Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Вовед во науката за податоци

Креирање графови на знаење од неструктурирани податоци со помош на големи јазични модели

Ментор: М-р Благоја Јанкоски, проф. Д-р Слободан Калајџиски
Студенти: Ана Доказа, Новица Цветкоски, Давид Стојков

Содржина

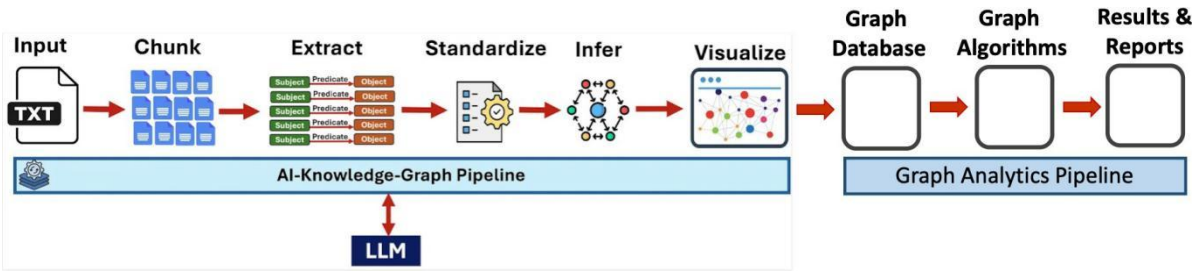
Вовед

Тема:

- Креирање графови на знаење од неструктурирани податоци со помош на големи јазични модели
- Ќе се користат готови решенија/апликации со код кој е отворен и достапен за експериментирање
- Домен/област: Проектен Менаџмент

Фази:

- Подготовка на податочно множество со неструктурирани податоци (PDF doc, webpages, video transcription)
- Креирање графови на знаење и нивно зачувување во граф база на податоци
- Анализа на добиените графови на знаење со примена на алгоритми за графови
- Подготовка на извештаи за сработеното и презентација на добиените резултати



Користено податочно множевство

Податочното множество кое е користено во експериментот се состои од неструктурирани текстуални содржини. Тие се добиени од три категории извори: ПДФ документи, веб страници и блогови, како и транскрипти од видеа.
Содржината на текстуалните податоци е насочена кон доменот на Проектниот менаџмент, при што се поставува посебен акцент кон Waterfall методологиите.

Неструктурирани податоци, односно транскрипти од YouTube:

1. The Complete Project Management Body of Knowledge in One Video (PMBOK 7th Edition)
    - Author: David McLachlan
    - Link: https://www.youtube.com/watch?v=2gmCr40uT4U
2. The PMP Fast Track - the FASTEST way to get up to speed for your PMP Exam
    - Author: David McLachlan
    - Link: https://www.youtube.com/watch?v=eUOJ_yEeyuc
3. The Complete Process Groups Practice Guide in One Video (Previously the PMBOK 6th Edition)
    - Author: David McLachlan
    - Link: https://www.youtube.com/watch?v=b5X3Z6X56uk
4. 63 Project Management Tools Explained: From the PMBOK Guide
    - Author: David McLachlan
    - Link: https://www.youtube.com/watch?v=vi0drXKr7PM
5. Project Management Simplified: Learn The Fundamentals of PMI's Framework ✓
    - Author: Deniz Sasal
    - Link: https://www.youtube.com/watch?v=ZKOL-rZ79gs
6. Project Management Fundamentals: It's all in the Basics!
    - Author: Deniz Sasal
    - Link: https://www.youtube.com/watch?v=KM3-H6PfTe0
7. PMBOK® Guide 6th Ed Processes Explained with Ricardo Vargas!
    - Author: Ricardo Vargas
    - Link: https://www.youtube.com/watch?v=GC7pN8Mjot8&t=1s

Податоци во форма на на Portable Document Format (PDF) ISO 32000:

8. The standard for program management – Fifth Edition
    - Published by: Project Management Institute, Inc.
    - ISBN: 978-1-62825-814-1
9. Process Groups: A Practice Guide
    - Published by: Project Management Institute, Inc.
    - ISBN: 978-1-62825-783-0
10. PMI – Project Performance Domains
    - Published by: Project Management Institute, Inc.
    - ISBN: N/A (Presentation)
11. PMI – Models Methods and Artifacts
    - Published by: Project Management Institute, Inc.
    - ISBN: N/A (Presentation)
12. A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Seventh Edition

- Published by: Project Management Institute, Inc.
- ISBN: 978-1-62825-664-2

13. PMI – 12 Principles of Project Management
    - Published by: Project Management Institute, Inc.
    - ISBN: N/A (Presentation)

Веб страници и блогови:

14. Title: PMI Articles
    Link: https://www.pmi.org/search#sort=datedesc&f:ContentType=[Article]&f:contentsourcestitle=[PM%20Network,Project%20Management%20Journal,Project%20Management%20Quarterly]&f:assetlanguages=[English]&numberOfResults=100
15. Title: PMI Blogs
    Link: https://www.pmi.org/search#sort=datedesc&f:ContentType=[Blog]&numberOfResults=100
16. Title: PMI Community Active Blogs
    Link: https://www.projectmanagement.com/blog/allblogs.cfm?orderBy=date&pageNum=1&blogStatus=1
17. Title: PMI Community Archived Blogs
    Link: https://www.projectmanagement.com/blog/allblogs.cfm?orderBy=date&pageNum=1&blogStatus=2

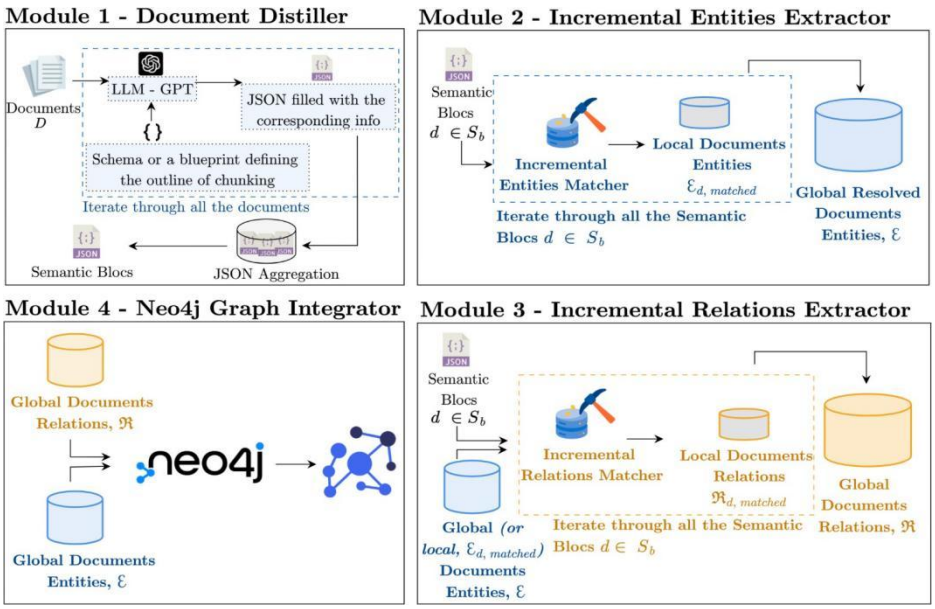
Детален извештај – itext2kg

1. Цел

Пакетот **iText2KG** е развиен со цел да овозможи **автоматско и инкрементално конструирање на конзистентни knowledge graphs (KG)** од неструктурирани текстуални документи. Основната идеја е преку **LLM модели** да се изврши извлекување на ентитети и односи и тие да се интегрираат во централизирана граф структура. Системот е дизајниран за **zero-shot** примена, без потреба од претходно тренирање за одреден домен.

