УДК 004.91
DOI https://doi.org/10.32782/2663-5941/2026.3.1/24

Нікітін В.С.
https://orcid.org/0009-0001-1385-8714
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Гіоргізова-Гай В.Ш.
https://orcid.org/0000-0001-6224-3532
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПІДХІД ДО ГЕНЕРАЦІЇ ШАБЛОНІВ ДОКУМЕНТІВ ІЗ ЗАСТОСУВАННЯМ LLM АГЕНТІВ

У статті запропоновано підхід до автоматичної генерації шаблонів документів на основі зображень паперових або електронних форм із використанням великих мовних моделей та агентної обробки. Актуальність дослідження зумовлена потребою у стандартизації документообігу, зростанням обсягів неструктурованих даних і необхідністю скорочення часу на ручне проєктування електронних форм та шаблонізованих документів. Запропонований підхід забезпечує перетворення зображення документа на структурований JSON-шаблон, придатний для подальшого рендерингу у вигляді інтерактивної форми та шаблонізованого документа. Визначено основні вимоги до формату шаблону: безпека даних, легкість парсингу та валідації, а також придатність до генерації засобами LLM. Шаблон поділено на три складові: конфігурацію даних, структуру електронної форми та структуру фінального документа.

Архітектура системи включає вхідне зображення документа, приклад документа з еталонним JSON-шаблоном, доменно-специфічні інструкції, а також два LLM-агенти – FirstPass і Refiner. Перший агент формує початковий шаблон, а другий доопрацьовує його з урахуванням результатів автоматичної валідації.

Для оцінювання якості генерованих шаблонів використано порівняння з еталонними шаблонами та метрики структурної подібності. Експериментальне дослідження проведено на документах з кількох предметних областей, серед яких юридичні, військові та документи з охорони праці. Порівнювались результати роботи системи із застосуванням низки сучасних пропрієтарних і open-weight моделей та результати роботи двох LLM-агентів. Результати дослідження свідчать, що застосування двоетапної схеми генерації з подальшою валідацією дозволяє зменшити кількість структурних помилок у фінальному результаті. Імплементація запропонованого підходу дозволить прискорити цифровізацію документів, уніфікувати структуру форм, знизити навантаження на розробників та створити базу для подальшої автоматизації побудови електронних сервісів документообігу. Запропонований підхід може бути використаний як основа для розроблення гнучких систем генерації форм і документів у різних прикладних сферах.

Ключові слова: *великі мовні моделі, електронні форми, шаблони документів, автоматизація документообігу, генеративні користувацькі інтерфейс, інтелектуальні розподілені обчислення.*

Постановка проблеми. Питання стандартизації та автоматизації документообігу набувають особливої актуальності в контексті цифрової трансформації суспільства. Уніфіковані підходи до структурування документів є необхідною умовою їх сумісності, верифікації та довгострокового зберігання, тоді як автоматизація процесів генерації документів суттєво знижує операційні витрати й мінімізує вплив людського фактора [8, с. 10]. Дослідження цих аспектів відкриває можливості для розроблення ефективніших інструментів управління документацією у великих інформаційних системах.

Про актуальність даної проблеми свідчить, наприклад, активне впровадження та розширення функціоналу електронних рапортів у мобільному застосунку Армія+ [10], що будує документообіг за допомогою покрокових форм з чіткими правилами валідації, на противагу неструктурованим файлам у форматі *.docx.

Окремо слід підкреслити, що, згідно з дослідженнями останніх десяти років, близько 80% даних, створюваних різноманітними організаціями, належать до неструктурованих [2, с. 2], що суттєво ускладнює їх аналітичне опрацювання.

Сфери застосування шаблонів документів включають:

– юридичні документи (договори, заяви тощо): стандартизовані шаблони дозволяють швидко конструювати документи з повторюваних блоків і погоджених формулювань, знижуючи ризик юридично значущих помилок. Вони полегшують комплаєнс: зміни відслідковуються, версії зберігаються, а контрольні списки та перевірки інтегруються безпосередньо в процес підготовки;

– військові документи: у військовій сфері стандартизація документів набуває особливого значення через потребу в оперативному обміні інформацією та мінімізації двозначності. Застосунки на кшталт Армія+ демонструють, як структуровані електронні форми забезпечують швидке та точне заповнення звітів, рапортів та заяв навіть у польових умовах. Чітка структура та обмеження на формат даних знижують імовірність критичних помилок, полегшують автоматичну обробку великих обсягів документів та забезпечують можливість швидкого пошуку й аналізу інформації, що є життєво важливим для прийняття своєчасних управлінських рішень;

– документи з охорони праці (інструктажі тощо). Електронні форми допомагають системно збирати підтвердження проходження інструктажів, фіксувати події безпеки та підтримувати актуальність інструкцій. Шаблони гарантують повноту обов'язкових розділів, спрощують аудит відповідності і пришвидшують оновлення матеріалів при зміні нормативної бази.

