DOI: 10.35681/1560-9189.2024.26.2.316711

УДК 004.5

Д. В. Ланде[1], О. О. Рибак[2]
[1]Інститут проблем реєстрації інформації НАН України,
вул. М. Шпака, 2, 03113 Київ, Україна
[2]Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Берестейський проспект, 37, 03056 Київ, Україна

Семантичне індексування та кластерний аналіз документів з кібербезпеки

Розглянуто методи екстракції концептів із текстів і побудови семантичних мереж для аналізу даних у контексті кібербезпеки. Основна увага приділена використанню великих мовних моделей (LLM) для автоматизованого витягу сутностей і побудови мереж концептів. Це дозволяє визначати взаємозалежності та структурувати інформацію, формувати семантичні мережі. Такі мережі можна використовувати для подальшого кластерного аналізу, що дає можливість автоматично групувати вузли за схожістю та визначати нові закономірності в даних. Досліджено побудову мереж близькості документів, що дозволяє оцінювати ступінь схожості текстів на основі їхніх семантичних структур. Запропонований підхід дозволяє виявляти тематично споріднені документи, що можуть містити важливу інформацію для аналізу, а також визначати інформаційні ланцюжки та ключові тенденції у великих масивах текстових даних, ключові тенденції і загрози у сфері кібербезпеки.

Ключові слова: *семантичне індексування, кластерний аналіз, модулярність, великі мовні моделі (LLM), кібербезпека, аналіз тексту, семантичні мережі.*

Вступ

Екстракція концептів із текстових повідомлень, формування семантичних мереж, їхній подальший кластерний аналіз, а також побудова мережі зв’язків окремих документів є важливими інструментами для обробки текстових даних, зокрема у сфері кібербезпеки. Вони дозволяють витягувати з текстів важливі концепти, такі як ключові слова, імена персон, організацій, хакерських угрупувань, назви злочинного програмного забезпечення тощо, а також будувати семантичні мережі зі зв’язками різних типів (серед яких найбільш важливі — причинно-наслідкові або кауза-

льні зв'язки), а також автоматично групувати вузли таких мереж для подальшого змістовного, сценарного аналізу, прогнозування тощо. Мережі зв'язків документів дозволяють виявляти ланцюжки найважливіших проблем у предметній галузі. У даній роботі розглянуто підхід до витягу концептів і зв'язків за допомогою промптів до великих мовних моделей (LLM) і подальшого аналізу мереж, що формуються.

Метою цієї роботи є представлення і обґрунтування нових методів екстракції. концептів із текстових повідомлень на базі застосування LLM, формування і аналіз семантичних мереж для аналізу текстових даних, а також мереж зв'язків документів, зокрема в контексті кібербезпеки.

У роботі [1] розглядаються можливості семантичного індексування у поєднанні з інформаційним пошуком. Автор стверджує, що LLMs не зможуть замінити пошукові системи. Проте підхід, що пропонується в нашій статті, показує напрямок застосування таких систем для створення інтелектуального індексу, що значно покращить показники інформаційного пошуку.

Екстракція понять з метою подальшого створення мереж уже широко застосовується [2], але питання саме інтеграції семантичного індексування з технологіями Big Data, інформаційним пошуком на даний час залишається відкритим.

Ми розглядаємо мережу зв'язків документів, яка формується із застосуванням семантичного індексування на основі штучного інтелекту. Раніше для цього вимагалися технології, що потребували великих ресурсів [3].

Семантичне індексування в кібербезпеці

Семантичне індексування — це процес, під час якого документи аналізуються для виділення їхньої сутності та змісту, за допомогою структурованих методів, таких як виділення ключових слів, імен власних, організацій, а також інших релевантних категорій, з метою полегшення пошуку, аналізу та кореляції між документами.

У кібербезпеці цей процес має критичне значення, оскільки текстові масиви можуть містити важливу інформацію про шкідливе програмне забезпечення, хакерські угрупування, типи атак, вразливості та інші аспекти. Автоматизовані методи семантичного індексування дозволяють швидко знаходити ключові дані у великих обсягах текстів (наприклад, у звітах про кіберзагрози, технічній документації, записах мережевих подій), тим самим забезпечуючи ефективний аналіз загроз і відповіді на інциденти.

Семантичне індексування дозволяє структурувати великі обсяги текстової інформації у кібербезпеці для швидкого пошуку та аналізу критично важливих даних, таких як назви шкідливих програм, хакерських груп і вразливостей.

Можна виділити чотири основні етапи семантичного індексування, а саме: 1) попередню обробку текстових документів; 2) екстракція сутностей; 3) виявлення зв'язків між сутностями; 4) створення індексів, їхня фіксація в полях даних.