2. Методологија

- **Entity & Relation Extraction со LLM-и** – користење на големи јазични модели за препознавање ентитети и нивните релации во текстот.
- **Incremental KG Construction** – новите документи континуирано се додаваат во графот, со резолуција на ентитети и избегнување на дупликации.
- **Document Distillation** – документите се трансформираат во семантички блокови со релевантни факти, со што се намалува шумот и се подобрува квалитетот на податоците.
- **Neo4j интеграција** – графот се визуализира во Neo4j за интерактивно разгледување и анализа.
- **Контрола на халуцинации кај LLM-и** – воведени механизми за замена на измислени ентитети и репромптирање при "forgetting effect".



3. Активности и функционалности

1. Основни модули на архитектурата:
    - *Document Distiller* – структурирање на текст во факти и семантички блокови.
    - *Incremental Entity Extractor* – извлекување и дисамбигуација на ентитети.
    - *Incremental Relation Extractor* – откривање на релации меѓу ентитетите.
    - *Graph Integrator & Visualization* – интеграција во Neo4j и визуелна анализа.
2. Нови функционалности (29/07/2025):
    - *iText2KG_Star* – нова, поедноставна верзија што директно извлекува релации и автоматски ги изведува ентитетите → побрзо и со помалку токени.

- *Facts-Based KG Construction* – градење на KG преку извлекување на факти од Document Distiller.
- *Dynamic Knowledge Graphs* – поддршка за динамички графови што се развиваат со текот на времето (tracking на snapshots со датуми на набљудување).

3. Перформанс и стабилност (19/07/2025):
   - *Asynchronous Architecture* – миграција кон async/await за побрзо и поефикасно извршување.
   - *Structured Logging* – воведено логирање наместо print, со нивоа DEBUG, INFO, WARNING, ERROR.
   - *Async Batch Processing* – оптимизација при обработка на повеќе документи паралелно.
   - *Error Handling & Retry* – поиздржлив систем за продукциска средина.
4. Претходни подобрувања (07/2024 – 09/2024):
   - Воведување на data models за Entity, Relation и KnowledgeGraph.
   - Ентитети се embed-ираат со комбинација од име и label (пример: *Python:Language ≠ Python:Snake*).
   - Контрола на тежини за име/label embeddings (0.6 : 0.4 default).
   - max_tries параметри за намалување на халуцинации при екстракција.
   - Поддршка за сите LangChain chat/embedding модели.

4. Резултати

- Пакетот овозможува **конзистентни и поиздржливи knowledge graphs**, дури и во присуство на шум или доменски непознат текст.
- *iText2KG_Star* значително ја зголемува ефикасноста со елиминација на непотребни чекори и со намалување на потрошувачка на токени.
- *Facts-based KG Construction* обезбедува поквалитетни и попрецизни графови преку фокус на факти.
- *Dynamic KG* ја проширува функционалноста, овозможувајќи следење на еволуцијата на знаењето низ времето.
- Новата async архитектура овозможува работа со поголеми количини на документи и подобро искористување на LLM APIs.

5. Заклучок

iText2KG претставува современо решение за автоматизирано градење на knowledge graphs од неструктурирани текстови. Со најновите подобрувања – како *iText2KG_Star*, facts-based KG construction и поддршка за динамички графови – пакетот нуди балансирана комбинација на **прецизност, ефикасност и скалабилност**. Ова го прави применлив во домени како: истражување, управување со документи, enterprise knowledge management, како и интелигентни системи за анализа и препораки.

Главни точки кои ги постигнува кодот - explore_itext2kg.ipynb

1. Подготовка на алатки
   - Инсталирање на `langchain_ollama` и `ollama` за поврзување со LLM (локален јазичен модел).
   - Импортирање на `iText2KG_Star`, `DocumentsDistiller` и `iText2KG`, алатки за извлекување знаење од текст (Knowledge Graph Extraction).
2. Вчитување на документи
   - Се користи `PyPDFLoader` за вчитување и делење на PDF документот.
3. Дестилација на информации
   - Се креира `DocumentsDistiller` кој со помош на LLM извлекува структурирани информации (на пример, `Article`) од PDF текстот врз основа на зададен „информациски упит" (IE_query).
4. Генерирање семантички блокови
   - Од резултатот (`distilled_text`) се формираат чисти семантички парови (клуч:вредност) кои ќе се користат за генерирање на knowledge graph.
5. Градење на Knowledge Graph
   - Се иницијализира `iText2KG` со избраниот LLM и embedding модел.
   - Се врши:
     - Екстракција на ентитети и релации од документи,
     - **Автоматска проверка и спојување на ентитети** кои се дупликати или не се добро дефинирани ("invented entities").

ЦЕЛ НА ЦЕЛИОТ ПРОЦЕС:

Да се обработи PDF документ (во случајов test.pdf) и автоматски да се извлече структурирана репрезентација на знаење (ентитети и нивни односи) во форма на **Knowledge Graph**, со помош на LLM и `iText2KG`

Детален извештај за кодот – testingDifferentModels.ipynb

1. Цел

Главната цел на овој notebook е да се испита и спореди примената на различни јазични модели (LLM) во задачи поврзани со **обработка на документи, извлекување информации и креирање на embeddings**. Се нагласува евалуација на перформансите и резултатите на моделите во контекст на автоматизирано управување со знаење.

2. Методологија

- Користење на **LangChain** како интеграциона рамка за работа со повеќе модели.
- Поврзување на модели преку **Ollama**, со акцент на:
  - **TinyLlama 1.1b**
  - **Llama3:8b**
  - **Llama3.2:1b**
  - **Gemma2:2b**
- Примена на **PyPDFLoader** за вчитување и поделба на документи (пример: test.pdf во случајов).
- Користење на **DocumentsDistiller** за дестилација и поедноставување на содржини.
- Вклучување на метрики за време на извршување со цел мерење на ефикасност.

3. Користени ресурси

Хардвер:
AMD Ryzen 7 6800H with Radeon Graphics
NVIDIA GeForce RTX 3050 Laptop GPU
16GB DDR5-4800 RAM
Податочно множевство:
[10, 11, 13] од податочното множевство комбинирани во еден тест фајл.
PMI – Project Performance Domains
PMI – Models Methods and Artifacts
PMI – 12 Principles of Project Management

4. Активности

1. Иницијализација на моделите и embeddings преку Ollama.
2. **Вчитување на PDF документ** и сегментирање на текстот по страници за полесна анализа.
3. **Примена на DocumentsDistiller** за добивање појасни, концизни и структурирани извадоци од документите.
4. **Тестирање на различни модели** врз истите документи со цел да се спореди нивната ефикасност и точност.
5. **Мерење на време на извршување** за секој модел и анализа на нивната практична примена.

5. Резултати

Models		Total time			Total extracted				Time per page		
Destill	Extract	Destill	Extract		Length	Entites	Relation		Destill	Extract	Total
Gemma2:2b	Gemma2:2b	485	693,86	sec	17624	16	30		34,64	49,56	84,20429
Llama3.2:1b	Llama3:8b	126,4	549	sec	8581	7	15		9,03	39,21	48,24286
Llama3:8b		775,29	1000 - T		4289				55,38		

Табелава ги прикажува спроведените експерименти за брзината на генерирање графови користејќи различни комбинации од модели, односно кој модел е користен, времето за дестилација на текст, времето за извлекување, бројот на извлечени ентитети и релации и врепе по станица.

Extrapolated Test											
Models		Total time			Total extracted				Time per page		
Destill	Extract	Destill	Extract		Length	Entites	Relation		Destill	Extract	Total
Gemma2:2b	Llama3:8b	485	693,86	sec	17624	14,38	30,81		34,64	39,21	73,86
Gemma2:2b	Gemma2:2b	485	549	sec	17624	16,00	30,00		34,64	49,56	84,20
	Llama3:8b	126,4	693,86	sec	8581	7,00	15,00		9,03	39,21	48,24
Llama3.2:1b	Gemma2:2b	126,4	549	sec	8581	7,79	14,61		9,03	49,56	58,59
Llama3:8b	Llama3:8b	775,28	693,86	sec	4289	3,50	7,50		55,38	39,21	94,59
Llama3:8b	Gemma2:2b	775,29	549	sec	4289	3,89	7,30		55,38	49,56	104,94

Табелава ги прикажува екстраполираните експерименти за брзината на генерирање графови користејќи различни комбинации од модели, односно кој модел е користен, времето за дестилација на текст, времето за извлекување, бројот на извлечени ентитети и релации и врепе по станица.