Разом з тим, сучасні великі мовні моделі демонструють значний потенціал як інструмент автоматизації: зокрема, їхня здатність до генерації програмного коду істотно спрощує процес переходу від існуючих неструктурованих документів до стандартизованих форматів електронного документообігу [1, с. 9].

Аналіз останніх досліджень і публікацій. Розробник британського державного порталу GOV.UK, у двох публікаціях описав підхід до автоматизованого перетворення PDF-форм на веб-форми за допомогою великих мовних моделей [3; 7].

Запропонований інструмент приймає на вхід PDF-документи, зображення у форматах JPEG або PNG, а також рукописні ескізи форм, і генерує на виході структурований JSON-опис форми. Такий опис містить назву вихідного файлу, структуру сторінок із зазначенням кількості запитань на кожній із них, а також детальну специфікацію кожного запитання. Для кожного елемента фіксуються текст запитання, підказки для користувача, тип відповіді (число, дата, текст тощо) та можливі варіанти вибору. Процес генерації побудовано максимально просто й фактично зведено до одного виклику мовної моделі з використанням механізму «tools» для отримання структурованої відповіді.

Водночас запропоноване рішення має низку обмежень. Насамперед можна відзначити слабке відокремлення структури даних від їх подання: результат роботи системи подається як масив елементів інтерфейсу без унікальних ідентифікаторів, що може ускладнювати подальше опрацювання відповідей кінцевого користувача в межах більшої інформаційної системи. Крім того, у підході відсутні чітко визначені критерії оцінювання якості згенерованої форми: автор наводить приклади некоректної роботи системи, однак не пропонує формалізованих способів виявлення або запобігання подібним помилкам.

Як зазначають автори роботи [6, с. 2], генеративні підходи до побудови користувацьких інтерфейсів суттєво ускладнюють забезпечення їх передбачуваності та керованості. Динамічна генерація елементів інтерфейсу під конкретне завдання загрожує порушенням принципу консистентності – одного з фундаментальних принципів проєктування інтерфейсів: нові елементи можуть з'являтися в непередбачуваних місцях і не узгоджуватися з рештою системи, що призводить до когнітивного перевантаження та дезорієнтації користувача.

У роботі [4, с. 133] наводяться нові емпіричні дані, що засвідчують: проблеми доступності у згенерованих інтерфейсах залишаються актуальними попри стрімкий розвиток ШІ та посилення нормативних вимог. На думку авторів, це зумовлює необхідність зосередитися на розробленні стратегій, які змінять саме розуміння доступності системами штучного інтелекту – від формального дотримання технічних чеклістів до орієнтації на реальні потреби користувачів та контексти їхньої взаємодії з продуктом.

Постановка завдання. Метою статті є представлення підходу до автоматичної генерації шаблонів документів на основі зображень існуючих форм з використанням великих мовних моделей. Запропоноване рішення призначене для інтеграції в робочі процеси розробників програмного

забезпечення, а не для використання як окремий користувацький продукт.

Практичним результатом роботи системи є два взаємодоповнювальні артефакти. Перший – це електронна форма введення даних, яка забезпечує зручний структурований інтерфейс із підказками та автоматичною валідацією, що унеможливлює формальні помилки ще на етапі заповнення. Другий артефакт – це шаблонізований документ, що відповідає за перетворення зібраних даних на читабельний вихідний файл у форматі PDF, придатний для друку чи архівування. Розмежування між логікою даних і конкретним засобом відображення досягається через платформонезалежне JSON-представлення, що дозволяє інтегрувати систему в мобільні застосунки, десктопні програми або чат-боти без переробки базової логіки.

Виклад основного матеріалу. Вимоги до формату генерованих шаблонів документів включають:

– безпеку даних – відсутність виконуваного коду в шаблоні з метою запобігання потенційним вразливостям (ін'єкції коду тощо);

– легкість парсингу та валідації – можливість швидкого розбору шаблону за допомогою стандартних бібліотек (JSON, YAML), а також можливість декларативного опису формату для валідації шаблонів;

– зручність для генерації за допомогою LLM.

