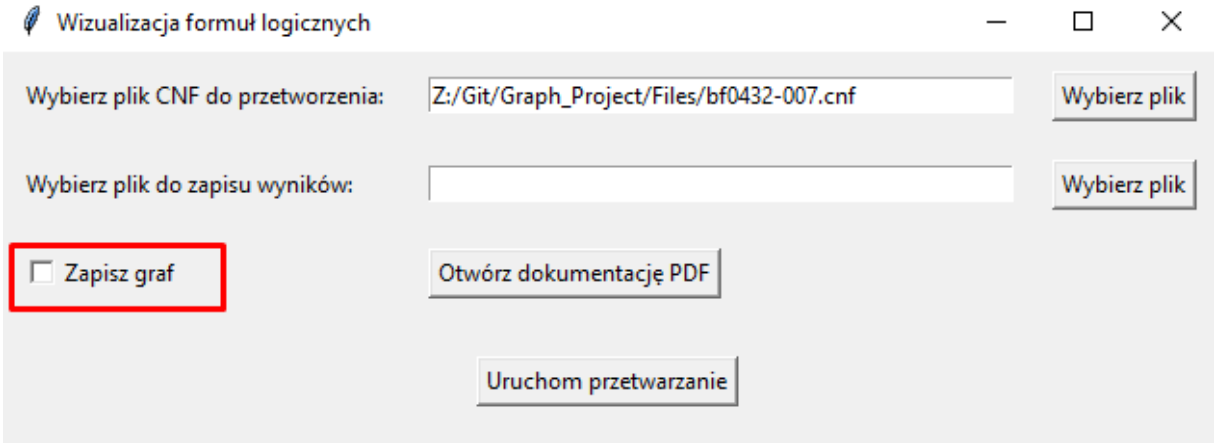
- Działanie aplikacji:



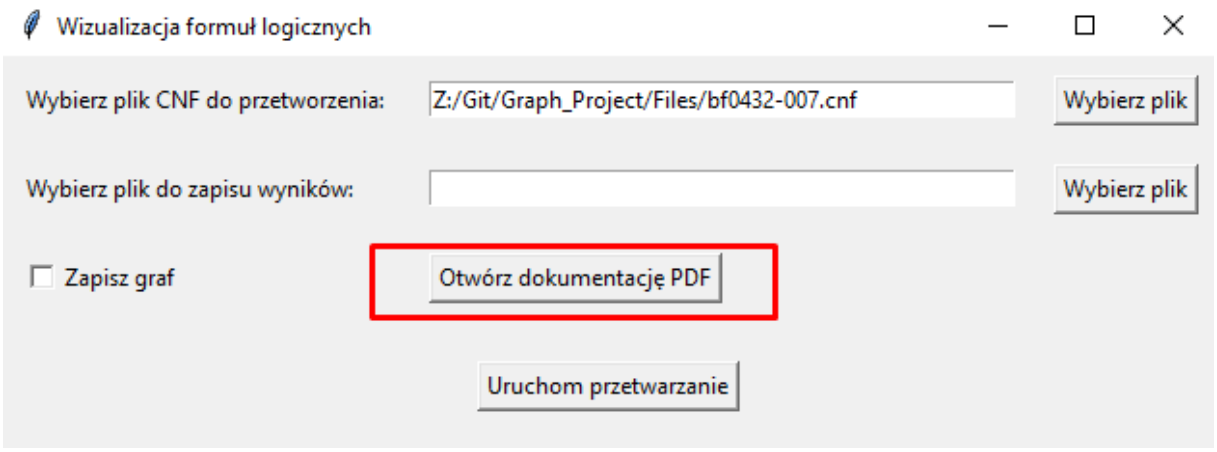
Wybieramy plik w formacie CNF, który zostanie przetworzony.



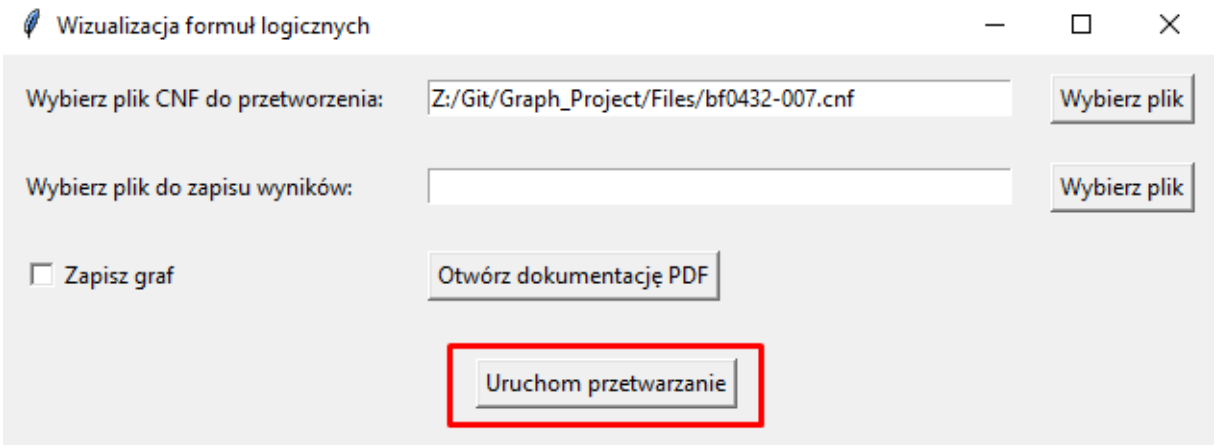
Jeżeli chcemy zapisać wyniki przeliczenia metodą DeepWalk możemy wprowadzić tutaj plik docelowy. Alternatywnie można ręcznie wpisać w tym oknie np. „wynik.txt”, co spowoduje zapisanie pliku w głównym folderze projektu.



Zaznaczenie opcji „Zapisz graf” daje możliwość zapisania w postaci pliku png wygenerowany graf.



Przycisk „Otwórz dokumentację PDF” otworzy aktualnie czytaną dokumentację w domyślnej przeglądarce.



Przycisk „Uruchom przetwarzanie” rozpoczyna właściwe działanie programu i rozpoczyna się przeliczenie, a następnie zostanie wyświetlony graf.

Bibliografia

https://arxiv.org/abs/1403.6652
https://arxiv.org/abs/1607.00653
https://arxiv.org/abs/1503.03578
https://arxiv.org/abs/1609.02907
https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf
https://towardsdatascience.com/graph-embeddings-the-summary-cc6075aba007
http://forvis.agh.edu.pl/docs
https://people.sc.fsu.edu/~jburkardt/data/cnf/cnf.html