Star Test												
Models			Total time			Total extracted				Time per page		
Destill	Extract	Embed	Destill	Extract		Length	Entites	Relation		Destill	Extract	Total
Llama3.2:1b	Gemma2:2b	nomic-text	123,76	183	sec	8581	33,00	42,00		8,84	13,07	21,91

Табелава го прикажува спроведениот експеримент за брзината на генерирање графови користејќи ја најдобрата комбинација од модели и itext2kg star односно кој модел е користен, времето за дестилација на текст, времето за извлекување, бројот на извлечени ентитети и релации и врепе по станица.

	Objects per 1k characters		
	Enteties	Relations	Total
Gemma2:2b	0,91	1,70	2,61
Llama3:8b	0,82	1,75	2,56
Llama3.2:1b	N/A	N/A	N/A

Табелава го прикажува просечниот број на ентите и врски помеѓу нив извадени од 1000 карактери изворен текст.

6. Заклучок

Овој експеримент потврдува дека **различните модели имаат различен баланс меѓу брзина и квалитет**. Помалите модели како Llama3:2:1b и Gemma2:2b се соодветни за побрзи, едноставни задачи, додека за покомплексна анализа се потребни посилни модели.

Но, сепак брзината добиена од користењето на овие модели при ваквите помали експерименти, кобинирано со користењето на itext2kg_star се покажа како погодоно решение.

Детален извештај за кодот – testingIterationWithStar.ipynb

1. Цел

Главната цел на овој notebook е да се тестира и евалуира употребата на различни јазични модели (LLM) за **итеративна обработка и структурирање на знаење** од сложени документи. Посебен акцент е ставен на примената на `iText2KG_Star`, модул кој овозможува создавање, надградување и подобрување на структурирани претстави на текстуални информации.

2. Методологија

- Користење на **LangChain** како рамка за интеграција на различни модели и алатки.
- Вклучување на повеќе **LLM модели** преку Ollama:
  - **Llama3.2:1b** – модел за основна генерација и анализа на текст.
  - **Gemma2:2b** – модел за споредбено тестирање и подлабока анализа.
- Генерирање embeddings преку моделот **`nomic-embed-text:latest`** за репрезентација на текстуални податоци.
- Вчитување и обработка на документи преку **`PyPDFLoader`** (пример: *PMI Project Performance Domains*).

3. Активности

1. **Подготовка на околина и модели** – иницијализација на LLM-и и embeddings.
2. **Вчитување на документ** – поделба на PDF-датотеката на сегменти за полесна анализа.
3. **Примена на `iText2KG_Star`** – итеративно извлекување знаење од текстот и претворање во структурирана форма (knowledge graph стил).
4. **Дестилација на информации** – преку `DocumentsDistiller` се добиваат поконцизни и релевантни претстави на текстуалните содржини.

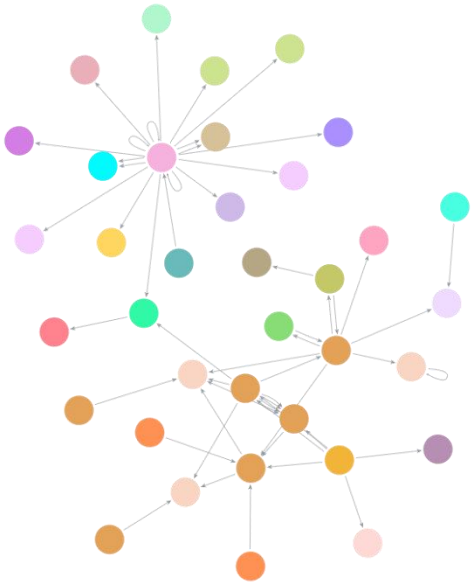
4. Резултати

- Успешно е воспоставен **итеративен процес на обработка**, кој овозможува постепено подобрување на квалитетот на извлечените информации.
- Се добиваат **структурирани претстави на текстуалните содржини**, погодни за понатамошна анализа или интеграција во knowledge graph системи.

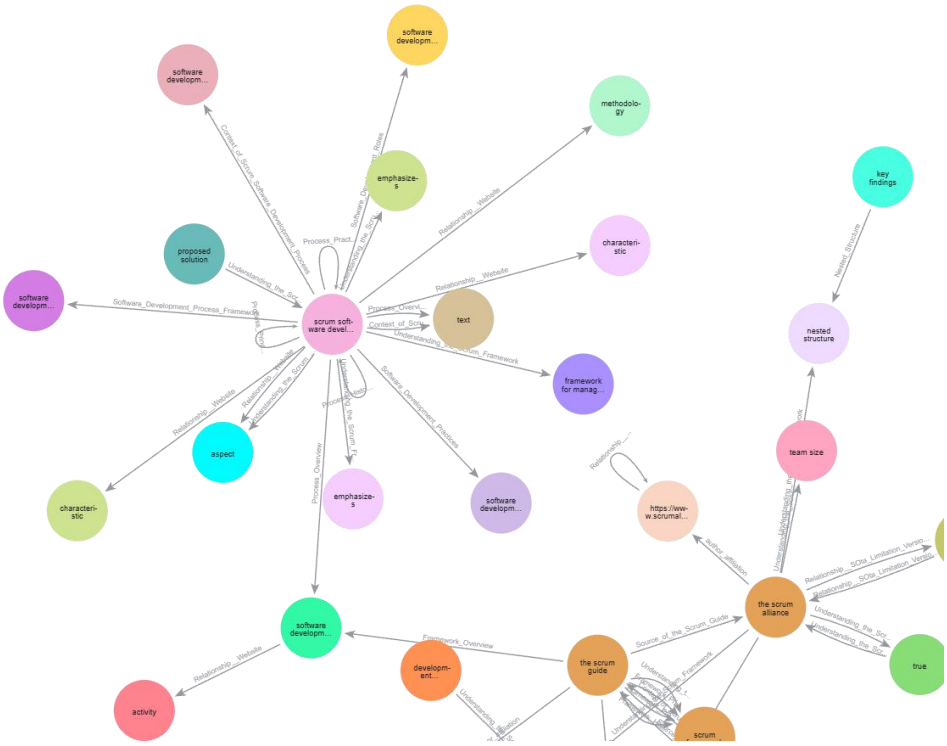
5. Заклучок

Овој експеримент потврдува дека итеративната обработка преку `iText2KG_Star` е ефективен пристап за **автоматско извлекување, структурирање и организирање на знаење од документи**.