Концептуально шаблон документа розділено на три основні складові, кожна з яких відповідає за окремий аспект представлення інформації. Перший елемент – конфігурація даних, що описує змінні та їхні типи. Ця частина визначає, які саме дані мають бути зібрані від користувача або надані системою для заповнення документа. Типізація змінних охоплює базові примітиви на кшталт тексту і чисел, складні структури як об'єкти і масиви, а також специфічні для документообігу типи – підписи, зображення, часові мітки. Кожна змінна описується не лише своїм типом, але й додатковими властивостями, такими як обов'язковість заповнення, значення за замовчуванням, та обмеження на мінімальну і максимальну кількість елементів для колекцій.

Друга складова шаблону – структура електронної форми, що складається з компонентів. Компоненти визначають інтерфейс взаємодії користувача з системою при заповненні даних. До них належать текстові поля введення, числові поля, перемикачі, випадаючі списки, компоненти для підпису документів і багато інших. Кожен компонент пов'язаний з однією або декількома змінними з конфігурації даних і визначає спосіб візуалізації та валідації введеної інформації. Така архітектура забезпечує чітке розмежування між структурою даних і способом їх збору, що дозволяє використовувати одні й ті самі змінні в різних контекстах або, навпаки, застосовувати різні компоненти для збору однакових за типом даних залежно від бізнес-логіки.

Третя складова – це опис структури фінального документа через систему віджетів. Віджети відповідають за представлення даних у згенерованому PDF документі або іншому форматі виводу. Вони визначають розташування текстових блоків, таблиць, зображень, підписів та інших елементів на сторінках документа. Важливою особливістю є підтримка складних ієрархічних структур через контейнери, що можуть містити інші віджети, та ітератори, що дозволяють генерувати повторювані блоки на основі масивів даних. Така гнучкість дозволяє створювати складні документи з адаптивною структурою без необхідності генерувати множину окремих шаблонів для різних сценаріїв використання.

Мінімальний приклад шаблону у форматі JSON наведено на рис. 1.

```
{
  "variables": {
    "applicant_full_name": {
      "type": "TEXT",
      "required": true,
      "defaultValue": ""
    }
  },
  "components": [
    {
      "type": "HEADING",
      "level": 2,
      "text": "Дані заявника"
    },
    {
      "type": "INPUT",
      "label": "ПІБ",
      "placeholder": "Ім'я ПРІЗВИЩЕ",
      "variable": "applicant_full_name"
    }
  ],
  "widgets": [
    {
      "type": "PARAGRAPH",
      "widgets": [
        {
          "type": "STATIC_TEXT",
          "text": "Я, "
        },
        {
          "type": "TEXT_VALUE",
          "variable": "applicant_full_name"
        },
        {
          "type": "STATIC_TEXT",
          "text": ", прошу [...]"
        }
      ]
    }
  ]
}
```

Рис. 1.Мінімальний приклад шаблону у форматі JSON

На рисунку 2 представлено основні компоненти системи та потоки даних між ними.

