

УДК 342.3:004.89

ЛАНДЕ Д.В., доктор технічних наук, професор, керівник наукового центру цифрової трансформації і права Державної наукової установи «Інститут інформації, безпеки і права Національної академії правових наук України», завідувач кафедри КПІ ім. Ігоря Сікорського.

ORCID: <https://orcid.org/0000-0003-3945-1178>.

ФУРАШЕВ В.М., кандидат технічних наук, с.н.с., перший заступник директора з наукової роботи Державної наукової установи «Інститут інформації, безпеки і права Національної академії правових наук України».

ORCID: <https://orcid.org/0000-0001-7205-724X>

ЗАСТОСУВАННЯ ФРЕЙМВОРКУ БЕЗКОДОВОГО ПРОГРАМУВАННЯ ПРИ ВИРІШЕННІ ЗАВДАНЬ ПАРЛАМЕНТСЬКОГО КОНТРОЛЮ

DOI: [https://doi.org/10.37750/2616-6798.2025.2\(53\).334055](https://doi.org/10.37750/2616-6798.2025.2(53).334055)

Анотація: У статті розглядається застосування фреймворку безкодового програмування у процесах парламентського контролю в умовах цифровізації. Автори пропонують формалізовану методологію аналізу правових документів, засновану на використанні великих мовних моделей (ВММ) та примітивів безкодової логіки — таких як "умова", "цикл", "функція", "перехід" і "мітка". Такий підхід дозволяє автоматизувати процеси аналізу законопроектів, виявлення ризиків, суперечностей з чинним законодавством та корупційних загроз. Стаття демонструє практичне застосування запропонованого фреймворку через кейси аналізу законопроектів та звітів виконавчої влади, що забезпечує стандартизацію, ефективність та масштабованість парламентського контролю. Особливо актуальною ця методологія є в умовах посилення децентралізації, тривалого воєнного стану та швидкого розвитку цифрових технологій.

Ключові слова: парламентський контроль, штучний інтелект, велика мовна модель, безкодове програмування, аналіз законопроектів, методологія.

Abstract: The article examines the application of a no-code programming framework in parliamentary control processes under digitalization. The authors propose a formalized methodology for analyzing legal documents, based on the use of large language models (LLMs) and no-code logic primitives — such as "condition," "loop," "function," "transition," and "label." This approach enables the automation of draft legislation analysis, identification of risks, inconsistencies with current legislation, and corruption threats. The article demonstrates the practical application of the proposed framework through case studies involving the analysis of draft laws and reports from the executive branch, ensuring standardization, efficiency, and scalability of parliamentary oversight. This methodology is particularly relevant in the context of increasing decentralization, prolonged martial law, and the rapid development of digital technologies.

Keywords: parliamentary control, artificial intelligence, large language model, no-code programming, legislation analysis, methodology.

Вступ. Сучасні умови здійснення парламентського контролю характеризуються посиленням децентралізації владних повноважень, прискореними темпами цифровізації та тривалим воєнним станом, викликаним агресією путінської росії проти нашої країни. У цих обставинах ефективність системи контролю значно ускладнюється, що потребує не лише модернізації правових механізмів, а й впровадження новітніх технологій для

автоматизації аналітичних процесів. Таким чином, здійснення парламентського контролю в умовах децентралізації влади та сучасних темпів цифровізації, як вже відзначалося у наукових роботах [1-5], справа досить непроста та трудомістка.

У рамках наукового дослідження за темою “Організаційно-правові і технологічні засади розвитку системи нормативно-правової інформації та парламентського контролю в умовах цифровізації” було встановлено, що однією з ключових можливостей підвищення ефективності парламентської діяльності є використання великих мовних моделей (ВММ), які працюють за принципом людського мозку, здатні аналізувати та генерувати тексти, а також прогнозувати наслідки прийняття рішень.

Велика мовна модель (ВММ) — це тип алгоритму штучного інтелекту (ШІ), який ґрунтується на методах глибокого навчання та опирається на великі масиви даних для розуміння, узагальнення, створення та прогнозування нового контенту. Термін “генеративний ШІ” тісно пов’язаний із ВММ, оскільки це тип ШІ, спеціально створений для генерації текстового контенту.

Впровадження таких систем має особливе значення через їхню здатність:

- обробляти великі обсяги даних;
- автоматизувати рутинні процеси;
- забезпечувати якісний аналіз інформації;
- прогнозувати наслідки прийняття рішень.

Особливо важливим є використання фреймворку безкодового програмування, який дозволяє формалізувати логічні конструкції, стандартизувати результати та забезпечити повторне використання типових шаблонів. Це робить процеси аналізу доступними для широкого кола користувачів, серед яких переважають правники, а не технічні фахівці.

У сучасних наукових дослідженнях [6] висвітлюється потенціал використання ВММ у законодавчому процесі, зокрема у формуванні, аналізі та оцінці проектів правових актів. Дослідження [7] акцентують важливість стандартизації аналітичних процесів через формальні методи програмування, особливо в умовах масового використання генеративних моделей.

Фреймворки безкодового програмування, як показано в [8], забезпечують можливість побудови складних логічних конструкцій без участі технічних спеціалістів, що робить їх ідеальними для використання в парламентських установах, де домінують правники.