Етапи семантичного індексування

Попередня обробка текстових документів

На цьому етапі текстові документи проходять базову обробку для підготовки до подальшого аналізу. Основними завданнями попередньої обробки є, по-перше,

очищення тексту, а саме видалення зайвих символів, пунктуації, а також стоп-слів (загальних слів, які не несуть специфічного значення, таких як «і», «або», «це»). Після цього здійснюється лематизація та стемінг [4], а саме, приведення слів до їхньої базової форми для уніфікації обробки. Це дозволяє скоротити різноманіття словоформ та покращити точність пошуку ключових термінів. При цьому стемінг — це процес зведення слова до його «стему» або кореня, який може не завжди бути словниковою формою. Стемінг працює за принципом відсікання суфіксів і префіксів від слова, щоб отримати корінь. Лематизація — це процес зведення слова до його леми, тобто базової чи словникової форми, враховуючи контекст і граматичні особливості слова. Лематизація є більш точним методом, оскільки вона використовує лінгвістичні правила для розпізнавання правильної базової форми слова залежно від його частини мови та контексту. Іноді комбіноване застосування стемінгу та лематизації дозволяє підвищити загальну ефективність системи обробки тексту, отримуючи швидкий результат зі стемінгу та коригуючи його там, де це необхідно, за допомогою лематизації. І завершує етап попередньої обробки документів токенізація, тобто поділ тексту на окремі одиниці (токени), що можуть бути словами або фразами, що потім будуть аналізуватися в подальших етапах. Токенізація є основним етапом попередньої обробки тексту в процесі семантичного індексування. Вона забезпечує структуроване представлення текстових даних і підготовлює їх для подальших етапів, таких як екстракція сутностей і зв'язків. Точна і правильно налаштована токенізація є ключовим елементом для досягнення ефективності на наступних етапах індексування, аналізу та пошуку у великих масивах текстових даних. Токенізація є ключовим процесом на етапі попередньої обробки тексту, який грає фундаментальну роль у подальших етапах семантичного індексування. Це процес розбиття тексту на окремі частини, які називаються токенами. Токен може бути словом, фразою, символом або іншим елементом, що має значення в контексті аналізу тексту.

Токенізація служить базовою операцією, яка підготовлює текст для екстракції сутностей, зв'язків і подальшого семантичного аналізу. Зупинимося детальніше на цьому процесі, розглянувши його різновиди, важливість і приклади застосування.

Токенізація включає розбиття тексту на менші одиниці [5]. Наприклад, під час аналізу речення: «Шкідлива програма WannaCry використовує вразливість CVE-2023-1234» можуть бути виділені такі токени: [«Шкідлива», «програма», «WannaCry», «використовує», «вразливість», «CVE-2023-1234»], або токени як мітки сутностей: [«Malware», «WannaCry», «Action», «Vulnerability», «CVE-2023-1234»]

Існує декілька видів токенізації, серед яких токенізація на рівні слів, токенізація на рівні слів символів, токенізація на рівні слів фраз або N-грам тощо.

Токенізація на рівні слів — це найбільш поширений підхід, де текст розбивається на окремі слова. Він підходить для більшості природних мов, де слова чітко розділені пробілами або знаками пунктуації. Розглянемо приклад токенізації речення: «Атака на сервер була проведена вночі». У результаті можна отримати токени: [«Атака», «на», «сервер», «була», «проведена», «вночі»]. Слід зазначити, що при такому процесі виникають виклики при роботі з такими елементами як абревіатури, багатокомпонентні назви (наприклад, CVE-2023-1234), та складні слова.

При токенізації на рівні символів текст розбивається на окремі символи. Така токенізація зазвичай використовується у випадках, коли важливо зберегти всі символи або при роботі з мовами, де немає чітких розділювачів між словами (наприклад, китайська), наприклад, фрагмент тексту «CVE-2023-1234» розбивається на токени: ["C", "V", "E", "-", "2", "0", "2", "3", "-", "1", "2", "3", "4"]. Цей підхід дозволяє працювати з окремими символами та спеціальними знаками, що може бути корисним при аналізі технічних текстів (наприклад, у кібербезпеці при роботі з назвами вразливостей чи версій програм).

При токенізаціїї на рівні фраз або *N*-грам текст розбивається на групи слів або фрази, що дозволяє зберегти більше контексту між словами. *N*-грами — це послідовність із *N* слів, які розглядаються як єдиний токен. Наприклад, текст «Шкідлива програма WannaCry використовує вразливість» розбивається на токени (біграми, $N = 2$): [«Шкідлива програма», «програма WannaCry», «WannaCry використовує», «використовує вразливість». Такий підхід часто використовується для поліпшення точності аналізу при витягуванні сутностей і зв'язків між ними, особливо для складних термінів, імен або фраз.