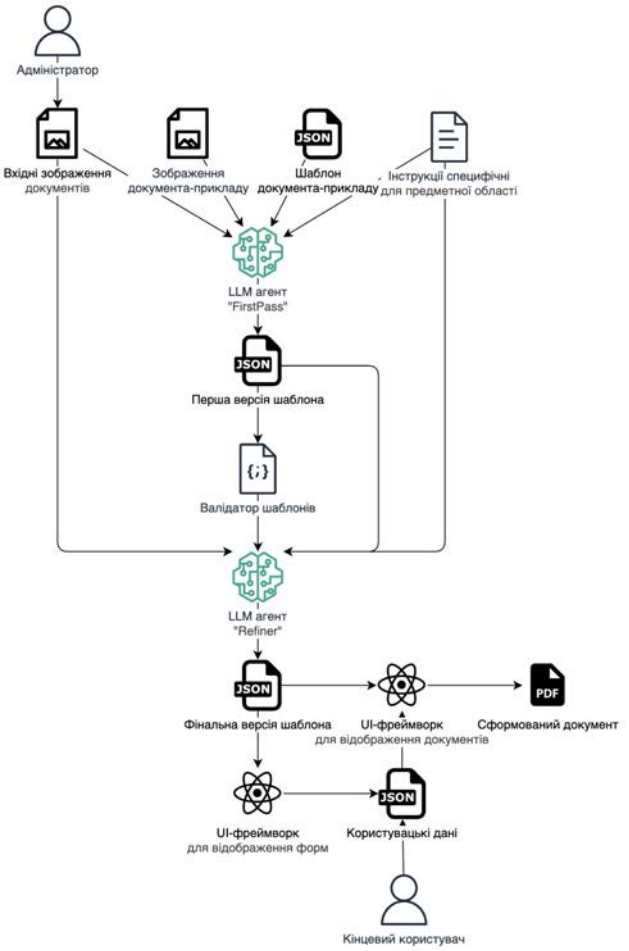


Рис. 2. Функціональна схема системи генерації шаблонів документів

Ключовими елементами на рисунку 2 є:

1. Вхідні зображення документа – скан-копії або фото реальних форм, які потрібно оцифрувати і перетворити на шаблон.

2. Зображення документа-прикладу та його JSON-шаблон демонструють моделі типову структуру даних у певній предметній області. Такий приклад визначається розробником системи і є відмінним від вхідного документа з п.1.

3. Інструкції, специфічні для предметної області, описують правила, яких мають дотримуватись всі згенеровані шаблони, у довільній формі. Наприклад: «All generated text (labels, placeholders etc.) must be in Ukrainian».

4. FirstPass – це LLM-агент, який на основі зображень та інструкцій будує першу чернетку шаблону у форматі JSON.

5. Валідатор – це модуль, який автоматично перевіряє згенерований шаблон на структурну коректність, повноту полів і відповідність схемі системи.

6. Refiner – це другий LLM-агент, який, використовуючи результати валідації, покращує й виправляє початковий шаблон, формуючи його фінальну версію.

Оцінка коректності шаблону фокусується на аналізі типів використаних елементів незалежно від їх порядку в документі. Структурна оцінка перевіряє, чи правильно мовна модель ідентифікувала типи компонентів форми (INPUT, TEXTAREA, YES_NO_CHECKBOX тощо), типи змінних (TEXT, NUMERIC, BOOLEAN тощо) та типи віджетів документа (CONTAINER, TABLE, LABELED_ITEM тощо).

Для оцінки якості розпізнавання типів використовується індекс Жаккара [5, с. 74]:

$$TP_c = min\left(count_{pred}(c), count_{true}(c)\right)$$

$$J_c = \frac{TP_c}{count_{true}(c) + count_{pred}(c) - TP_c}$$

де $count_{pred}(c)$ – кількість елементів класу c в згенерованому шаблоні,

$count_{true}(c)$– кількість елементів класу c в еталонному шаблоні.

Дана метрика спочатку обчислюється на рівні окремих класів, після чого агрегується у середнє зважене (weighted average) значення, що дозволяє отримати загальну оцінку якості окремого компонента шаблону.

$$J_{weighted} = \frac{\sum_{c \in C} J_c \cdot count_{true}(c)}{\sum_{c \in C} count_{true}(c)}$$

Для отримання єдиної оцінки для всього шаблону використовується гармонійне середнє:

$$J_{template} = \frac{3}{\frac{1}{J_{variables}} + \frac{1}{J_{components}} + \frac{1}{J_{widgets}}}$$

Систему було протестовано на документах з трьох предметних областей: україномовні юридичні документи (договори, заяви), україномовні військові документи (рапорти та заяви) та англомовні документи з охорони праці на будівництві (pre-task plan). Для кожної предметної області було відібрано по три приклади документів, специфічних для конкретної галузі. Для кожного документа було розроблено еталонний шаблон, з яким в подальшому порівнювались результати генерації моделей.

Тестування виконувалось для новітніх пропрієтарних великих мовних моделей (Claude Sonnet 4, Gemini 2.5 Flash, Gemini 2.5 Pro, GPT-5 High) та open-weight моделей (Llama 4 Maverick та Qwen3 VL 235B A22B Thinking).

Порівняння результатів тестування за індексом Жаккара (рис. 3) показало суттєву перевагу про-

прієтарних LLM над open-weight моделями у відтворенні структури шаблонів. Значення метрики для пропрієтарних моделей наближаються до 1, що вказує на точніше розпізнавання полів та вищу узгодженість конфігурації змінних з еталоном. Найстабільніші показники в усіх трьох предметних областях продемонстрували Gemini 2.5 Flash, Gemini 2.5 Pro та GPT-5, що робить їх пріоритетними кандидатами для впровадження у розроблену систему.