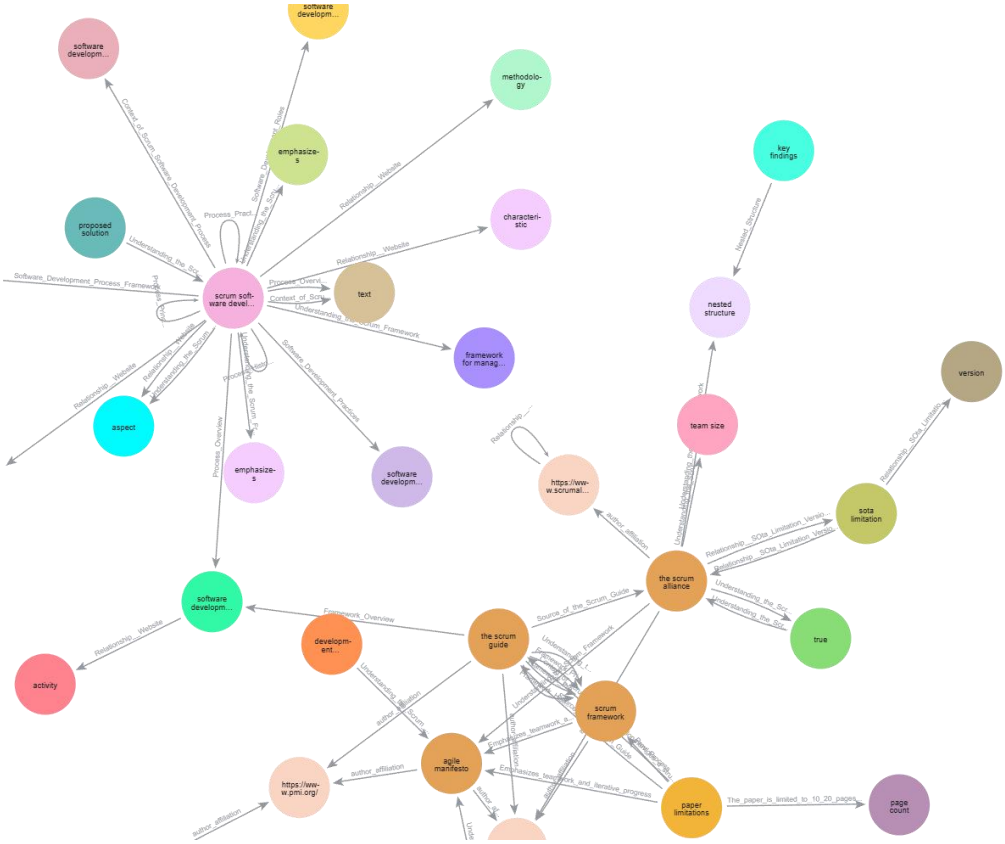
6. Графот добиен од експериментов



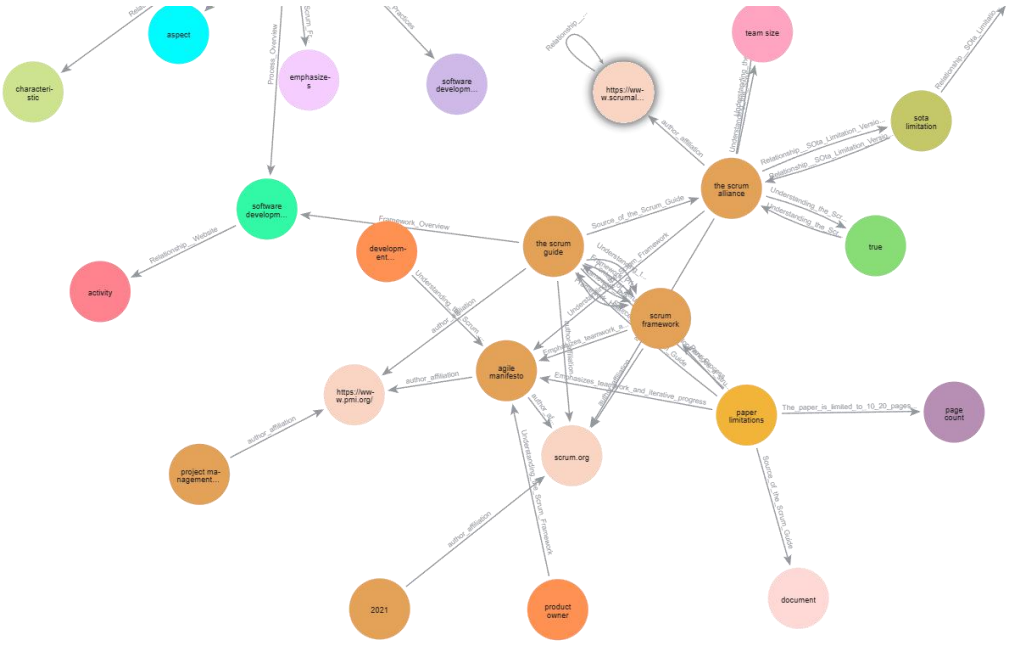
Графот содржи 36 ентитети и 54 релации.



Првата групација во графот



Втората групација во графот



Третата групација во графот

7. Користено податочно множевство: 10, 11, 13 од листата.

Извештај за кодот – testingUnstructuredSplitting.ipynb, unstructuredProcessing.ipynb, transcriptsWithSaving.py

Овие 3 документи опишуваат целосен работен тек за обработка на YouTube транскрипти за Управување со проекти (Waterfall методологии) и нивна конверзија во графови на знаење, вклучувајќи го и тестирањето за користените методи. Процесот е поделен во две главни фази поради ограничувања на хардверските ресурси при обработка на големи транскрипти.

Фаза 1: Дестилација на документи (Python скрипта - transcriptsWithSaving.py)

1.1 Цел
Python скриптата (transcriptsWithSaving.py) ја врши иницијалната обработка на суровите YouTube транскрипти со нивно делење на управливи сегменти и извлекување на структуирани информации.

1.2 Клучни компоненти

1.2.1 Модели на податоци

```
class ContentSource(BaseModel):
    name: str = Field(description="Name of the author/speaker/presenter")
    role: Optional[str] = Field(description="Role or position")
    affiliation: Optional[str] = Field(description="Organization or company
affiliation")


class Content(BaseModel):
    title: str = Field(description="Title of the content (article/video)")
    sources: List[ContentSource] = Field(description="Authors/speakers involved")
    summary: str = Field(description="Brief summary of the content")
    key_concepts: List[str] = Field(description="Main concepts or topics covered")
    insights: str = Field(description="Key insights and findings")
    challenges: str = Field(description="Challenges or limitations discussed")
    solutions: str = Field(description="Proposed solutions or approaches")
    practical_applications: str = Field(description="Practical applications or
implementations mentioned")
```

Моделот Content беше создаден да не собира само површни информации (наслов, резиме), туку и подлабоки слоеви:
Key_concepts - за индексирање и пребарување.
Insights - дестилирани „поуки".
Challenges & solutions - структуриран поглед на проблеми и решенија.
Practicle_applications - ја поврзува теоријата со праксата.

Оваа структура ја зголемува веројатност да за вадење ентитети од одреден текст, но нивото на noise во конечниот граф покажува кон преобемност на моделот, особено освртувајќи се кон ContentSource моделот, кој се покажа непотребен.

1.2.2 Стратегија на обработка
- Пакетна обработка: Транскриптите се обработуваат во мали пакети (batch_size=3) за оптимизација на меморија
- Ракување со грешки: Механизам за повторување со тајмаут за неуспешни пакети
- Управување со ресурси: Се користи garbage collection меѓу пакети за спречување проблеми со меморија
- Следење на прогрес: Детално логирање на времето и статусот на секој пакет

1.2.3 Техничка имплементација
- Користи ChatOllama со модел llama3.2:1b за лесна обработка
- Имплементира DocumentsDistiller од библиотеката itext2kg

- Обработува документи со специјализирани прашања за Управување со проекти
- Го зачувува секој пакет како индивидуален JSON во директориум distilled_results/

1.2.4. Клучни карактеристики

- Заштита со тајмаут: 300 секунди по пакет
- Оптимизација на меморија: Присилна garbage collection меѓу пакети
- Толеранција на грешки: Продолжува со обработка и ако некој пакет не успее
- Дебаг опции: process_single_batch_debug() за тестирање на поединечни пакети

Фаза 2: Конструкција на граф на знаење (Jupyter Notebook - Document 1)

2.1 Цел

Jupyter тетратката (unstructuredProcessing.ipynb) ги зема дестилираните JSON резултати и гради сеопфатен граф на знаење користејќи семантички блокови.

