Big Data Analysis
Group 6

Cedric Mingneau (1232611)
William Thenaers (1746077)
2019-12-01

Clustering: Topic Drift

1. *Clearly state the algorithm used to perform clustering on the chosen dataset. Explain* why *you have used this algorithm in the first place.*

   - In onze aanpak gebruiken we het *Latent Dirichlet Allocation* (LDA) algoritme [1] voor het clusteren van de dataset. Dit is een vaak gebruikte methode uit het Natural Language Processing domein dat steunt op volgende eigenschappen:
     - Documenten met gelijkaardige woorden, hebben gewoonlijk ook dezelfde topic;
     - Documenten waarin een bepaalde groep van woorden frequent voorkomt, hebben de zelfde topic.

     In dit geval bestaan de documenten uit de gefilterde en versimpelde woorden uit de titels van de publicaties in de DBLP-dataset.
   - LDA is een *unsupervised machine learning* algoritme dat met behulp van een statistisch model als metriek *topic modeling* toepast. Hierbij steunt het algoritme op de frequentie en kans dat een woord voorkomt in een document (of titel) ten opzichte van andere woorden, over alle documenten.
   - De implementatie maakt gebruik van het de scikit-learn module in Python en diens *LatentDirichletAllocation* methode onder de decomposition categorie [2].
   - Voordat we begonnen, dachten we na op welke manier er clusters gevormd konden worden met de titels uit de DBLP-dataset om bijvoorbeeld k-means toe te passen. Het leek voor de hand te liggen om de frequentie van woorden te bepalen in de volledige dataset van titels. Voor elke publicatie zouden we dan een combinatie kunnen maken van de kansen dat elk woord in diens titel voorkomt om hiermee een similariteit tussen twee publicaties te kunnen berekenen. Deze “afstand” zou een clusteringalgoritme dan kunnen gebruiken om te bepalen welke documenten kort bij elkaar liggen en mogelijk over dezelfde topic gaan. Het leek echter niet triviaal om dit in de bestaande module toe te voegen.

     We zochten dus verder naar een ander onderdeel in scipy of scikit-learn dat gebaseerd was op de kansen dat woorden voorkwamen als metriek. We stuitte op de *CountVectorizer* klasse die deze omzetting deed en we zagen dat die gebruikt werd in de context van topic modeling. Het LDA-algoritme maakt hiervan gebruik en na wat initiële tests, vonden we dat dit het resultaat gaf waar we naar zochten.

2. *Elaborate on the chosen clustering metric and potential alternatives. Moreover, discuss how the number of clusters k (or the cluster sizes) are established.*

   - Zoals vermeld in punt 1, maken we gebruik van de *CountVectorizer* als metriek voor LDA. Deze klasse zet de *corpus* data van alle publicaties (een lijst van alle titels) om naar een *sparse matrix* van token tellingen.
   - Bij het inlezen van de DBLP-dataset, worden enkel de titels in een bepaalde categorie beschouwd (vb sigmod) en gegroepeerd per publicatiejaar. Voordat ze worden toegevoegd, worden ze schoongemaakt: leestekens en andere speciale characters worden verwijderd, nummers genegeerd, er wordt *stemming* toegepast (PorterStemmer), en ten laatste wordt een woord pas beschouwd als het niet in een negeerlijst staat.
   - De *CountVectorizer* verwijdert op zijn beurt ook nog eens standaard stopwoorden uit het Engels. En geeft buiten de eerser genoemde matrix, ook een *vocabulary* of woordenlijst van

Cedric Mingneau (1232611)
William Thenaers (1746077)

elk woord over alle titels. Dit wordt gebruikt om later indices te mappen naar het oorspronkelijke woord.