У сфері кібербезпеки, крім стандартних методів токенізації, можуть застосовуватися спеціалізовані підходи. Наприклад, токенізація може виділяти абревіатури типу CVE, назви версій програмного забезпечення або типи атак, оскільки вони мають важливе значення в контексті безпеки.

Токенізація тісно пов'язана із застосуванням великих мовних моделей, зокрема можливостями їхнього додаткового навчання (тюнингу). Розбиття на токени допомагає зосередитись на найважливіших елементах тексту, видаляючи непотрібну інформацію (таку як пробіли, незначущі символи). Завдяки токенізації сутності можна чітко виділяти й аналізувати, що значно полегшує роботу на етапах індексування та класифікації. Крім того, формування ефективних індексів базується на чіткому поділі тексту на сутності й взаємозв'язки між ними, що стає можливим завдяки токенізації.

Після токенізації текст можна перетворити в різні формати для збереження та подальшого аналізу. Одним із таких форматів є JSON.

Приклад показує, як можна зберігати інформацію про токени у структурованому вигляді разом із додатковою інформацією, такою як тип токену та його позиція у вихідному тексті.

```
{
 "original_text": "Шкідлива програма WannaCry використовує вразливість CVE-2023-1234.",
 "tokens": [
  {
   "token": "Шкідлива",
   "type": "adjective",
   "position": [0, 9]
  },
  {
   "token": "програма",
   "type": "noun",
   "position": [10, 18]
  },
```

```
    {
     "token": "WannaCry",
     "type": "Malware",
     "position": [19, 27]
    },
    {
     "token": "використовує",
     "type": "verb",
     "position": [28, 40]
    },
    {
     "token": "CVE-2023-1234",
     "type": "Vulnerability",
     "position": [41, 55]
    }
  ]
}
```


Екстракція сутностей

Після попередньої обробки тексту відбувається екстракція важливих сутностей [6]. Цей етап полягає у виокремленні з тексту фактографічної інформації, яка має значення для конкретної області аналізу. У контексті кібербезпеки це можуть бути:

— назви шкідливих програм (наприклад, «WannaCry», «Emotet»);

— назви хакерських угруповань (наприклад, «APT28», «DarkSide»);

— типи атак (наприклад, «DDoS», «SQL Injection»);

— вразливості (наприклад, «CVE-2023-1234»).

Екстракція сутностей може бути реалізована за допомогою методів автоматичного аналізу тексту, таких як машинне навчання або використання словників з наперед визначеними сутностями. Основна мета цього етапу — знайти ключові елементи інформації, які є критичними для аналізу кіберзагроз.

Витяг зв'язків між сутностями

Цей етап полягає у виявленні та фіксації зв'язків між сутностями, які раніше були витягнуті з тексту. Зв'язки можуть мати різні типи та контекстуальні значення. У сфері кібербезпеки це можуть бути такі зв'язки як, наприклад: зв'язок між хакерськими угрупованнями та шкідливими програмами, які вони використовують; зв'язок між типами атак і вразливостями, які експлуатуються; зв'язок між різними загрозами, що належать до одного типу або мають спільні характеристики. Ці зв'язки є важливою частиною семантичної мережі, оскільки вони дозволяють не лише ідентифікувати окремі сутності, але й виявляти їхні взаємозв'язки, що важливо для прогнозування поведінки загроз.

Зв'язки між сутностями можуть бути збережені у вигляді індексних структур, де кожна сутність пов'язана з іншими через певні зв'язки. Це дозволяє швидко знаходити не тільки сутності, але і контексти, в яких вони зустрічаються. Кожен зв'язок може також мати свою вагу, що відображає ступінь важливості або ймовірність цього зв'язку.

На цьому етапі виділені сутності також структуруються та фіксуються у вигляді індексів. Індекси — це дані, що дозволяють швидко знаходити релевантну інформацію у великому масиві текстових документів. Процес індексування включає: асоціацію сутностей з документами, коли кожен документ отримує відповідний набір індексів на основі сутностей, знайдених у тексті; структуроване збереження, коли індекси фіксуються у полях баз даних або інших форматах (наприклад, у вигляді таблиць або графів); крім того, індекси можуть бути динамічно оновлюванні у випадку надходження нових документів або змін у базі даних.