2.2 Клучни компоненти

2.2.1 Креирање на семантички блокови

Функцијата load_and_create_semantic_blocks():

- Ги вчитува сите JSON датотеки од директориумот со дестилирани резултати
- Конвертира структуирани Content објекти во семантички текстуални блокови
- Управува со различни типови податоци (листи, стрингови)
- Санитизира содржина со замена на проблематични знаци ({} → [])

2.2.2 Процес на градење на граф

- Инкрементална конструкција: Обработка во пакети од 40-50 блокови
- Конфигурација на прагови:
  - Праг за ентитети: 0.8-0.9
  - Праг за релации: 0.7-0.8
- Прогресивно градење: Користи existing_knowledge_graph за надградување на претходни итерации

2.2.3 Техничка имплементација

- Користи ChatOllama со модел gemma2:2b за градење граф
- Користи nomic-embed-text:latest за embeddings
- Имплементира iText2KG_Star за изградба на графот
- Интеграција со Neo4j за визуализација и складирање

2.2.4 Визуелизација и складирање

- Интеграција со Neo4j: Поврзување со локална Neo4j инстанца (bolt://localhost:7687)
- Санитизација на јазли: За јазли со нумерички имиња се додава префикс 'N_'
- Визуелизација: Можност за визуелно прикажување на графот

2.3 Стратегија на обработка

2.3.1 Управување со меморија

И двете фази вклучуваат стратегии за внимателно користење на ресурси:

- Пакетна обработка: Деление на податоците во управливи сегменти
- Инкрементално градење: Прогресивна обработка на делови од графот
- Чистење на ресурси: Експлицитна употреба на garbage collection

2.3.2 Ракување со грешки

- Селективна обработка: Рачно отстранување на проблематични блокови
- Механизам за повторување: Повеќе обиди при неуспех
- Грациозна деградација: Продолжување со обработка и при делумни неуспеси

2.3.3 Хардверски услови

Работниот тек е дизајниран за средини со ограничени ресурси:

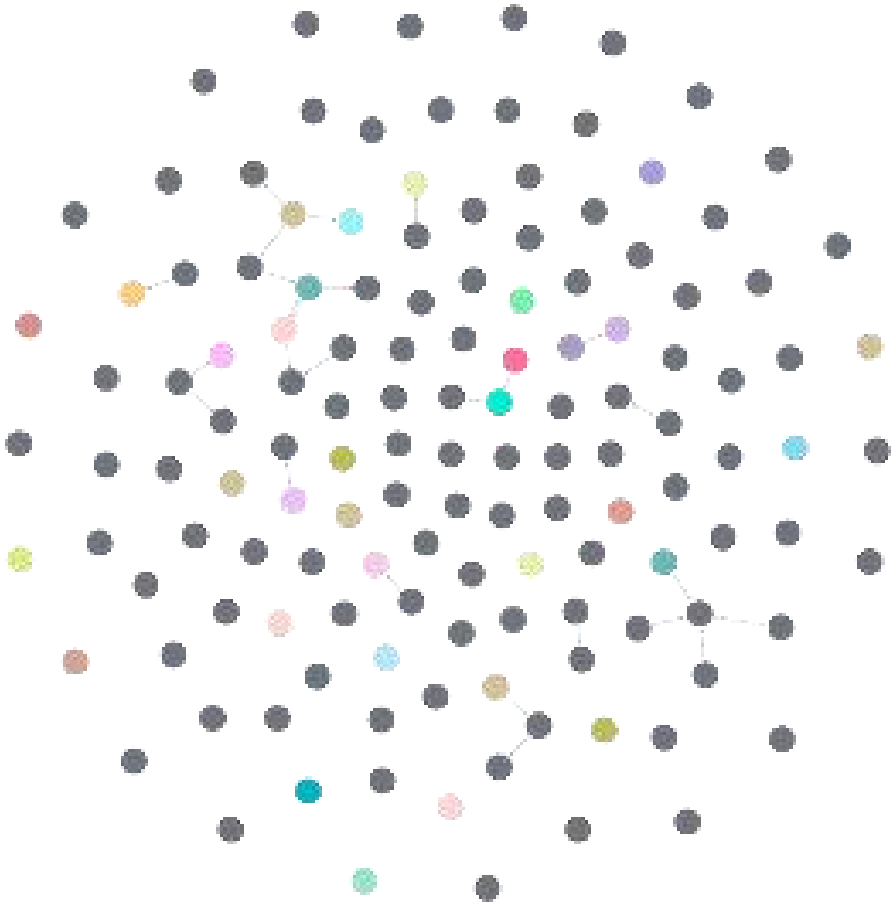
- Мали пакети: Обично 3-50 елементи по пакет зависно од фазата
- Оптимизација на меморија: Редовно чистење и постепена обработка
- Тајмаут менаџмент: Спречување на преоптоварување од заглавени процеси

3. Резултати

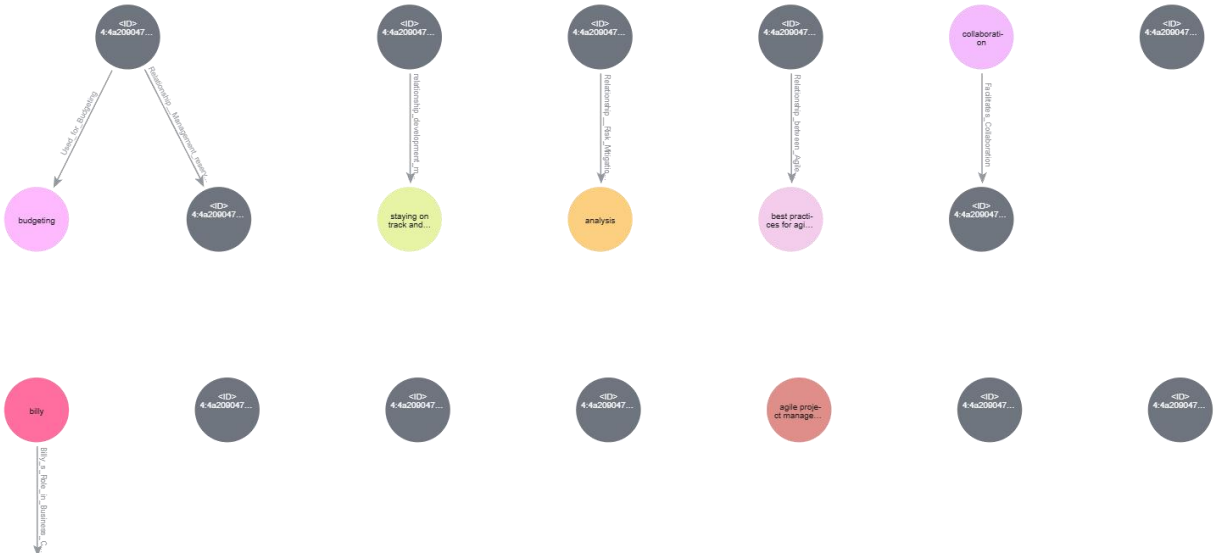
Резултатите постигнати со процесирањето на неструрираните податоци остануваат незадоволителни, резултирајќи во графови со големи количини на noise и неповрзани темиња.

Од податоците успеваме да извадиме дури 134 ентитети, но само 26 врски меѓу нив.
Од 134 темиња, 37 само имаат врски (само 27,6%) од кои:

- 7 се единични врски, вкупно 14 ентитети
- 3 се низа од 2 врски, вкупно 9 ентитети
- 1 е едно теме со 4 единечни врски, вкупно 5 ентитети
- само 1 е сложен граф од вкупно 9 ентитети



Приказ на графот од Neo4j



По детален изглед од добиениот граф.