Методологія

Для вирішення завдань парламентського контролю використано фреймворк безкодового програмування на основі промпт-інжинірингу, описаний у [9]. Цей фреймворк базується на трьох основних примітивах:

- Condition (If-Else) – для розгалужень.
- Loop (For-Loop) – для повторення дій.
- Function (Абстракція) – для узагальнення операцій.

Ці примітиви використовуються для побудови промптів, які автоматизують аналіз текстів законопроектів, виявлення ризиків, виявлення суперечностей із чинним законодавством, пошук корупційних загроз, формування запитів до органів влади, аналіз громадської думки тощо.

Основна застосування цього фреймворку полягає у представленні промптів як математично формалізованих аналогів базових конструкцій мов програмування (умови, цикли, функції). Це перетворює довільні текстові запити на структуровані логічні примітиви, які можна систематично комбінувати, оптимізувати та аналізувати. Наприклад, умова (If-Else) набуває форми математичної функції з вхідними

параметрами та вихідними діями, цикл (For-Loop) моделюється як оператор багаторазової обробки, а функція (Абстракція) визначається як процес перетворення даних із параметрами.

Такий підхід дозволяє створювати складні логічні системи через текстові інструкції, що узгоджується з парадигмою “програмування за прикладом” (PBE), але з більшою точністю та контролем.

Запропонована “мова промптів” відрізняється від традиційних предметно-орієнтованих мов гнучкістю та відсутністю жорсткого синтаксису. Вона має властивості модульності, параметризації та композиційності. У межах цієї мови кожен промпт виконує атомарне завдання (наприклад, “витягти терміни” чи “побудувати зв’язки”), змінні, взяті у фігурні дужки, дозволяють адаптувати логіку без зміни основної структури, а промпти можна об’єднувати в ланцюжки чи графи завдань, аналогічно функціям у програмах.

Це також зменшує поріг входу до програмування для користувачів, оскільки не вимагає навичок написання коду, водночас забезпечуючи гнучкість для створення складних систем.

Опишемо фреймворк детальніше. Суть запропонованого підходу для створення систем полягає у представленні промптів як аналогів програмних конструкцій (умовні оператори, цикли, функції) шляхом математичної формалізації їх логіки та взаємодій. Для формування промптів застосовуються такі основні примітиви (“цеглинки”), такі як “Умова”, “Цикл” і “Функція”, а також методи композиції цих примітивів для побудови складних систем, зокрема семантичних мереж.

Опишемо фреймворк, на основі якого у подальшому будуть формуватись промпти великими мовними моделями, які надалі будуть застосовуватись для формування семантичних мереж, виконуючись, у свою чергу, у середовищі великих мовних моделей.

1. Примітив “Умова” (If-Else)

Нехай:

- Input – вхідні дані (наприклад, текст, числовий параметр тощо).
- C – умова (предикат), що повертає значення True або False .
- A1, A2 – дві можливі дії, які виконуються у випадках, коли умова дорівнює True або False відповідно.

Промпт-функція P визначається як:

$P(\text{Input})=A1$, якщо $C(\text{Input})=\text{True}$

$P(\text{Input})=A2$, якщо $C(\text{Input})=\text{False}$

Це формальне визначення аналогічне класичному оператору if-else у мовах програмування, але застосовується через текстові інструкції.

Механізм роботи примітива “Умова” полягає в тому, щоб передати LLM задачу з двома можливими сценаріями, один з яких виконується в залежності від результату перевірки умови. Умова може бути будь-якою логічною конструкцією, наприклад:

- Перевірка наявності певного слова у тексті.
- Порівняння чисел.
- Аналіз контексту або категорії даних.

Приклад:

“Якщо текст містить термін “кібербезпека”, поверни його визначення; інакше – список споріднених термінів. ”

Тут:

- C – наявність слова “кібербезпека” у тексті.

- A1 – генерація визначення терміну “кібербезпека”.
- A2 – пошук асоціацій до терміну “кібербезпека”.

Цей примітив дозволяє моделі приймати рішення на основі заданих критеріїв, що ефективно моделює логічну умову.

2. Примітив “Цикл” (For-Loop)

Нехай:

- $S = \{s_1, s_2, \dots, s_n\}$ – множина елементів, які потрібно обробити.
- F – операція, яка застосовується до кожного елемента множини.

Промпт-функція $P(S)$ визначається як об’єднання результатів застосування операції F до кожного елемента множини:

$$P(S) = \bigcup_{i=1, \dots, n} F(s_i)$$

Це формальне визначення аналогічне класичному циклу `for` у мовах програмування, де одна і та ж операція повторюється для всіх елементів множини.

Механізм роботи примітива “Цикл” полягає в тому, щоб передати LLM список елементів разом із операцією, яку потрібно застосувати до кожного елемента. Результатом є об’єднання всіх отриманих відповідей. Такий підхід корисний для масового аналізу даних, виділення паттернів або генерації відповідей для декількох входів одночасно.

Приклад:

“Для кожного терміну зі списку [“фішинг”, “брандмауер”] знайди 3 приклади його використання.”

Тут:

- $S = \{\text{“фішинг”}, \text{“брандмауер”}\}$ – множина термінів.
- F – операція знаходження прикладів використання терміну.

Результатом буде об’єднання прикладів для обох термінів.

3. Примітив “Функція” (Abstraction)

Нехай:

- $F: X \rightarrow Y$ – функція, яка перетворює елементи з множини X у елементи множини Y .
- x – вхідний елемент.
- параметр – набір параметрів, які регулюють поведінку функції.