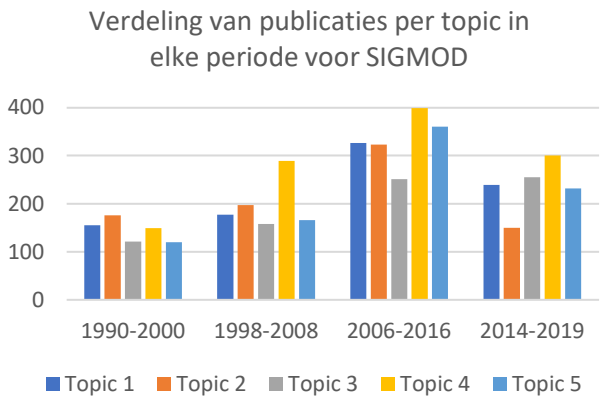
- Het aantal clusters, of topics, die in het resultaat voorkomen wordt bepaald door een inputparameter van de gebruiker. LDA bepaalt zelf welke woorden een hoge kans hebben om samen een goede topic te vormen, en kent elke publicaties toe aan één van deze topics aan de hand van de frequenties uit de input data en welke woorden samen voorkwamen in elke titel.
- Het eigenlijke clusteren gebeurt door in een (n-dimensionale-)matrix voor elk document aan te duiden met welke kans een woord erin voorkomt, en wellicht met extra dimensies, welke woorden samen voorkomen. Door het herrangschikken van de rijen en kolommen, ontstaan er vervolgens clusters van woorden die vaak samen voorkomen en een hoge kans hebben om zo ook voor te komen.

3. *Present your results, preferably accompanied by a visualization.*

- Tijdens het uitvoeren van LDA, worden de top 10 meest voorkomende woorden uitgeprint voor elke topic, samen met hoeveel documenten voldeden aan elke topic en voor welke periode de classificatie geldt.
    - Voor de primaire visualisatie gebruiken we een *wordcloud* representatie van elke topic en een samengevoegde wordcloud voor alle topics samen. Deze bevat een cloud voor elk van de topics en een cloud waarin de meest voorkomende woorden voor de hele periode werden samengevoegd.
    - Een secundaire visualisatie werd gemaakt met behulp van de pyLDAvis module [3]. Deze geeft een visuele representatie weer in de vorm van een html-pagina van het resultaat van het LDA-algoritme. De dimensies van de interne parameters worden op een 2D vlak geprojecteerd zodat er effectief clusters te zien zijn. Verder is er een overzicht van de woorden in elke topic met hun voorkomen. Klikken op een woord geeft aan hoe significant het is in elke topic.
- Voor elk van de voorgestelde categorieën uit de opgave (edbt, icde, icdm, kdd…) alsook voor alle publicaties samen werden outputs gegenereerd met 5 clusters, over periodes van 10 jaar, vanaf 1990, en met 2 jaar overlap. (vb 1990-2000, 1998-2008, 2006-2016…)

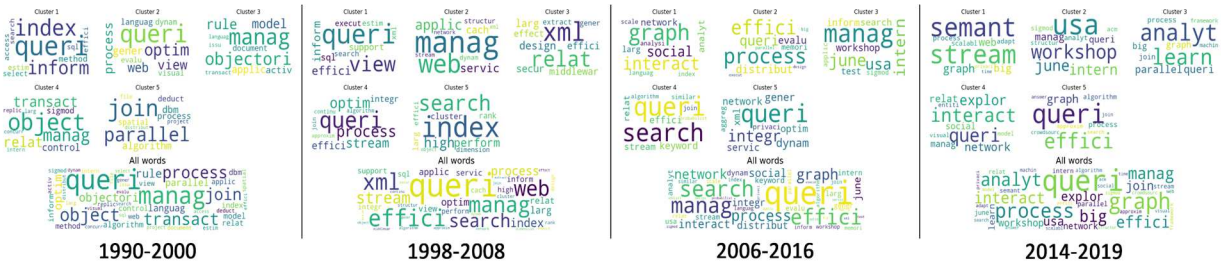
  Deze zijn allemaal te vinden in de *visualisation* map.