Дел од врските кои што се добиени:

- Facilitates врски
- Source_Of врски
- Relationship_between врски

Јазлите кои имаат релации се:

- Best_Practies_for_Agile_Development
- Analysis
- Budgeting
- Collaboration
- Challenges

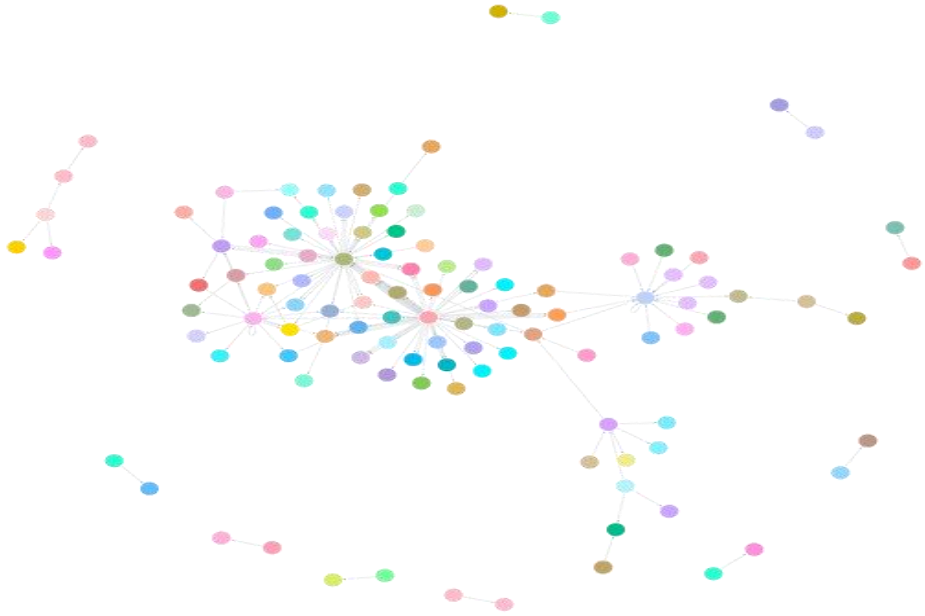
Додека, јазлите кои немаат релации се:

- Cost-to-benefit-ratio
- Analogy
- Management
- Benefits ratio
- Competitive Analysis

4. Користено податочно множевство: Број 4 од листата, односно 63 Project Management Tools Explained: From the PMBOK Guide транскриптот.

Транскриптот беше добиена користејќи youtube-transcript.io, што резултираше во .txt датотека од 74010 карактери, односно 28 batch датотеки после дестилација.

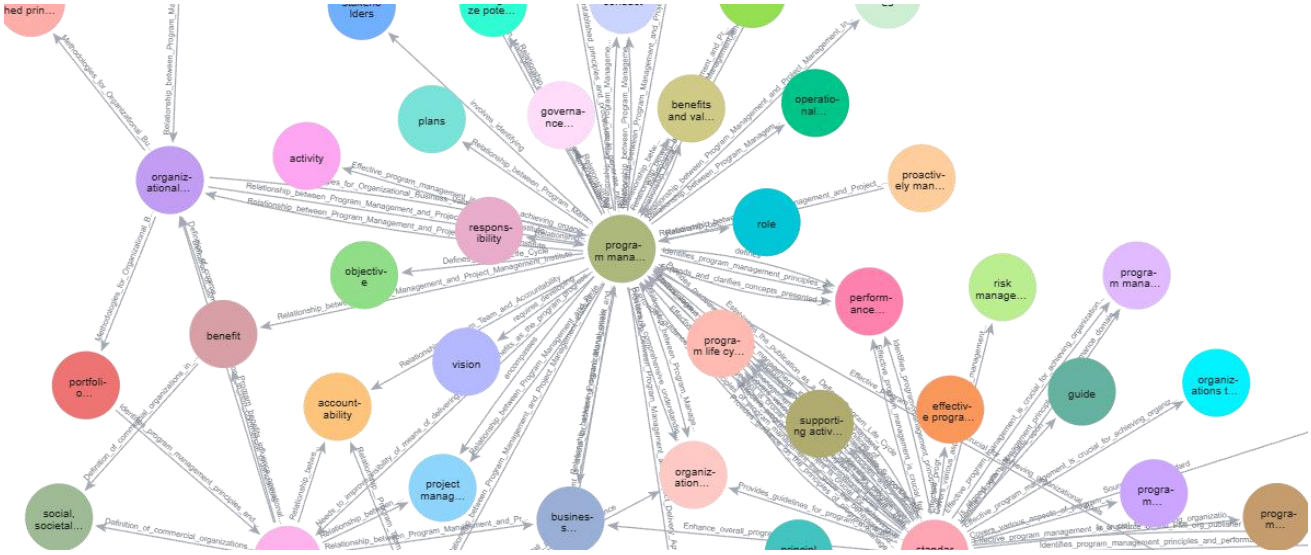
Споредби



Овај граф е добиен од книгата “The Standard for Project Managment” користејќи Itex2KG.

Добиениот граф користејќи ја нашате метода (itext2kg + Llama3.2:1b / Gemma2:2b + nomic-text-embedding) содржи 113 ентитети и 160 релации.(Користејки го StructureFileIntoChunks.py, преку кој што се добија chunks од документот “The Standard for Project Managment” и потоа преку фајлот StructuredGraphGeneration.ipynb се користат истите и преку Itext2KG се генерира граф)

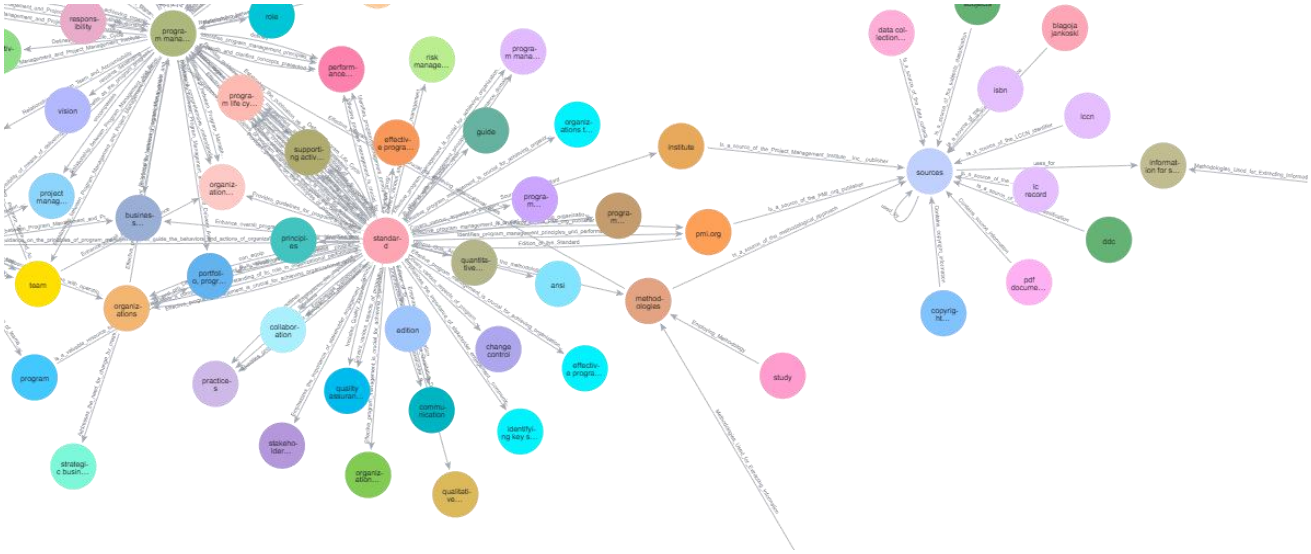
Структурата на графот изгледа случајна или полу-случајна.
Се приметува кластерирање на левата страна од графот.(Околу центарот на графот)



Централниот јазол во графот.

Ова ги покажува истите ентитети позиционирани според нивната семантичка сличност во високодимензионален простор на вградувања.
Различните бои одговараат на трите типа ентитети:

- Различните бои ги претставуваат трите различни типа јазли од графот на знаење
    - Defines – јазли кои претставуваат дефиниции или концепти што се формално опишани во текстот.
    - Emphasizes – јазли кои означуваат нагласување на одредени аспекти, принципи или практики.
    - Relationship_between – јазли кои ја претставуваат општата поврзаност или релација меѓу два ентитета.
- Обрасците на групирање покажуваат семантичка сличност меѓу ентитетите
- Просторната близина укажува на слични векторски ембедингс



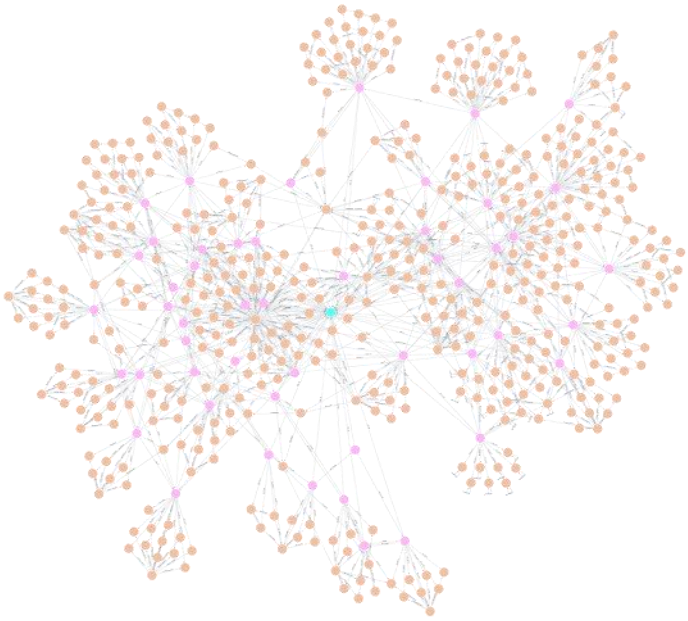
Пример за ентитетите и релациите добијани во графот.