Промпт-функція $F_{\text{extract}}(x, \text{параметр})$ реалізується через інструкцію, яка передається моделі:

$$F_{\text{extract}}(x, \text{параметр}) = \text{Prompt}(x, \text{інструкція з параметр})$$

Це формальне визначення аналогічне абстракції у мовах програмування, де функція може приймати параметри для регулювання своєї логіки.

Механізм роботи примітива “Функція” полягає в тому, щоб передати LLM конкретну задачу з параметрами, які визначають деталі її виконання. Це дозволяє створювати гнучкі інструкції, які можуть бути повторно використані для різних входів.

4. Композиція примітивів у системи

Система — це оркестрація примітивів, аналогічна програмам. Синтаксис промтів утворює мову з формальною граматикою, яку можна представити у вигляді:

$$\text{Prompt} ::= \text{Примітив} \mid (\text{Prompt} \oplus \text{Prompt}) \mid \text{Умова}(\text{Prompt}, \text{Prompt})$$

де \oplus — композиція.

Ці три примітиви — “Умова”, “Цикл” та “Функція” — утворюють основу для безкодового створення систем через промпт-інжиніринг, дозволяючи використовувати природну мову як інструмент для управління логікою AI-систем.

Множина задач парламентського контролю. На основі аналізу потреб парламентських комітетів, профільних груп та експертів виявлено широкий спектр задач, які можуть бути автоматизовані або підтримані через застосування фреймворку безкодового програмування. Серед них:

- Виявлення ризиків і суперечностей у текстах законопроектів
- Генерація стислих резюме для депутатів
- Аналіз звітів виконавчої влади на відповідність чинному законодавству
- Формування запитів до органів влади для уточнення інформації або звітності
- Виявлення можливих лобістських інтересів у законодавчих ініціативах
- Пошук формулювань (повторюваних), що вказують на лобіювання певних інтересів
- Аналіз громадської думки щодо певних законодавчих ініціатив
- Автоматичний пошук рішень, які використовуються в інших країнах
- Генерація рекомендацій на основі найкращих світових практик
- Контроль за виконанням доручень парламенту
- Аналіз бюджетних видатків на відповідність затвердженому плану
- Тематичний аналіз парламентських запитань
- Прогнозування наслідків прийняття рішень
- Пошук аналогів у міжнародному досвіді
- Виявлення актуальності нормативно-правового регулювання
- Перевірка на відповідність міжнародним стандартам
- Виявлення корупційних загроз у звітах
- Формування експертних полягань
- Моніторинг реалізації держпрограм
- Оцінка ефективності законодавчих змін

Ці задачі можуть бути реалізовані через промпти до систем штучного інтелекту, формалізовані через примітиви фреймворку, що забезпечує повторність, стандартизацію та масштабованість аналітичних процесів.

Приклади використання фреймворку безкодового програмування

Приклад 1. Автоматичне виявлення ризиків, суперечностей і корупційних загроз у тексті законопроекту

Для вирішення цієї задачі складається первинний запит (промпт), який має вигляд:

“Ви – експерт із законодавства та боротьби з корупцією. Ваше завдання – проаналізувати текст законопроекту на наявність:

1. Можливих ризиків (економічних, соціальних, правових).
2. Суперечностей із чинним законодавством.
3. Потенційних корупційних загроз (наприклад, надмірно вигідні умови для окремих осіб чи груп, лобістські формулювання тощо).

Проаналізуйте текст законопроекту [вставте текст] і надайте:

- список знайдених ризиків із поясненнями;
- перелік суперечностей із чинним законодавством (якщо є, зазначте, з якими саме законами);
- Опис можливих корупційних загроз із прикладами фрагментів тексту.

Висновки подайте у форматі:

1. Ризики: [список]
2. Суперечності: [список]
3. Корупційні загрози: [список із поясненнями]”

Ось текст законопроекту: [Тест]

Складений промпт вже може виконуватись у середовищі ВММ, але відповіді, що він буде давати, мають слабко структурований, нечіткий вигляд, не зрозуміло, якою мірою враховано інформацію первинного документа. Для перетворення цього документа у структурований промпт, до ВММ завантажуються текст фреймворку безкодового програмування:

У відповідності із наведеним фреймворком перетвори промпт для автоматичного виявлення ризиків, суперечностей чи корупційних загроз у тексті законопроекту. Ось первинний промпт: [промпт]

У відповідності з фреймворком безкодового промпт-інжинірингу, пропонуються наступні кроки для реалізації задачі через формальні примітиви:

1. Використання примітива “Функція” для аналізу ризиків

Формалізація:

Нехай:

- $F_risk: X \rightarrow Y$ — функція для виявлення ризиків.
- X — текст законопроекту.
- параметр = {“тип_ризиків”: [“економічні”, “соціальні”, “правові”]}.
- $F_risk(X, \text{параметр})$ повертає список ризиків із поясненнями.

Промпт-функція:

“Виділи з тексту [X] всі можливі ризики типів: {тип_ризиків}.

Для кожного ризику:

- опиши його сутність;
- поясни потенційні наслідки;
- наведи цитату з тексту законопроекту, що його викликає.”

2. Використання примітива “Функція” для аналізу суперечностей

Формалізація:

Нехай:

- $F_conflict: X \rightarrow Z$ — функція для виявлення суперечностей із чинним законодавством.
- параметр = {“джерела_законодавства”: [список_законів]}.
- $F_conflict(X, \text{параметр})$ повертає перелік суперечностей.