Окрім метрик, заснованих на порівнянні з еталонним шаблоном, додатково аналізувалася кількість помилок валідації на кожному етапі генерації. На рисунку 4 ці дані подано у форматі «кількість помилок FirstPass → кількість помилок Refiner». На рисунку видно, що в 33 із 45 експериментів початковий шаблон порушує декларативні правила валідації, закладені в системі, і не може бути безпосередньо використаний для відображення форми або генерації документа. Водночас у переважній більшості випадків Refiner зменшує кількість помилок або залишає її незмінною, скорочуючи число невалідних шаблонів до 19.

Розглянемо приклад роботи системи з використанням моделі Google Gemini 2.5 Pro. В якості вхідних даних використовувався зразок рапорту про недоотримане майно, отриманий з відкритих джерел [9]. Вхідними даними виступає скріншот незаповненого зразка в текстовому редакторі (рис. 5).

На рисунку 6 зображено фрагмент електронної форми, згенерованої системою. На даному прикладі видно, що:

– модель виділила логічні секції («Дані отримувача», «Деталі рапорту», «Дані заявника») та створила для них відповідні заголовки;

– змінні реквізити були коректно представлені полями типу INPUT (текстові поля для посади, військової частини, звання, ПІБ тощо) та NUMERIC_INPUT (поля з числовими значеннями, зокрема

Рис. 3. Гармонійне середнє значення коефіцієнтів Жаккара для набору тестових документів

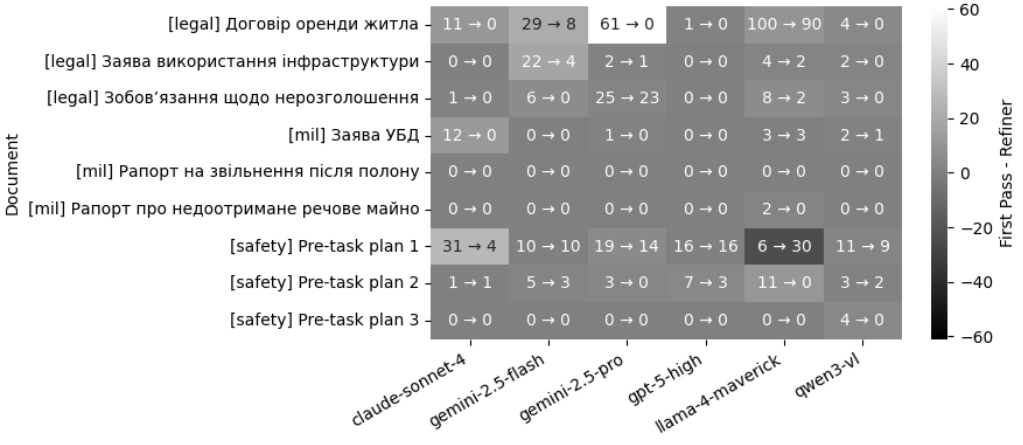
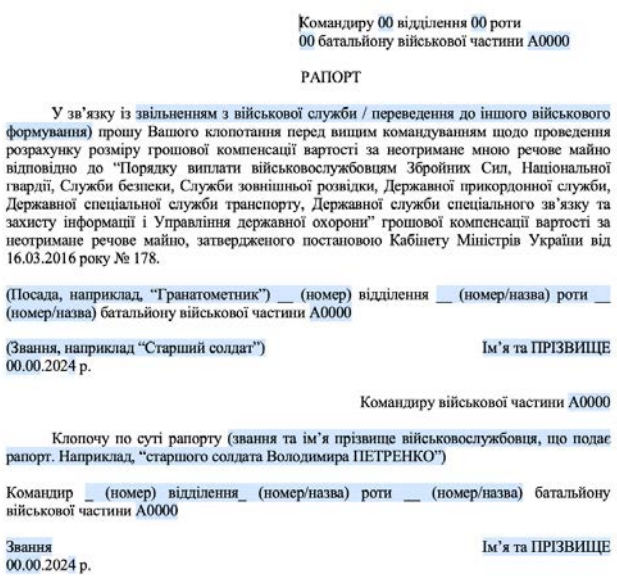


Рис. 4. Кількість помилок валідації для FirstPass та Refiner



Командиру 00 відділення 00 роти
00 батальйону військової частини А0000

РАПОРТ

У зв'язку із звільненням з військової служби / переведення до іншого військового формування) прошу Вашого клопотання перед вищим командуванням щодо проведення розрахунку розміру грошової компенсації вартості за неотримане мною речове майно відповідно до "Порядку виплати військовослужбовцям Збройних Сил, Національної гвардії, Служби безпеки, Служби зовнішньої розвідки, Державної прикордонної служби, Державної спеціальної служби транспорту, Державної служби спеціального зв'язку та захисту інформації і Управління державної охорони" грошової компенсації вартості за неотримане речове майно, затвердженого постановою Кабінету Міністрів України від 16.03.2016 року № 178.