Има одредено мешање на типовите ентитети во просторот на вградување, што сугерира семантичко преклопување меѓу различните типови јазли, што се групирани според значењето. Ова е поради што секој ентитет е преставен со вектор на високодимензионален ембединг, што е добиен со специјализиран модел за вградување(embedding model).

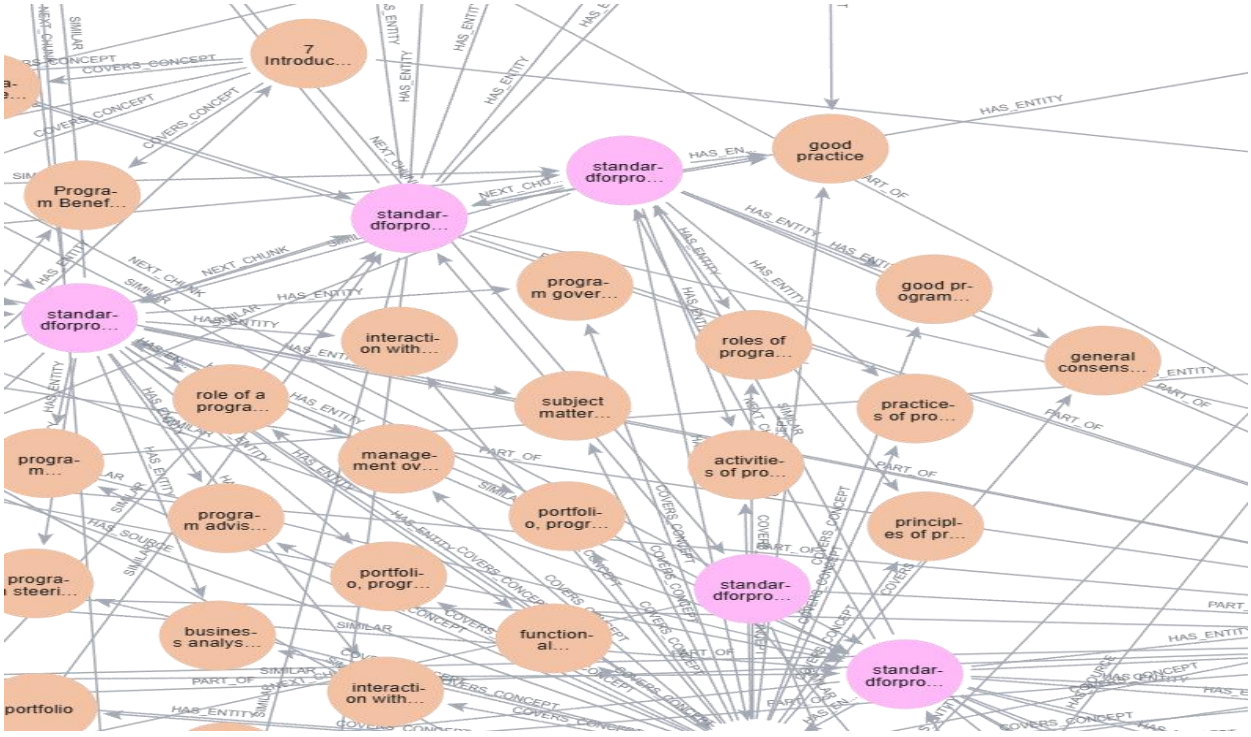
Користејќи го тој модел, се проектира во 2D простор, и според тој простор, ентитети со слична семантика се групираат просторно блиску еден до друг. Мешањето на различните типови ентитети во исти кластери укажува на семантичко преклопување.

Графот содржи околу 4 главни кластери, и тие се:

- **Голем централен кластер** - Најгустата концентрација на јазли во централно-левиот дел од сликата, со многу меѓусебно поврзани точки од различни бои
- **Горен десен кластер** - Помала но јасна група на јазли во горниот десен дел
- **Долен расфрлан кластер** - Поослабена колекција на јазли во долниот дел од сликата
- **Изолирани јазли** - Неколку поединечни јазли кои изгледаат како самостојни или многу слабо поврзани



Ова е граф што е генериран според истата книга, но користејќи го референтното решение, односно Neo4j LLM Knowledge Graph Builder.



Пример за ентитетите и релациите добијани во графот.

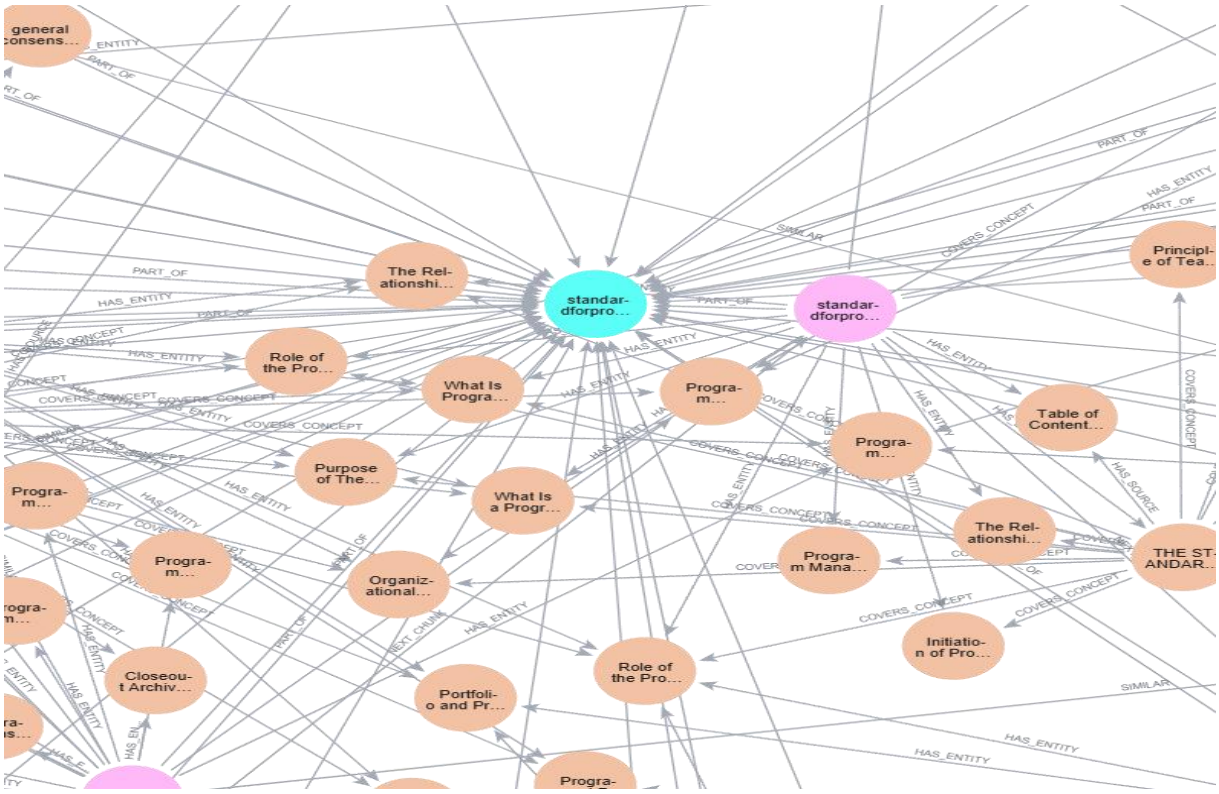
Добиениот граф користејќи Neo4j LLM Knowledge Graph Builder содржи 530 ентитети и 1,202 релации. Ова претставува мрежа на односи меѓу три типа ентитети:

- **Process_Step_Input** (јазлите со структурирана ѕвездообразна форма)
- **Supplier_of_Process_Step** (средни поврзувачки јазли)
- **Process_Step_Supplier** (периферни јазли)

  Високо организираната, радијална форма укажува на јасна хиерархиска структура на односи, веројатно покажувајќи како добавувачите се поврзуваат со процесните чекори преку различни влезни односи.