Зібрані сутності та їхні зв’язки фіксуються в індексах, які можуть бути представлені у структурованих форматах, таких як JSON. Це забезпечує гнучкість і простоту в обробці та зберіганні даних для подальшого аналізу. У форматі JSON можна зберігати як самі сутності, так і їхні зв’язки та ваги. Наведемо приклад даних у форматі JSON:

```
{
 "document_id": "doc123",
 "entities": [
  {
   "type": "Malware",
   "name": "WannaCry",
   "id": "entity001",
   "start_position": 15,
   "end_position": 24
  },
  {
   "type": "HackerGroup",
   "name": "APT28",
   "id": "entity002",
   "start_position": 58,
   "end_position": 63
  },
  {
   "type": "Vulnerability",
   "name": "CVE-2023-1234",
   "id": "entity003",
   "start_position": 87,
   "end_position": 100
  }
 ],
 "relations": [
  {
   "source": "entity002",
   "target": "entity001",
   "relation_type": "uses",
   "weight": 0.9
  },
  {
   "source": "entity003",
   "target": "entity001",
```

```
    "relation_type": "exploits",
    "weight": 0.8
   }
  ]
}
```

У наведеному прикладі документ містить три сутності: шкідливу програму WannaCry, хакерську групу APT28 та вразливість CVE-2023-1234. Ці сутності мають певні зв'язки між собою, наприклад, APT28 використовує WannaCry («uses»), а WannaCry експлуатує вразливість CVE-2023-1234 («exploits»). Кожен зв'язок також має вагу, що може відображати ступінь впевненості або важливості зв'язку.

Формат JSON є зручним для зберігання даних про сутності та їхні зв'язки, оскільки він дозволяє легко представляти складні структури у вигляді вкладених об'єктів. Крім того, JSON може бути легко інтегрований з багатьма мовами програмування та базами даних, що робить його універсальним форматом для зберігання індексованої інформації.

Подальші етапи обробки документів

Після проведення саме семантичного індексування, можливі подальші етапи змістової обробки. Ці етапи зосереджені на глибшому аналізі даних і витягуванні додаткової інформації.

Класифікація сутностей

На цьому етапі сутності, що були виділені під час індексування, можуть бути класифіковані за категоріями або типами. Це може бути важливо для подальшого аналізу, зокрема для групування шкідливих програм за типами атак або класифікації загроз за рівнем ризику.

Витяг зв'язків і побудова семантичних мереж

Після класифікації сутностей можливе визначення зв'язків між різними сутностями. Наприклад, можна встановити зв'язок між хакерським угрупуванням і шкідливим програмним забезпеченням, яке воно використовує, або між вразливістю і типом загрози, що її експлуатує. Це дозволяє побудувати семантичні мережі, в яких сутності виступають як вузли, а зв'язки між ними — як ребра. Такі мережі дозволяють виявляти приховані зв'язки між подіями та сутностями.

Кластерний аналіз і відображення семантичних мереж

Після побудови семантичних мереж можна застосовувати кластерний аналіз для виявлення груп сутностей, що мають схожі характеристики або поведінку. Наприклад, можна кластеризувати шкідливі програми за типами атак або створити групи вразливостей, які використовуються для певного типу загроз.

Визначення мережі зв'язків документів і її кластеризація

На цьому етапі, для кожного документа будується семантична мережа, що відображає концепти та зв'язки між ними. Далі ми порівнюємо ці семантичні мережі

попарно, будуємо мережу зв'язків між документами, яка відображає схожість між ними. Потім ця мережа кластеризується, і отримані кластери документів відповідають основним подіям за вибраною тематикою за певний період часу.

Пошук як результат індексування

Інформаційний пошук у системах кібербезпеки є кінцевим результатом семантичного індексування та класифікації даних. Після того як документи були індексовані, класифіковані та проаналізовані, користувачі можуть виконувати запити для отримання конкретної інформації. Пошукові системи, що побудовані на основі індексів, дозволяють швидко знаходити релевантні документи або сутності за допомогою ключових слів або складних запитів. Пошук є не лише частиною користувацького інтерфейсу, але і важливою складовою для безпеки, оскільки він дозволяє швидко ідентифікувати критичні загрози або ризики у великих текстових масивах.

Практичне застосування семантичного індексування у системі «Кіберагрегатор»

«Кіберагрегатор»— це система моніторингу і аналізу соціальних медіа, розроблена для збору та обробки контенту із соціальних мереж за вибраними темами, зокрема кібербезпеки [7]. Останній напрямок її розвитку — це інтеграція можливостей пошуку в соціальних мережах і штучного інтелекту, що дозволяє значно покращити аналітичні можливості системи.

Аналіз існуючих підходів до збору тематичних новин вказав на потребу в інструменті, здатному здійснювати комплексний контент-моніторинг. «Кіберагрегатор» поєднує методи інформаційного пошуку, аналізу даних та агрегування інформаційних потоків, що робить його потужним інструментом для роботи з великими обсягами даних.

Основною особливістю цієї системи є здатність автоматично обробляти повні тексти із соціальних мереж і відслідковувати динаміку інформаційних потоків у часовому розрізі. Це дозволяє відстежувати розвиток кіберзагроз, трендів і подій, що пов'язані із темою кібербезпеки, в режимі реального часу.