(Посада, наприклад, "Гранатометник") __ (номер) відділення __ (номер/назва) роти __ (номер/назва) батальйону військової частини А0000

(Звання, наприклад "Старший солдат") Ім'я та ПРІЗВИЩЕ
00.00.2024 р.

Командиру військової частини А0000

Клопочу по суті рапорту (звання та ім'я прізвище військовослужбовця, що подає рапорт. Наприклад, "старшого солдата Володимира ПЕТРЕНКО")

Командир _ (номер) відділення_ (номер/назва) роти __ (номер/назва) батальйону військової частини А0000

Звання Ім'я та ПРІЗВИЩЕ
00.00.2024 р.

Рис. 5. Вхідне зображення документа

номери відділення, роти та батальйону);

– для окремих полів, що мають обмежену кількість можливих варіантів, наприклад «Причина» подання рапорту, було згенеровано поле типу DROPDOWN, що дозволяє обрати значення зі списку.

Отже, система коректно інтерпретувала структуру вхідного документа і перетворила його на валідний опис електронної форми у форматі JSON.

Другим артефактом роботи системи є шаблонізований текст рапорту, який так само зберігається у форматі JSON і описується через систему віджетів (параграфи, статичний текст, змінні, розриви рядків тощо). Через значний обсяг та вкладеність цієї структури відобразити її в межах даного прикладу складно, тому на рисунку 7 наведено готове представлення згенерованих віджетів засобами UI-фреймворку. Для наочного співставлення полів форми з відповідними місцями в тексті документа використовується синтаксис {{назва_змінної}}, наприклад {{applicant_full_name}}.

Загалом система коректно відтворила структуру вхідного документа: дані, які вводить користувач, оформлено як окремі поля форми, тоді як незмінний текст рапорту перенесено в шаблон без суттєвих спотворень.