- Als voorbeeld de output voor de SIGMOD categorie. Figuur 1 geeft het aantal publicaties per periode weer. Merk op dat de topics voor elke periode wel verschillen en dit enkel een indicatie geeft van hoeveel publicaties uit de SIGMOD categorie in elke topic zitten. Op Figuur 3 zijn de wordclouds te zien die de inhoud van elke topic bepalen. Door vijf clusters te kiezen, zien we hier dan ook vijf topics terug.



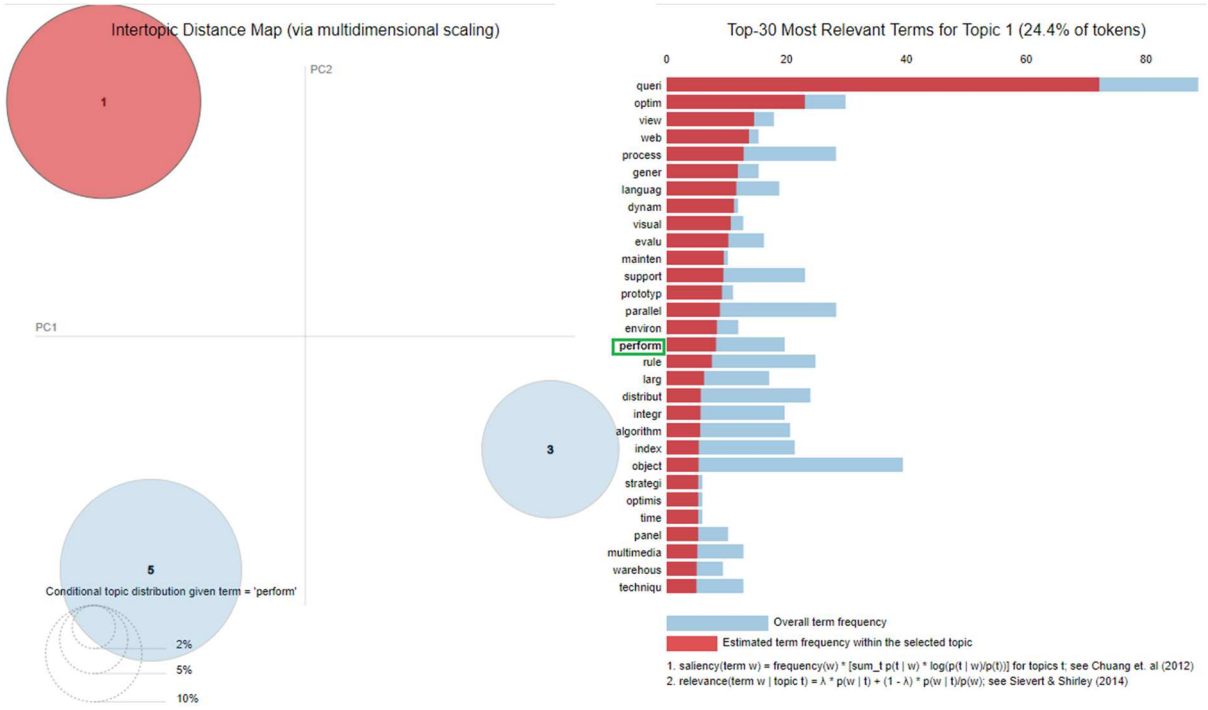
Figuur 1 Topic verdeling voor SIGMOD categorie

Cedric Mingneau (1232611)
William Thenaers (1746077)



Figuur 3 Wordclouds voor elke periode voor SIGMOD categorie

Figuur 2 hieronder laat de output van de pyLDAvis module zien. Links de vijf clusters, waarbij cluster 1 geselecteerd werd en rechts de woorden in die cluster met hun voorkomen in die cluster (rode balk) ten opzichte van het totaal (blauwe balk). In dit geval werd het woord “perform” geselecteerd, waardoor enkel de clusters waar dat woord in zit worden getoond, met hun relatieve grootte van hoeveel publicaties met dat woord er in die cluster zitten.