Промпт-функція:

“Перевір текст [X] на відповідність законам: {джерела_законодавства}.

Для кожної суперечності:

- зазнач назву закону;
- опиши конфліктну норму у законопроекті;
- наведи цитату з тексту.”

3. Використання примітива “Функція” для аналізу корупційних загроз

Формалізація:

Нехай:

$F_corruption: X \rightarrow W$ — функція для виявлення корупційних загроз.

- параметр = {“ознаки_корупції”: [“надмірні_вигоди”, “лобістські_формулювання”, “відсутність_контролю”]}.
- $F_corruption(X, \text{параметр})$ повертає список загроз із прикладами.

Промпт-функція:

“Аналізуй текст [X] на наявність ознак корупції: {ознаки_корупції}.

Для кожної ознаки:

- опиши, як вона проявляється у тексті;
- наведи цитату з законопроекту;
- поясни, хто може скористатися перевагами. ”

4. Композиція примітивів у систему

Формалізація:

Система S реалізується через послідовне застосування трьох функцій:

$$S(X) = F_risk(X, \{\text{“тип_ризиків”}: [\dots]\}) \oplus F_conflict(X, \{\text{“джерела_законодавства”}: [\dots]\}) \oplus F_corruption(X, \{\text{“ознаки_корупції”}: [\dots]\})$$

де \oplus – операція об’єднання результатів у структурований вивід.

У результаті формальний промпт для ВММ має вигляд:

Ви експерт із безкодового аналізу текстів. Виконайте завдання через три кроки:

****Крок 1. Аналіз ризиків****

Використовуй функцію F_risk для виявлення економічних, соціальних та правових ризиків.

****Крок 2. Аналіз суперечностей****

Застосуй F_conflict для перевірки відповідності законам: [перелік_законів].

****Крок 3. Аналіз корупційних загроз****

Виконай F_corruption для пошуку надмірних вигод, лобістських формулювань тощо.

Результат подай у форматі:

1. Ризики: [список із поясненнями та цитатами]
2. Суперечності: [список із назвами законів та цитатами]
3. Корупційні загрози: [список із прикладами з тексту]

Функції:

1. F_risk: $X \rightarrow Y$ — функція для виявлення ризиків.

X — текст законопроекту.

параметр = {“тип ризиків”: [“економічні”, “соціальні”, “правові”]}.

F_risk(X, параметр) повертає список ризиків із поясненнями.

Промпт-функція:

“Виділи з тексту [X] всі можливі ризики типів: {тип_ризиків}.

Для кожного ризику:

- опиши його сутність;
- поясни потенційні наслідки;
- наведи цитату з тексту законопроекту, що його викликає. ”

2. F_conflict: $X \rightarrow Z$ — функція для виявлення суперечностей із чинним законодавством.

параметр = {“джерела законодавства”: [список законів]}.

F_conflict(X, параметр) повертає перелік суперечностей.

Промпт-функція:

“Перевір текст [X] на відповідність законам: {джерела законодавства}.

Для кожної суперечності:

- зазнач назву закону;
 - опиши конфліктну норму у законопроекті;
 - наведи цитату з тексту. ”

3. $F_{corruption}: X \rightarrow W$ — функція для виявлення корупційних загроз.
 параметр = {“ознаки_корупції”: [“надмірні_вигоди”, “лобістські формулювання”, “відсутність_контролю”]}.

$F_{corruption}(X, \text{параметр})$ повертає список загроз із прикладами.
 Промпт-функція:
 “Аналізуй текст [X] на наявність ознак корупції: {ознаки корупції}.”
 Для кожної ознаки:
 - опиши, як вона проявляється у тексті;
 - наведи цитату з законопроекту;
 - поясни, хто може скористатися перевагами. ”

Нижче наведено приклад застосування сформованого структурованого промπτу до Закону України “Про державне регулювання у сфері комунальних послуг” № 3610-VI від 07.07.2011. Відомості Верховної Ради України, 2010, № 49, ст. 571

1. Ризики:

Економічні ризики:

Нестабільність тарифів на комунальні послуги: Встановлення тарифів, нижчих за економічно обґрунтовані витрати, може призвести до фінансової неплатоспроможності підприємств.

Цитата: “Встановлення тарифів на комунальні послуги, нижчих за рівень економічно обґрунтованих витрат на їх виробництво, не допускається...” (ст. 10, ч. 4).

Перехресне субсидіювання: Можливе використання коштів з регульованих ринків для підтримки інших напрямів діяльності підприємств, що порушує конкуренцію.

Цитата: “Контроль за недопущенням перехресного субсидіювання під час провадження господарської діяльності суб'єктами природних монополій...” (ст. 6, п. 8).

Соціальні ризики:

Обмежений доступ до послуг: Невиконання суб'єктами монополій вимог щодо якості послуг може спричинити соціальну напруженість.

Цитата: “Забезпечення прогнозованості цінової і тарифної політики... стимулювання підвищення їх якості...” (ст. 3, п. 4).

Правові ризики:

Конфлікти між регуляторами: Розмиття компетенцій між Національною комісією та місцевими органами влади.

Цитата: “Національна комісія здійснює контроль за дотриманням законодавства Радою міністрів Автономної Республіки Крим...” (ст. 15, ч. 2).