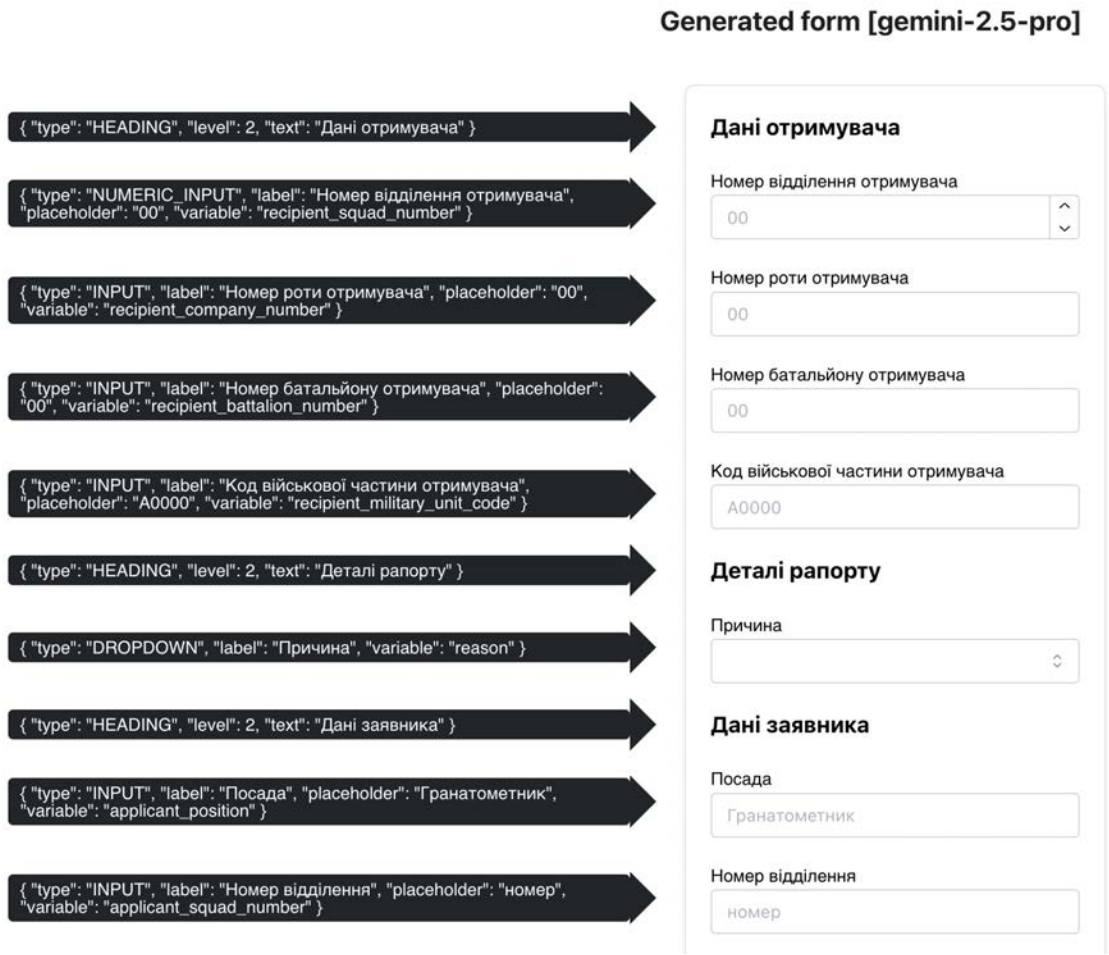


Рис. 6. Фрагмент згенерованої форми

Generated document

Командиру {{recipient_squad_number}} відділення {{recipient_company_number}} роти
{{recipient_battalion_number}} батальйону військової частини {{recipient_military_unit_code}}

РАПОРТ

У зв'язку із {{reason}} прошу Вашого клопотання перед вищим командуванням щодо проведення розрахунку розміру грошової компенсації вартості за неотримане мною речове майно відповідно до "Порядку виплати військовослужбовцям Збройних Сил, Національної гвардії, Служби безпеки, Служби зовнішньої розвідки, Державної прикордонної служби, Державної спеціальної служби транспорту, Державної служби спеціального зв'язку та захисту інформації і Управління державної охорони" грошової компенсації вартості за неотримане речове майно, затвердженого постановою Кабінету Міністрів України від 16.03.2016 року № 178.

{{applicant_position}} {{applicant_full_name}}
{{applicant_squad_number}} відділення
{{applicant_company_number}} роти
{{applicant_battalion_number}}
батальйону військової частини
{{applicant_military_unit_code}}
{{applicant_rank}}
{{report_date}} р.

Командиру військової частини {{approving_commander_unit_code}}

Клопочу по суті рапорту {{applicant_rank_and_name_genitive}}.

Командир {{applicant_squad_number}} {{approver_full_name}}
відділення
{{applicant_company_number}} роти
{{applicant_battalion_number}}
батальйону військової частини
{{applicant_military_unit_code}}
{{approver_rank}}
{{approval_date}} р.

Рис. 7. Шаблонізований документ

Висновки. Проведене дослідження підтверджує доцільність застосування великих мовних моделей для автоматизації процесу генерації шаблонів документів. Аналіз експериментальних результатів показав, що на сьогодні можливості open-weight моделей є недостатніми для рішення даного класу задач та суттєво поступаються можливостям пропрієтарних моделей, серед яких Gemini 2.5 Pro продемонструвала найбільш збалансовані показники якості генерованих шаблонів документів.

Застосований двоетапний підхід з агентами «FirstPass» та «Refiner» показав свою ефективність, підтвердивши здатність мовних моделей до самокорекції при наданні детальної інформації про виявлені помилки, хоча у деяких випадках навіть повторний запит не призводить до повного виправлення помилок, що вказує на необхідність розробки більш складних механізмів контролю якості.

Запропонований підхід може бути використаний як основа для розроблення гнучких систем генерації форм і документів у різних прикладних сферах. Для його успішної реалізації головне значення мають: наявність чітко визначених метрик оцінки якості результату для вибору ефективних LLM агентів, використання декларативних правил валідації на основі схем, а також надання моделі прикладів у форматі few-shot або one-shot learning безпосередньо в контекстному вікні, що дозволяє сформувати адекватне уявлення про очікуваний формат виводу.