- Графот има околу 20 главни кластери, од кои што поголемите се:

- **Централно јадро** - Група на централни јазли (во розова/виолетова боја) кои служат како главни конектори
- **Радијални кластери** - Дефинирани кластери кои се прошируваат од централното јадро во радијален образец, секој составен од:
  - Еден или неколку централни јазли
  - Многу периферни јазли (претежно во портокалова/кафена боја) поврзани со централните
- **Хиерархиска структура** - Секој кластер има ѕвезда-слична или дендритна структура каде што централните јазли се поврзуваат со многу листови-јазли



Централниот јазол во графот.

Графот изваден користејќи го референтното решение беше поголем, има и голема разлика во бројот на ентитети помеѓу двата графови(369% повеќе ентитети) , исто така има и многу повеќе релации(651% повеќе релации).

153 од ентитети се преклопуваат кај двата графови, ентитети кои се преклопуваат:
Benefit, change, concept, context, document, domain, edition, goal, governance, guide, identifier, license, operation, outcome, output, performance, portfolio, practice, principle, process, program, project, result, source, stakeholder, standard, value....

Повеќето од заедничките ентитети се од основни концепти, организациски елементи, резултатиски индикатори и документациски индикатори.

Метрика	Itex2KG	Референтно решение	Однос(Itex2KG/Референтно)
Ентитети	113	530	4.69x
Релации	160	1,202	7.51x
Густина на граф	0.025	0.009	0.36x
Просечна поврзаност	2.83	4.54	1.60x

Споредба на ентитетите и релациите добијани во двата графови, сосе нивната просечна поврзаност и густина.

Не е пронајдено преклупување кај релациите, ова е поради графот добијан од референтното решение многу повеќе ги генерализира релациите помеѓу ентитети. Релациите се од типот:

- COVERS_CONCEPT
- HAS_ENTITY

- HAS_SOURCE
- SIMILAR
- Додека Itext2KG графот има повеќе доменско-специфични релации од типот:
- Emphasizes_Collaboration
- Emphasizes_Communication
- Defines_Practies
- Defines_Program_Life_Cycle

```python
class ContentSource(BaseModel):
    name: str = Field(description="Name of the author/speaker/presenter")
    role: Optional[str] = Field(description="Role or position")
    affiliation: Optional[str] = Field(description="Organization or company affiliation")


class Content(BaseModel):
    title: str = Field(description="Title of the content (article/video)")
    sources: List[ContentSource] = Field(description="Authors/speakers involved")
    summary: str = Field(description="Brief summary of the content")
    key_concepts: List[str] = Field(description="Main concepts or topics covered")
    insights: str = Field(description="Key insights and findings")
    challenges: str = Field(description="Challenges or limitations discussed")
    solutions: str = Field(description="Proposed solutions or approaches")
    practical_applications: str = Field(description="Practical applications or implementations mentioned")
    # Additional fields observed in your JSON files
    methodology: Optional[str] = Field(default=None, description="Research methodology")
    conclusions: Optional[str] = Field(default=None, description="Conclusions")
```

Шемата што ја искористивме во Itext2KG.

Ова е шемата што ја користиме за itext2KG, референтното решение ја користи истата шема.

```
CREATE CONSTRAINT content_id IF NOT EXISTS FOR (c:Content) REQUIRE c.title IS UNIQUE;
CREATE CONSTRAINT source_name IF NOT EXISTS FOR (s:ContentSource) REQUIRE s.name IS
UNIQUE;
CREATE CONSTRAINT concept_name IF NOT EXISTS FOR (k:Concept) REQUIRE k.name IS UNIQUE;


CREATE (c:Content {
  title: "AI in Finance",
  summary: "Explores AI use cases in financial fraud detection",
  insights: "AI improves fraud detection accuracy",
  challenges: "Data imbalance and explainability",
  solutions: "Federated learning and interpretable models",
  practical_applications: "Banking fraud prevention",
  methodology: "Experimental study",
  conclusions: "Promising but needs regulation"
});

// Example content source node
CREATE (s:ContentSource {
  name: "John Doe",
  role: "Researcher",
  affiliation: "MIT"
});


CREATE (k1:Concept {name: "AI"});
CREATE (k2:Concept {name: "Fraud Detection"});


MATCH (c:Content {title: "AI in Finance"}),
      (s:ContentSource {name: "John Doe"}),
      (k1:Concept {name: "AI"}),
      (k2:Concept {name: "Fraud Detection"})
CREATE (c)-[:HAS_SOURCE]->(s),
       (c)-[:COVERS_CONCEPT]->(k1),
       (c)-[:COVERS_CONCEPT]->(k2);
```

Шемата што ја искористивме во Neo4j.

Графот добијан од Itext2KG ги покажува имплицитните семантички сличност помеѓу ентитети во книгата, додека графот добијан од референтното решение ги покажува експлицитните.

Мрежните односи во графот од референтното решение не одговараат на семантичките групи во Itext2KG графот.

Ова сугерира дека ентитетите што се директно поврзани во графот на знаење можеби немаат најслични вградувања.Спротивно, ентитетите со слични вградувања (групирани заедно во Itext2KG графот) можеби не се директно поврзани во мрежата.

Во Itext2KG графот имаме подобро зачувување на контекстуалното значење, полесно е за разбирање и податоците се поконцентрирани.
Додека референтното решение има поопсежно покривање на книгата и користи универзални релациски типови.

Заклучок

Споредбата меѓу двата графикони открива фундаментална разлика во начинот на кој ги разбираме комплексните податоци. Графот добијан од референтното решение ни покажува архитектурата на системот - кој со кого комуницира и како се организирани врските. Додека Itext2KG графот ни открива длабоката семантичка структура - што значат овие ентитети и колку се слични според нивната суштина.

Клучното откритие е дека структурната поврзаност не се преклопува со семантичката сличност. Ентитети што се директно поврзани во мрежата можат да имаат различни семантички карактеристики, додека ентитети што никогаш не интерагираат директно можат да бидат семантички многу слични.
На пример, во Neo4j графот постојат експлицитни релации од типот HAS_ENTITY или COVERS_CONCEPT што директно поврзуваат ентитети како „process" и „document". Сепак, нивните embeddings во itext2kg графот покажуваат дека тие не се толку блиски семантички – односно, иако се структурно поврзани, нивното значење е различно.

Од друга страна, ентитети како „goal" и „outcome" во itext2kg графот се позиционираат блиску еден до друг во embedding-просторот (семантички слични), но во Neo4j графот тие не се директно поврзани со релација.

Референци

Lairgi, Yassir, et al. “iText2KG: Incremental Knowledge Graphs Construction Using Large Language Models.” arXiv Preprint arXiv:2409.03284, 2024, https://arxiv.org/abs/2409.03284
(Lairgi, Moncla, Cazabet, Benabdeslem, & Cléau, 2024)
GitHub: https://github.com/AuvaLab/itext2kg

Neo4j LLM Knowledge Graph Builder - Extract Nodes and Relationships from Unstructured Text - Neo4j, Inc.

www.youtube-transcript.io - Ripple Consulting BV.