Витяг важливих концептів і зв'язків (одночасно етапи 1 і 2 семантичного індексування) у системі «Кіберагрегатор» здійснюється за допомогою промптів до LLM. Модель оцінює важливість концептів і зв'язків, використовуючи свою базу знань.

Формально цю це можна визначити як витяг із тексту T множини концептів: $C(T) = \{c_1, c_2, \ldots, c_{n_t}\}$ і зв'язків між ними: $R(T) = \{(c_1, c_2) | relationship(c_1, c_2)\}$, де $relationship(c_1, c_2)$ визначає зв'язок між концептами c_1, c_2. Практично будь яка сучасна модель LLM (застосовується Llama-3.1) на основі промптів (запитів) одночасно визначає важливі концепти і зв'язки між ними.

Промпт до LLM, у тіло якого має агрегуватися текст документа, виглядає так:

> *Вибери до 20 пар зв'язаних понять українською мовою (саме пари понять, а не окремі поняття) із тексту і виведи ці пари у вигляді нумерованого списку через ";", як "поняття;поняття". Кожне поняття може складатися з декількох слів. Ось текст:*
> *ГУР знову атакувало російські телеканали — джерела*

Головне управління розвідки (ГУР) Міноборони України провело масштабну кібератаку на російських провайдерів і заблокувало десятки ресурсів промислових об'єктів рф.

…

На основі відповіді моделі отримуємо такі результати:

```
ГУР;кібератака
Міноборони;розвідка
російські ресурси;війна
інтернет-провайдери;мобільні оператори
промислові об'єкти;спецтехніка
військово-промисловий комплекс;силові відомства
…
```

У наведеному прикладі як концепти розглядаються ключові слова, які разом із визначеними зв'язками між ними дозволяють формувати мережу концептів $G = = (V, E)$, де V — множина вузлів (концептів), E — множина ребер (зв'язків між концептами).

Система типу LLM може надавати різні варіанти відповідей під час повторної обробки тексту, причому більшість із них є логічно обґрунтованими з точки зору експерта—людини. Підхід із залученням рою віртуальних експертів полягає в тому, що система LLM генерує кілька відповідей на одні й ті ж запити, що дозволяє аналізувати концепти та зв'язки з різних точок зору. Кожна відповідь віртуального експерта може відповідати окремій ролі або погляду на задачу [8]. Після отримання множини відповідей від різних віртуальних експертів, їх узагальнюють, об'єднують в єдиний файл, наприклад у форматах CSV або JSON:

```
{
 "concepts": [
  {"id": 1, "label": "Cybersecurity"},
  {"id": 2, "label": "Malware"},
  {"id": 3, "label": "Threats"}
 ],
 "connections": [
  {"source": 1, "target": 2, "weight": 0.85},
  {"source": 1, "target": 3, "weight": 0.75}
 ]
}
```

Зв'язки між концептами отримують ваги залежно від частоти їх згадування. Це дає змогу створювати багатий набір даних для подальшого аналізу та інтелектуального пошуку.

Для кожної пари концептів (c_i, c_j) система LLM (або рій віртуальних експертів) може або встановити зв'язок між ними, або не встановити. Кожен раз, коли система встановлює зв'язок між c_i та c_j, ми збільшуємо вагу цього зв'язку.

Позначимо вагу зв'язку між концептами c_i та c_j як w_{ij}. Ця вага залежить від кількості разів, коли рій віртуальних експертів підтверджує наявність зв'язку між цими концептами.

Нехай $r_{ij}(k)$ — результат відповіді k-го віртуального експерта для пари концептів(c_i, c_j), де:

— $r_{ij}(k) = 1$, якщо експерт підтвердив зв'язок між концептами c_i та c_j;

— $r_{ij}(k) = 0$, якщо експерт не підтвердив зв'язок.

Загальна вага зв'язку між c_i та c_j визначається як сума всіх підтверджень від експертів:

$$w_{ij} = \sum_{k=1}^{K} r_{ij}^{k},$$

де K — загальна кількість запитів до рою віртуальних експертів.

Для того, щоб усі ваги були у межах від 0 до 1, можна нормалізувати ваги:

$$w_{ij}^{norm} = \frac{w_{ij}}{K},$$

де w_{ij}^{norm} — нормалізована вага зв'язку, яка відображає відсоток експертів, що підтвердили цей зв'язок.

Для того, щоб залишити лише найбільш сильні та значущі зв'язки, використовуємо пороговий критерій. Якщо вага зв'язку w_{ij}^{norm} перевищує певний поріг θ, то зв'язок зберігається, інакше він відкидається. Тобто залишаємо зв'язок між c_i та c_j, якщо $w_{ij}^{norm} \geq \theta$.