2. Суперечності:

Порушення Закону України “Про природні монополії”:

Невідповідність у визначенні монополій: У Законі про природні монополії вказано, що монополії регулюються окремим органом, але в тексті досліджуваного закону ці повноваження передані Національній комісії.

Цитата: “Органом державного регулювання діяльності суб'єктів природних монополій є національна комісія...” (ст. 2, ч. 1; порівняйте з Законом “Про природні монополії”, ст. 4).

Порушення Закону України “Про місцеве самоврядування”:

Обмеження права місцевих рад встановлювати тарифи:

Уточнення, що тарифи на водопостачання та тепlopостачання встановлює Комісія, суперечить правам місцевого самоврядування.

Цитата: “Тарифи на послуги централізованого водопостачання... встановлюються національною комісією... ” (ст. 19, п. 5.3; порівняйте з Законом “Про місцеве самоврядування”, ст. 28).

3. Корупційні загрози:

Надмірні вигоди для регулятора:

Монополія на встановлення тарифів: Національна комісія має широкі повноваження щодо затвердження тарифів, що створює ризик корупційних домовленостей з підприємствами.

Цитата: “Встановлення тарифів на комунальні послуги суб’єктам природних монополій...” (ст. 6, п. 2).

Лобістські формулювання:

Привілеї для підприємств з нетрадиційної енергетики: Виключення з регулювання виробників енергії на основі поновлюваних джерел може бути лобістським тиском.

Цитата: “Тарифи на виробництво теплової енергії на установках з використанням нетрадиційних або поновлюваних джерел енергії не регулюються...” (ст. 1, п. 3).

Слабкий механізм відшкодування шкоди:

Невизначеність у порядку відшкодування шкоди через неправомірні рішення Комісії.

Цитата: “Збитки... підлягають відшкодуванню у порядку, передбаченому законодавством” (ст. 18, ч. 2), але конкретний механізм відсутній.

Примітка: Аналіз засновано на тексті закону з урахуванням змін до 2022 року.

Приклад 2. Генерація коротких резюме законопроектів для депутатів

Для реалізації цього завдання створюється первинний промпт, який після цього треба формалізувати:

Вам потрібно підготувати стисле, чітке та зрозуміле резюме для депутатів про законопроект. Законопроект стосується [вказіть тему, наприклад, “реформи системи охорони здоров’я” або “зміни у податковому законодавстві”].

Включіть наступну інформацію:

1. Назва законопроекту.
2. Мета законопроекту (1-2 речення).
3. Основні положення (3-5 пунктів).
4. Очікувані переваги чи результати (1-2 речення).
5. Можливі виклики чи ризики (за потреби).

Використовуйте максимально доступну мову, уникайте складної юридичної термінології. Резюме має бути не більше 200 слів.

Ось текст законопроекту:

На запит створення структурованого промпту для вирішення цього завдання ВММ Qwen, зокрема, видає рішення, що у відповідності з фреймворком безкодового промпт-інжинірингу пропонується наступна структура для автоматичного створення резюме законопроекту для депутатів:

Ти експерт із безкодового аналізу текстів. Виконай завдання через п’ять кроків:

****Крок 1. Витягни назву****

Застосуй F_title до тексту [X].

****Крок 2. Визнач мету****

Використовуй $F_purpose$ для формулювання мети.

****Крок 3. Виділи основні положення****

Розбий текст на розділи та опиши кожен у форматі "пункт: сутність" (3-5 пунктів).

****Крок 4. Оціни переваги****

Застосуй $F_benefits$ для опису очікуваних результатів.

****Крок 5. Перевір ризику****

Якщо законопроект містить суперечливі або нереалістичні положення, опиши їх.

Інакше — пропустіть.

Результат подай у форматі:

1. Назва: [рядок]
2. Мета: [1-2 речення]
3. Основні положення: [список пунктів]
4. Переваги: [1-2 речення]
5. Виклики/ризик: [список або "відсутні"]

Обмеження: не більше 500 слів, доступна мова, мінімум термінології.

1. Примітив "Функція" для витягування назви законопроекту

$F_title: X \rightarrow Y$ — функція для витягування назви.

X — текст законопроекту.

Y — рядок із назвою.

Промпт-функція:

"Виділи назву законопроекту з тексту $[X]$ і поверни її як один рядок."

2. Примітив "Функція" для визначення мети

$F_purpose: X \rightarrow Z$ — функція для формулювання мети.

параметр = {"довжина": "1-2 речення"}.

Промпт-функція:

"Сформулюй мету законопроекту $[X]$ у 1-2 реченнях, використовуючи просту мову."

3. Примітив "Цикл" для аналізу основних положень

$S = \{\text{розділ1, розділ2, ...}\}$ — множина розділів законопроекту.

$F_provision: s_i \rightarrow W$ — функція для стислого опису кожного розділу.

Результат: об'єднання описів ($W_1 \oplus W_2 \oplus \dots$).

Промпт-цикл:

"Для кожного розділу законопроекту $[X]$:

1. Виділи головну ідею.
2. Сформулюй її як короткий пункт (3-5 пунктів загалом)."

4. Примітив "Функція" для опису переваг

$F_benefits: X \rightarrow V$ — функція для оцінки переваг.

параметр = {"довжина": "1-2 речення"}.

Промпт-функція:

"Опиши очікувані переваги законопроекту $[X]$ у 1-2 реченнях, уникай термінології."