Figuur 2 De pyLDAvis output voor SIGMOD in de periode 1990-2000, waar het woord "perform" werd aangeduid.

- Op Figuur 3 is aan de grootte van elk woord duidelijk te zien welke het prominents zijn in elke topic. Zo zien we in voor elke topic (met woorden in stamvorm door *stemming*) in de periode 1990-2000:
  - Topic 1: queri, index, inform, method
  - Topic 2: queri, optim, view, web
  - Topic 3: manag, objectori, rule, model
  - Topic 4: object, manag, transact, relat
  - Topic 5: join, parallel, algorithm, dbm

  Er is een duidelijk onderscheid te zien van woorden, wat erop duidt dat de publicaties in elk van deze topics waarschijnlijk over een onderwerp gerelateerd aan deze woorden zal gaan, zoals LDA ook voorspelt. Voor de volgende periode van 1998 tot 2008, zien we dat “web management”, “XML relations” en “search index” prominenter worden. Dit komt natuurlijk

Cedric Mingneau (1232611)
William Thenaers (1746077)

door de sterke stijging in de grootte en het gebruik van internet waar zoekmachines stijgen in populariteit en XML-gebaseerde HTML pagina's de norm worden.

4. *Describe potential improvements and issues encountered along the way.*

- Het uitvoeren van LDA gebeurt relatief snel, maar omdat het model gebaseerd is op kansen, is er mogelijk nog enige afstelling nodig op goede resultaten te bekomen. Ook is de output enkel een collectie van woorden die een grote kans hebben om samen een topic te vormen, maar welke topic dit precies is, moet door de observant bepaald worden. Enkele parameters die aan te passen zijn:
  - Meerdere uitvoeringen doen met andere randomisatie en de resultaten vergelijken;
  - Misschien nog meer hoog frequente of irrelevante woorden uitfilteren;
  - Meer of minder clusters/topics proberen eruit te halen.
- Terwijl het model in zekere zin kan starten met trainingsdata en daarna echte data kan *transformen*, is het niet rechtstreeks mogelijk om op voorhand aan te geven welke woorden altijd bij elkaar horen of welke topics al gekend zijn. Dit komt natuurlijk voort uit het feit dat het naar ontwerp *unsupervised* is en is gebaseerd op willekeurigheid.
- Een voorbeeld van een probleem dat meer afstelling vereist, is te zien op Figuur 3 voor de periode 2014-2019. Blijkbaar bevatten de titels van een 150-tal publicaties vooral "USA", "workshop" en "June". Dit wijst erop dat er verschillende publicaties ook de plaats, datum en jaar van de conferentie waarin ze werden gepubliceerd in hun titel hebben staan. Deze woorden zouden mogelijk ook gefilterd kunnen worden (vb namen van maanden en dagen en van de meest bekende plaatsen van conferenties).
- Aan de andere kant is de manier waarop LDA werkt, met een gewogen lijst van elk woord en de kansen dat het voorkomt, vrij intuïtief. Doordat we ook eerst aan deze manier dachten, zijn we uiteindelijk ook bij LDA uitgekomen.

Cedric Mingneau (1232611)
William Thenaers (1746077)

Verwijzingen

[1] D. M. Blei, A. Y. Ng en M. I. Jordan, „Latent Dirichlet Allocation," *Journal of Machine Learning Research,* nr. 3, pp. 993-1022, 2003.

[2] „Decomposing signals in components (matrix factorization problems)," scikit-learn developers, [Online]. Available: https://scikit-learn.org/stable/modules/decomposition.html#latent-dirichlet-allocation-lda. [Geopend 1 12 2019].

[3] bmabey, „bmabey/pyLDAvis: Python library for interactive topic model visualization. Port of the R LDAvis package.," [Online]. Available: https://github.com/bmabey/pyLDAvis. [Geopend 1 12 2019].