Значення порога θ можна налаштувати залежно від потрібної точності. Вищі значення θ залишатимуть лише ті зв'язки, які підтвердили більшість експертів, що збільшить точність мережі.

У результаті попередніх етапів побудовано мережу концептів на базі аналізу 30 документів із мережевих ЗМІ, результатів інформаційного пошуку, центральна частина яких у нашому прикладі має вигляд, що наведений на рис. 1.

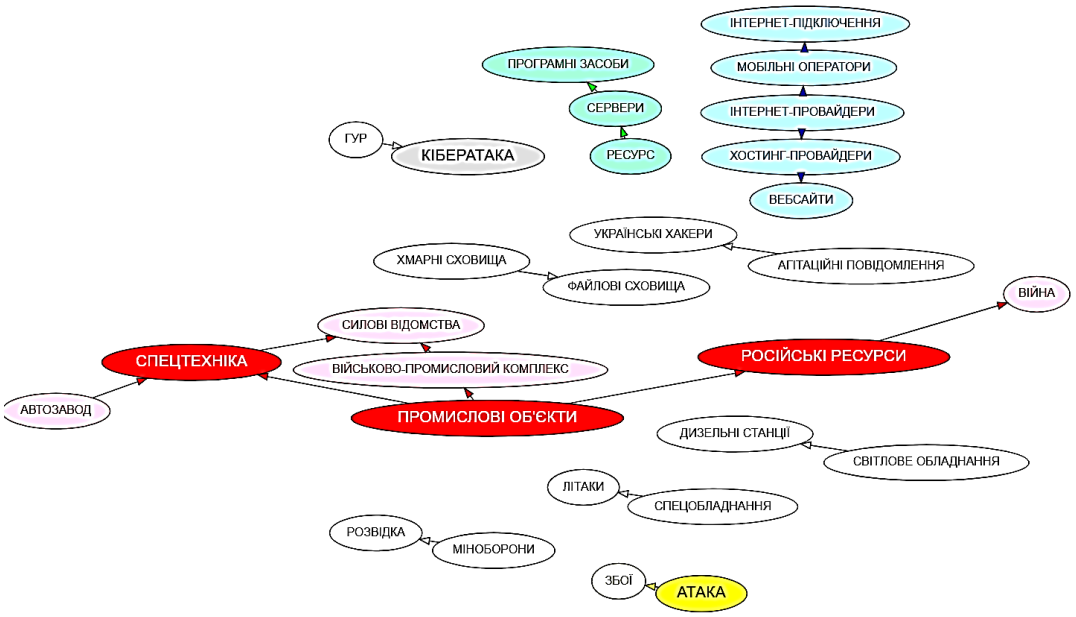

Рис. 1. Фрагмент мережі концептів, що відповідають тематиці «кібератаки на російські ресурси»

На етапі формування та кластеризації мережі документів для порівняння документів і визначення їхньої близькості можна використовувати підхід, який заснований на підрахунку спільних зв'язків понять у кожному документі. Формально, матриця близькості M будується наступним чином.

Нехай $D = \{d_1, d_2, \ldots, d_m\}$ — це множина документів. $CN(d_k)$ — мережа концептів документа d_k. Для кожного документа d_k визначимо множину пар понять: $P(d_k) = \{(c_i, c_j) | c_i, c_j \in CN(d_k), w_{ij} > 0\}$, Множини $P(d_k)$ включають усі пари зв'язаних концептів у документі d_k.

Для кожної пари документів (d_k, d_l) кількість спільних пар понять між документами визначається як

$$S(d_k, d_l) = |P(d_k) \cap P(d_l)|,$$

де $|P(d_k) \cap P(d_l)|$ — це кількість спільних пар понять між документами d_k та d_l.

Матриця близькості $M = \|m_{kl}\|$ розмірності $m \times m$, де m — кількість документів, визначається таким чином:

$$m_{kl} = \frac{S(d_k, d_l)}{\max(|P(d_k)|, |P(d_l)|)},$$

де $\max(|P(d_k)|, |P(d_l)|)$ — це нормувальний фактор, що дозволяє враховувати різну кількість пар у кожному документі та уникнути переоцінки подібності через розмір документів.

Елементи матриці близькості m_{kl} варіюються від 0 до 1, де 1 означає, що документи d_k та d_l мають максимальну схожість у парі понять, а 0 означає відсутність спільних пар понять. На рис. 2 наведено фрагмент матриці близькості.

Для групування документів на основі матриці близькості документів нами застосовується метод кластеризації за класами модулярності. Існують різні види модулярності [9], але автори застосовували модель Поттса [10], яка враховує так звану розподільну здатність.