Подальший напрямок досліджень може бути спрямований на експерименти з донавчанням open-weight моделей.

Список літератури:

1. Achachlouei M.A., Patil O., Joshi T., Nair V.N. Document Automation Architectures: Updated Survey in Light of Large Language Models. arXiv preprint. Ithaca, NY, 2023. 15 p. DOI: https://doi.org/10.48550/arXiv.2308.09341

2. Baviskar D., Ahirrao S., Potdar V., Kotecha K. Efficient Automated Processing of the Unstructured Documents Using Artificial Intelligence: A Systematic Literature Review and Future Directions. *IEEE Access. 2021*, Vol. 9. P. 72894–72936. DOI: https://doi.org/10.1109/ACCESS.2021.3072900

3. Could AI convert all the PDF forms on GOV.UK into web forms? URL: https://www.timpaul.co.uk/posts/could-ai-convert-all-the-pdf-forms-on-gov-uk-into-web-forms/ (дата звернення: 20.03.2026).

4. Gurita A.-E., Vatavu R.-D. When LLM-Generated Code Perpetuates User Interface Accessibility Barriers, How Can We Break the Cycle? *W4A '25: Proceedings of the 22nd International Web for All Conference* (Sydney, NSW, Australia, 2025). New York, NY, USA, 2025. P. 124–134. DOI: 10.1145/3744257.3744266.

5. Leskovec J., Rajaraman A., Ullman J. D. Mining of Massive Datasets. 2nd ed. Cambridge, 2014. 476 p.

6. Okopnyi P., Nordberg O. E., Guribye F. Against Generative UI. *Proceedings of the Halfway to the Future Symposium* (Santa Cruz, CA, USA, 2024). New York, NY, USA, 2024. P. 1–4. DOI: 10.1145/3686169.3686184.

7. Using AI to generate web forms from PDFs. URL: https://www.timpaul.co.uk/posts/using-ai-to-generate-web-forms-from-pdfs/ (дата звернення: 20.03.2026).

8. Walther B., Hossin S., Townend J., Abernethy N. et al. Comparison of Electronic Data Capture (EDC) with the Standard Data Capture Method for Clinical Trial Data. PLOS ONE. 2011. Vol. 6, No. 9. e25348. DOI: 10.1371/journal.pone.0025348.

9. Зразки рапортів - Турбота про військовослужбовця. URL: https://turbota.mil.gov.ua/zrazky-documentiv (дата звернення: 20.03.2026).

10. Рапорти на відпустку, виплати, лікування, звільнення | Армія+ : веб-сайт. URL: https://aplus.mod.gov.ua/raporty (дата звернення: 20.03.2026).

Nikitin V.S., Hiorhizova-Hai V.S. AN APPROACH TO DOCUMENT TEMPLATE GENERATION USING LLM AGENTS

The article proposes an approach to the automatic generation of document templates from images of paper or electronic forms using large language models and agent-based processing. The relevance of the study is driven by the need to standardize document workflows, the growing volume of unstructured data, and the need to reduce the time required for the manual design of electronic forms and templated documents. The proposed approach enables the transformation of a document image into a structured JSON template suitable for subsequent rendering as an interactive form and a templated document. The main requirements for the template format are identified as data security, ease of parsing and validation, and suitability for generation by LLMs. The template is divided into three components: data configuration, electronic form structure, and final document structure.

The system architecture includes an input document image, an example document with a reference JSON template, domain-specific instructions, and two LLM agents – FirstPass and Refiner. The first agent generates the initial template, while the second refines it based on the results of automatic validation.

To evaluate the quality of the generated templates, comparison with reference templates and structural similarity metrics were used. The experimental study was conducted on documents from several domains, including legal, military, and occupational safety documents. The results of the system using a number of modern proprietary and open-weight models, as well as the results produced by the two LLM agents, were compared. The findings indicate that applying a two-stage generation scheme followed by validation reduces the number of structural errors in the final result. Implementing the proposed approach can accelerate document digitalization, unify form structures, reduce the workload on developers, and provide a foundation for further automation in the development of electronic document workflow services. The proposed approach can serve as a basis for developing flexible systems for generating forms and documents in various application domains.

Keywords: *large language models, electronic forms, document templates, document workflow automation, generative user interfaces, intelligent distributed computing.*

Дата першого надходження статті до видання: 25.03.2026
Дата прийняття статті до друку після рецензування: 22.04.2026
Дата публікації (оприлюднення) статті: 19.05.2026