5. Примітив "Умова" для перевірки ризиків

$C_risk: X \rightarrow \{\text{True, False}\}$ — умова наявності ризиків.

A1: Якщо True, застосуй F_risks .

A2: Якщо False, поверни порожній список.

Промпт-умова:

"Перевір, чи законопроект $[X]$ містить елементи, що можуть викликати виклики або

ризика.
 Якщо так:
 - Опиши їх коротко (за потреби).
 Якщо ні:
 - Пропустіть цей розділ. ”

Після опрацювання формалізованого промпту на прикладі Закону України “Про хміль та хмелепродукти” від 27 березня 2025 року № 4341-ІХ отримуємо результат:

Назва: Закон України “Про хміль та хмелепродукти”

Мета: Регулювання вирощування, сертифікації, переробки та торгівлі хмелем і хмелепродуктами для забезпечення якості, підтримки українських виробників, відповідності міжнародним стандартам та стимулювання експорту.

Основні положення:

Сертифікація: Встановлення правил видачі сертифікатів на хміль і хмелепродукти, включаючи вимоги до документації (номер сертифіката, опис продукту, країна походження).

Контроль за переробкою: Обов’язок ведення обліку перероблених матеріалів, маркування упаковки під наглядом сертифікаційних центрів.

Імпорт: Вимоги до сертифікації іноземних партій хмелю, включаючи атестат еквівалентності від закордонних органів.

Відповідальність: Штрафи за порушення вимог щодо підготовки, маркування або сертифікації продукції (від 5 до 10 мінімальних зарплат).

Технічні стандарти: Визначення термінів (наприклад, “закритий операційний цикл”) та вимог до якості хмелю (вміст альфа-кислот).

Переваги: Покращення якості продукції, збільшення конкурентоспроможності українського хмелю на міжнародному ринку, уніфікація процедур сертифікації, підтримка експортерів через відповідність стандартам ЄС і США.

Виклики/ризика: Відсутні. Документ передбачає чіткі механізми реалізації, включаючи контроль, сертифікацію та штрафні санкції, без суперечливих положень.

Приклад 3. Аналіз звітів виконавчої влади на відповідність законодавству

Для аналізу звітів виконавчої влади на відповідність законодавству створюється первинний промпт:

Проаналізуй звіти або їх окремі розділи, підготовлені виконавчою владою, на відповідність прийнятим законам. Визнач, чи дотримані вимоги законодавства, чи виконані положення відповідних нормативно-правових актів, та чи є відхилення або невідповідності. Зроби акцент на виявленні:

1. Фактів неповного виконання законів.
2. Ризиків, що виникають через відхилення від норм.
3. Рекомендацій для забезпечення повної відповідності звітів законодавчим нормам.

Надай чіткий висновок щодо відповідності кожного розділу звіту конкретним законодавчим актам та пропозиції щодо покращення.

Ось текст звіту: [Текст]

Структурований промпт, що створюється ВММ у відповідності із завантаженим фреймворком має вигляд:

Проаналізуйте звіт або окремі його розділи, підготовлені виконавчою владою.

Виявіть:

Факти неповного виконання законів .

Ризики , що виникають через порушення норм (економічні, соціальні, правові).

Рекомендації для забезпечення повної відповідності звітів законодавству.

Формат відповіді

1. Відповідність законодавству

Примітив: Цикл (For-Loop)

Для кожного розділу звіту:

Нормативно-правовий акт: Вкажіть конкретний закон або статтю (наприклад, Закон України “Про державне регулювання у сфері комунальних послуг”, ст. 10).

Вимога закону: Наведіть текст норми, яка регулює цю сферу.

Оцінка виконання (Примітив: Умова):

- Відповідність: [Короткий опис виконання].

- Порушення: [Опис невідповідності], [Цитата зі звіту], [Посилання на порушений закон].

2. Суперечності з законодавством

Примітив: Функція (Abstraction)

Для кожного виявленого порушення:

Назва закону: Вкажіть закон, який порушено.

Конфліктна норма у звіті: Опишіть, як формулювання звіту суперечить закону.

Цитата зі звіту: Наведіть текст звіту, що підтверджує суперечність.

3. Корупційні загрози

Примітив: Функція (Abstraction)

Для кожного розділу звіту:

Ознака корупції: Вкажіть тип (надмірні вигоди, лобістські формулювання, відсутність контролю).

Опис прояву: Як ця ознака реалізується у тексті?

Цитата зі звіту: Наведіть фрагмент, що підтверджує загрозу.

Хто може скористатися: Вкажіть групу або особу, які отримують переваги.

4. Рекомендації

Примітив: Функція (Abstraction)

Для кожного виявленого порушення або ризику:

Пропозиція: Як усунути невідповідність?

Обґрунтування: На якому законі або нормі базується?

5. Загальний висновок

Примітив: Цикл + Умова

Підсумуйте:

Чи відповідає звіт законодавству?

Які розділи мають критичні невідповідності?

Які ризики найбільш суттєві?

Які кроки пріоритетні для усунення порушень?

Формалізація примітивів у промті

Цикл (For-Loop):

$S = \{\text{розділи звіту}\}$ — множина елементів для аналізу.

$F = \text{“перевірка відповідності законодавству”}$ — операція для кожного елемента.

Умова (If-Else):

$C = \text{“чи вимога закону виконана?”} \rightarrow \text{True/False.}$

A_1 = “опис відповідності”, A_2 = “опис порушення”.