	Документ 1	Документ 2	Документ 3	Документ 4	Документ 5	…
Документ 1	0	0,01	0,2	0,01	0,01	
Документ 2	0,01	0	0,01	0,01	0,04	
Документ 3	0,2	0,01	0	0,2	0,01	
Документ 4	0,01	0,01	0,2	0	0,06	
Документ 5	0,01	0,04	0,01	0,06	0	
…						

Рис. 2. Фрагмент матриці близькості документів

Цей підхід ґрунтується на фізичній моделі, яка використовує поняття енергії для оптимізації кластеризації у графах. Основна ідея полягає в мінімізації функціоналу енергії системи, що є аналогом задачі кластеризації. Формалізація цього підходу виглядає наступним чином:

Для графа з N вузлами та M ребрами, де a_{ij} — вага зв'язку між вузлами i та j, можна записати функціонал енергії, який потрібно мінімізувати:

$$E = -\frac{1}{2}\sum_{ij} J_{ij}\,\delta(c_i, c_j) + \sum_i h_i\,\delta(c_i)\,,$$

де J_{ij} — вага зв'язку між вузлами i та j,

$\delta(c_i, c_j)$ — функція дельти, яка дорівнює 1, якщо вузли i та j належать до одного кластера, і 0 — в іншому випадку,

h_i — зовнішній магнітний поляризаційний термін для вузла i.

Вагу зв'язку J_{ij} можна виразити через розподільну здатність γ як

$$J_{ij} = \gamma\left(a_{ij} - \frac{k_i k_j}{2m}\right),$$

де a_{ij} — вага зв'язку між вузлами i та j,

k_i та k_j — степені вузлів i та j,

m— загальна кількість зв'язків у графі,

γ — розподільна здатність

Функціонал енергії E можна переписати таким чином:

$$E = -\frac{1}{2}\sum_{ij} \gamma\left(a_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j) + \sum_i h_i\,\delta(c_i)\,,$$

де $\frac{1}{2}\sum_{ij} \gamma\left(a_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j)$ є частиною, яка визначає модулярність.

Розподільна здатність γ є константою, яка дозволяє масштабувати кількість очікуваних зв'язків між вузлами в межах одного кластера. Вона впливає на те, як розраховується очікувана кількість зв'язків між вузлами всередині кластера порівняно з випадковим розподілом:

— висока γ зменшує вагу зв'язків всередині кластерів, що може знизити модулярність;

— низька γ збільшує вагу зв'язків всередині кластерів, що може підвищити модулярність.

У практичних застосуваннях значення h_i (друга складова функціоналу енергії) можна визначити різними способами, залежно від контексту задачі. Ось кілька підходів:

1. Простий варіант визначення h_i може бути встановлення його фіксованого значення, наприклад $h_i = 1$. Це підходить, коли немає конкретних зовнішніх факторів, які впливають на кластеризацію.

2. Вибір h_i як степеня вузла k_i може бути розумним варіантом, особливо якщо важливість вузла в мережі визначається його степенем. У цьому випадку: $h_i = k_i$.

3. Можна використовувати нормалізовані значення степеню або інші метрики, які відображають важливість вузла, з метою забезпечення коректної масштабованості:

$$h_i = \frac{k_i}{\sum_j k_j}.$$

Цей підхід нормалізує ступінь вузла, роблячи його більш порівнянним із значеннями для інших вузлів.

На рис. 3 наведено приклад візуалізації мережі документів за допомогою програмного комплексу Gephi [11], у середовищі якого реалізовано кластеризацію графів за класами модулярності.

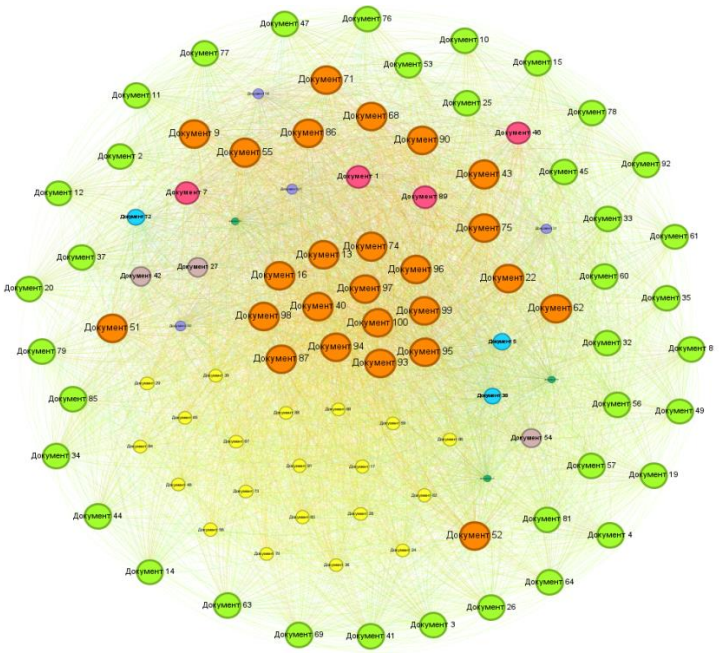

Рис. 3. Візуалізація мережі документів за допомогою Gephi

Висновки

Проаналізовано застосування великих мовних моделей (LLM) для екстракції концептів і зв'язків з документів у сфері кібербезпеки, семантичного індексування, кластерного аналізу мереж концептів у мережі документів. Наведено приклад успішної інтеграції сучасних технологій (LLM) з традиційними методами аналізу тексту (семантичне індексування, кластерний аналіз).

Проведене дослідження демонструє дієвість використання великих мовних моделей для автоматичної обробки текстових даних у сфері кібербезпеки. Побудовані семантичні мережі та застосування кластерного аналізу дозволили виявити складні взаємозв'язки між концептами і ефективно групувати документи за тематикою. Отримані результати відкривають нові можливості для автоматизації аналізу кіберзагроз і підтримки прийняття рішень.

Показано, що великі мовні моделі є потужним інструментом для автоматичного вилучення концептів і встановлення зв'язків між ними з текстових даних, особливо в галузі кібербезпеки. Побудовані семантичні мережі дозволяють візуалізувати і аналізувати складні взаємозв'язки між концептами, що є важливим для розуміння контексту кіберзагроз. Застосування кластерного аналізу за класами модулярності дозволило ефективно групувати документи за тематикою, що полегшує аналіз великих обсягів інформації.

Запропоновані методи дозволяють автоматизувати рутинні завдання з аналізу великих обсягів текстових даних, що звільняє аналітиків для виконання більш складних задач.

Результати дослідження можуть бути використані для створення інструментів підтримки прийняття рішень у галузі кібербезпеки, наприклад, систем раннього попередження про нові загрози. Запропонований підхід дозволяє глибше розуміти структуру та динаміку кіберзагроз, виявляти нові зв’язки та тренди.

1. Cheng Xiang Zhai. Large Language Models and Future of Information Retrieval: Opportunities and Challenges. SIGIR ‘24: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. Pages 481–490. DOI: 10.1145/3626772.365784.

2. H. Ambre Ayats. Knowledge graph construction from texts with an explainable, human-centered Artificial Intelligence. Artificial Intelligence [cs.AI]. Université de Rennes, 2023. English. NNT: 2023URENS095.

3. Michael Zgurovsky, Dmitry Lande, Kostiantyn Yefremov, Oleh Dmytrenko, Andriy Boldak, Artem Soboliev. Extracting and Identifying Relationships of Key Phrases in Information Flows. Published in: 2022 IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC) 04–07 October 2022. DOI: 10.1109/SAIC57818.2022.9923019.

4. Vinay Kumar Pant, Rupak Sharma, Shakti Kundu. An overview of Stemming and Lemmatization Techniques. In: Advances in Networks, Intelligence and Computing. CRC Press. 2024. ISBN: 9781003430421.

5. Ovalle, Anaelia, Ninareh Mehrabi, Palash Goyal, Jwala Dhamala, Kai-Wei Chang, Richard Zemel, Aram Galstyan, Yuval Pinter, and Rahul Gupta. Tokenization matters: Navigating data-scarce tokenization for gender inclusive language technologies. In Findings of the Association for Computational Linguistics: NAACL 2024, P. 1739–1756. 2024.

6. Basra Jehangir, Saravanan Radhakrishnan, Rahul Agarwal. A survey on Named Entity Recognition — datasets, tools, and methodologies. Natural Language Processing Journal. Volume 3, June 2023, 100017. DOI: 10.1016/j.nlp.2023.100017

7. Lande D., Subach I., Puchkov A. System of Analysis of Big Data from Social Media. *Information & Security: An International Journal*. **47**, Issue 1 (2020): 44–61. DOI: doi.org/10.11610/isij.4703.

8. Dmytro Lande, Leonard Strashnoy. GPT Semantic Networking: A Dream of the Semantic Web - The Time is Now. Kyiv: Engineering, 2023. 168 p. ISBN 978-966-2344-94-3.

9. Traag V.A., Waltman L., N. j. van Eck. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep 9, 5233 (2019). DOI: 10.1038/s41598-019-41695-z.

10. Wu F.Y. The Potts model. *Rev. Mod. Phys*. 54, 235. Published 1 January 1982.

11. Ken Cherven. Mastering Gephi Network Visualization. Packt Publishing, 2015. 378 p.

Надійшла до редакції 16.09.2024