Функція (Abstraction):

$F_{risk}(X, \text{параметр} = \{\text{“тип_ризиків”}: [\text{“економічні”}, \text{“соціальні”}, \text{“правові”}]\}) \rightarrow$ аналіз ризиків.

$F_{corruption}(X, \text{параметр} = \{\text{“ознаки_корупції”}: [\text{“надмірні_вигоди”}, \text{“лобістські_формулювання”}, \text{“відсутність_контролю”}]\}) \rightarrow$ виявлення загроз.

Вхідні дані для моделі

Текст звіту: [Вставте текст звіту].

Перелік законів: [Закон України “Про державне регулювання у сфері комунальних послуг”, Закон України “Про природні монополії”, Кодекс України про адміністративні правопорушення].

Вихідний формат

1. Відповідність законодавству:

- Розділ [X]:
- Нормативно-правовий акт: [Назва закону, стаття].
- Вимога: [Цитата з закону].
- Оцінка: [Опис, цитата зі звіту].

2. Суперечності з законодавством:

- Закон: [Назва закону].
- Конфліктна норма: [Опис невідповідності].
- Цитата: [Фрагмент звіту].

3. Корупційні загрози:

- Ознака: [Тип загрози].
- Опис: [Як реалізується].
- Цитата: [Фрагмент звіту].
- Хто виграє: [Група/особа].

4. Рекомендації:

- Пропозиція: [Дія].
- Обґрунтування: [Посилання на закон].

5. Загальний висновок:

- Відповідність: Так/Ні.
- Критичні розділи: [Перелік].
- Пріоритетні дії: [Перелік рекомендацій].

Результат виконання цього промпту у середовищі BMM Qwen для звіту Державної аудиторської служби України про здійснення регуляторної діяльності у 2021 році має вигляд:

1. Відповідність законодавству

Розділ 1: Вступна частина

Нормативно-правовий акт: Закон України “Про засади державної регуляторної політики у сфері господарської діяльності”, ст. 8.

Вимога закону: “До ухвалення регуляторного акту обов’язково проводиться аналіз його регуляторного впливу”.

Оцінка:

Порушення: Проекти двох регуляторних актів (пункти 21 та 22 Плану розробки актів) розроблені без проведення аналізу регуляторного впливу, оскільки перебували на стадії узгодження з Мінфіном. Цитата зі звіту: “відповідні заходи з регуляторної політики, визначені Законом, щодо них не проводились”.

Посилання на порушений закон: Ст. 8 Закону України “Про засади державної регуляторної політики”.

Розділ 2: Загальна частина

Нормативно-правовий акт: Закон України “Про засади державної регуляторної політики”, ст. 4, ст. 9.

Вимога закону: “Регуляторні акти мають відповідати міжнародним зобов’язанням України” (ст. 4); “публічне обговорення проєктів регуляторних актів” (ст. 9).

Оцінка:

Порушення: Проєкт закону про віддалений контроль був узгоджений з Держрегуляторною службою, але після зауважень Мін’юсту щодо невідповідності Конвенції про права людини, Мінфін повернув документ для доопрацювання. Цитата зі звіту: “через таку позицію Мінфіну... подальшу роботу над законопроектом припинено”.

Посилання на порушений закон: Ст. 4, ст. 9 Закону України “Про засади державної регуляторної політики”.

2. Суперечності з законодавством

Закон: Закон України “Про засади державної регуляторної політики”, ст. 8.

Конфліктна норма у звіті: Відсутність аналізу регуляторного впливу для проєктів актів, що порушує вимогу ст. 8.

Цитата зі звіту: “відповідні заходи з регуляторної політики, визначені Законом, щодо них не проводились”.

Закон: Закон України “Про засади державної регуляторної політики”, ст. 4.

Конфліктна норма у звіті: Неврахування положень Конвенції про права людини у проєкті закону.

Цитата зі звіту: “Мінфін листом від 18.11.2021 № 33010-03-5/35457 повернув законопроект для доопрацювання з метою забезпечення врахування положень зазначеної Конвенції”.

3. Корупційні загрози

Ознака: Відсутність контролю.

Опис прояву: Відсутність аналізу регуляторного впливу для проєктів актів створює ризик ухвалення норм, що не відповідають реальним потребам або міжнародним стандартам.

Цитата зі звіту: “відповідні заходи з регуляторної політики, визначені Законом, щодо них не проводились”.

Хто може скористатися: Лобістські групи, які могли б вплинути на формулювання актів без публічного обговорення.

4. Рекомендації

Пропозиція: Обов’язкове проведення аналізу регуляторного впливу для всіх проєктів актів перед їх узгодженням.

Обґрунтування: Ст. 8 Закону України “Про засади державної регуляторної політики”.

Пропозиція: Перед поданням проєктів законів на розгляд Уряду — перевірка відповідності міжнародним зобов’язанням України.

Обґрунтування: Ст. 4 Закону України “Про засади державної регуляторної політики”.

5. Загальний висновок

Відповідність: Ні.

Критичні розділи: Розділ 1 (не проведення аналізу регуляторного впливу), Розділ 2 (порушення вимог щодо міжнародних зобов'язань).

Пріоритетні дії:

Забезпечити обов'язковий аналіз регуляторного впливу для всіх проєктів актів.

Впровадити механізми перевірки відповідності проєктів законів міжнародним нормам перед їх поданням на узгодження.

Результати. Як можна бачити, запропонований підхід для вирішення задач парламентського контролю має такі переваги, як чіткість, гнучкість, композиція різних висновків у єдиний звіт. При цьому кожний примітив виконує окрему підзадачу, що відповідає принципам фреймворку; параметри (тип ризиків, джерела законодавства тощо) можна адаптувати під конкретний законопроект, результати декількох функцій об'єднуються в єдиний вивід, що моделює складну систему аналізу. Сформовані структури дозволяють автоматизувати аналіз тексту документів через формальні примітиви, уникнувши неоднозначностей традиційних промптів до ВММ.

Застосування фреймворку безкодового програмування до задач парламентського контролю показало:

- Ефективність: автоматизовані промпти скорочують час аналізу від кількох годин до кількох хвилин.
- Стандартизація: всі результати подаються у єдиному форматі, що полегшує їхню подальшу обробку.
- Точність: використання формальних примітивів забезпечує чіткість виводу та мінімізує суб'єктивність.
- Масштабованість: один і той самий шаблон може використовуватися для аналізу різноманітних документів.

Результати тестування на конкретних прикладах демонструють високу практичну цінність такого підходу. На основі проведеного аналізу пропонується:

- створити бібліотеку типових промптів для парламентського контролю;
- розробити графічний інтерфейс для вбудовування фреймворку в існуючі парламентські системи;
- інтегрувати внутрішні правові бази даних для підвищення точності аналізу;
- забезпечити тренінги для парламентаріїв щодо використання інструментів ШІ;
- встановити етичні та правові межі використання ШІ в парламентському контролі.

Висновки. Застосування фреймворку безкодового програмування в парламентському контролі надає можливість ефективно автоматизувати аналіз правових документів, виявляти потенційні ризики, юридичні суперечності та корупційні загрози. Використання формальних примітивів (умова, цикл, функція, мітка, перехід) забезпечує структурований підхід до обробки тексту, що робить процес повторним, стандартизованим і масштабованим. Це особливо важливо в сучасних умовах, коли необхідно оперативно реагувати на великі обсяги нормативно-правових актів.

Дослідження показало, що впровадження такого фреймворку може значно скоротити час на аналіз документів, підвищити точність виявлення порушень та забезпечити об'єктивність висновків. Крім того, формалізація логіки аналізу дозволяє легко адаптувати систему під різні категорії завдань — від перевірки відповідності законопроектів чинним нормам до оцінки звітів виконавчої влади.

Подальші дослідження можуть бути спрямовані на:

Створення бібліотеки типових промптів для різних сценаріїв парламентського контролю.

Інтеграцію фреймворку з внутрішніми правовими базами даних, що забезпечить більш глибоку верифікацію відповідності законодавству.

Розробку графічного інтерфейсу для простоти використання інструменту непрограмістами.

Тренінгову підготовку парламентаріїв для ефективного використання інструментів ШІ у повсякденній роботі.

Встановлення етичних і правових меж використання штучного інтелекту в парламентському контролі, щоб забезпечити прозорість, справедливість та захист даних.

Отже, запропонований фреймворк має потенціал суттєво підвищити ефективність парламентської діяльності, зробити її більш аналітичною, стратегічною та відкритою для суспільства.

Використана література

1. Інформаційне та соціально-правове моделювання: посібник /Ланде Д.В., Фурашев В.М., за заг. ред. Ланде Д.В. - Київ-Одеса: Фенікс, 2021, 282 с.
2. Парламентський контроль в умовах децентралізації державної влади та цифрової трансформації в Україні: стан і проблеми: Матеріали Першої всеукраїнської науково-практичної конференції / м. Київ, 30 березня 2021 р. / упор.: Фурашев В. М., Дорогих С.О. Київ, 2021. 184 с.
3. Ланде Д.В., Фурашев В.М., Брайчевський С.М. Інформатика парламентського контролю: навчальний посібник / Д.В. Ланде, В.М.Фурашев, Брайчевський С.М. Київ-Одеса: Фенікс, 2022. 226 с.
4. Ланде Д.В., Фурашев В.М. Парламентський контроль із застосуванням генеративного штучного інтелекту: монографія / Ланде Д.В., Фурашев В.М. Київ: ТОВ «Інжиніринг», 2023. 202 с.
5. Нормативно-правова інформація і парламентський контроль: матеріали наук.-практ. конф. (Київ, 21 верес. 2023 р.) / наук. керівник конф.: Д. В. Ланде ; упор.: В. М. Фурашев, А. І. Нижник, С. О. Дорогих ; ДНУ «Ін-т інформ., безпеки і права Нац. академії правов. наук України». Київ, 2023. 118 с.
6. Nadja M. The impact of artificial intelligence on legal systems: challenges and opportunities. Проблеми законності, 2024 (164), 285-303.
7. Brown T. et al. (2020). Language Models are Few-Shot Learners . arXiv preprint arXiv:2005.14165. DOI: 10.48550/arXiv.2005.14165
8. Zhong, C., Oruongo, J., & Kim, J. J. B. (2025). LLM-Powered Low-Code/No-Code Data Analytics in Education and Workforce Development. Computer, 58(3), 49-59.
9. Dmitry Lande, Leonard Strashnoy. Semantic AI Framework for Prompt Engineering. SSRN Preprint: <https://ssrn.com/abstract=5172867>.

~~~~~ \* \* \* ~~~~~