ISSN 2227-6890

ІНТЕГРАЦІЯ ФІЛОЛОГІЇ ТА ТЕХНІЧНИХ НАУК

УДК 004.91 **doi: 10.20998/2227-6890.2025.1.11**

Н.В. БАБКОВА, Д.О. ГУЛІЄВА, З.А. КОЧУЄВА, Н.С. УГОЛЬНІКОВА

СЕГМЕНТАЦІЯ ТЕКСТУ В БАГАТОСТОРОННІХ ДІАЛОГАХ

У статті йдеться про повноцінний відкритий інструмент, що дозволяє виконувати сегментацію діалогів кількох учасників з використанням мінімального обсягу навчальних текстів або без його використання; доведено, що основною проблемою існуючих рішень є неповноцінність їх інструментів, які виступають лише демонстрацією можливих способів решения; запропоновано гібридну модель сегментації, що базується на використанні ембедінгів BERT та деяких лінгвістичних ознак, унікальних для діалогів кількох учасників; представлено хороші результати моделі, які можна порівняти з провідними результатами, описаними в роботах за цією тематикою; підсумком роботи стало створення повноцінного інструменту, що дозволяє автоматично відтворювати сегментацію.

Ключові слова: сегментація тексту, BERT-ембедінги, обробка природної мови, лінгвістичні ознаки, автоматичне розпізнавання мовлення.

N.V. BABKOVA, D.O. HULIIEVA, Z.A. KOCHUIEVA, N.S. UGOLNIKOVA

TEXT SEGMENTATION IN MULTI-PARTY DIALOGUES

The article discusses a full-fledged open tool that allows segmentation of multi-participant dialogues with minimal or no training texts; It is proven that the main problem of existing solutions is the inferiority of their tools, which serve only as a demonstration of possible solutions; a hybrid segmentation model is proposed, based on the use of BERT embeddings and some linguistic features unique to multi-participant dialogues; good model results are presented, which can be compared with the leading results described in works on this topic; the result of the work was the creation of a full-fledged tool that allows automatic segmentation reproduction.

Key words: text segmentation, BERT embeddings, natural language processing, linguistic features, automatic speech recognition.

Постановка проблеми. У загальному випадку сегментація тексту при обробці природної мови – це завдання, що полягає у розбитті тексту на значні оформлені відрізки, які відповідають певній темі чи підтемі. Незважаючи на простоту цього завдання для людини, автоматичне вирішення цього завдання представляє досить велику складність. У той час, як людина може легко розмітити текст на сегменти, використовуючи знання мови та правила побудови текстів, для алгоритмів ці знання в загальному випадку недоступні. Для алгоритму весь текст є послідовність деяких токенів, які спочатку ніяк не пов'язані один з одним.

У деяких випадках розділити текст на сегменти буває важко навіть людині. Особливо це актуально в тих випадках, коли текст є неформальним діалогом або обговоренням, в яких одна тема може перетікати в іншу настільки плавно, що неможливо визначити, яке саме місце в діалозі слід вважати зміною теми. Додатковою складністю є те, що в таких діалогах учасники можуть повертатися до обговорення теми, закритої раніше, тобто діалог не має чіткої лінійної структури.

Сегментація тексту може застосовуватися для того, щоб зробити текст простішим для сприйняття або пошуку інформації, причому сегментований текст може згодом використовуватися як людиною, так і іншими алгоритмами при вирішенні інших завдань NLP. Наприклад, коректна сегментація діалогу покращує якість і швидкість виконання таких завдань, як вилучення фактів та автоматичне реферування (стислий переказ) розмови [1].

Сегментація діалогу кількох учасників – окремий випадок завдання сегментації тексту. Найчастішим прикладом такого діалогу є робочі дзвінки або інші онлайн-зустрічі з кількома учасниками, де немає яскраво вираженого оратора. У таких діалогах усі учасники рівноправні та черговість їхнього виступу не визначена. Останнім часом кількість таких діалогів збільшується й

актуальність завдання підвищується. Наприклад, багато колективів відчувають необхідність у обробці текстів робочих розмов телефоном. Однією з потреб може бути оформлення завдань, поставлених тим чи іншим співробітникам у процесі дзвінка письмово. У цьому випадку сегментація тексту може бути одним з етапів ланцюжка, що вирішує цю проблему. Логічно припустити, що модуль отримання фактів з тексту покаже набагато кращі результати на логічно розділеному на сегменти тексті. Було помічено, що пошук інформації за ключовими словами, для якого розподіл тексту на сегменти не є необхідним, є неефективним для випадку діалогів з кількома учасниками через велику кількість помилок при автоматичному розпізнаванні усного тексту.

У той самий час завдання сегментації діалогів кількох учасників становить додаткову складність. По-перше, дані діалоги більш «зачумлені». Через те, що яскраво вираженого оратора чи модератора немає, репліки учасників можуть перекривати одне одного, навіть ненавмисно. Через це погіршується якість розпізнавання мови та ускладнюється процес сегментації. По-друге, даних, у яких можна було б навчити алгоритм сегментації чи навіть оцінити його ефективність не так багато. Як було згадано вище, розмітка подібних діалогів становить складність навіть для людини, що робить процес анотації дорогим та трудомістким. Ситуація посилюється тим, що тексти діалогів, як правило, мають конфіденційний характер і не виявляються у вільному доступі.

Отже, завдання сегментації можна формалізувати в такий спосіб. На вході є послідовність висловлювань у текстовому вигляді. Для кожного висловлювання існують дві тимчасові мітки, що вказують у який час висловлювання почалося, і у який час воно закінчилося. Також кожне висловлювання пов'язане з конкретним учасником (учасники можуть позначатись будь-якими мітками). Завдання полягає в тому, щоб кожному висловлюванню зіставити мітку 0 або 1, де мітка 0 означає, що це висловлювання не є початком нового сегмента, а мітка 1 означає, що з цього висловлювання починається новий сегмент.

Аналіз останніх досліджень і публікацій. З часом для розв'язання завдання сегментації було запропоновано безліч підходів, що ґрунтувалися на різних аспектах розмовної мови. Зокрема, деякі роботи пропонують одночасно визначати межі та типи сегментів як єдине комплексне завдання [2]. Нижче розглянуто основні групи підходів.

Лексичні змінення. Цей підхід грунтується на тому, що зміни теми, як обговорюється, супроводжуються зміною лексики, що використовується в діалозі. Робиться спроба виявити ділянки у діалозі, у яких лексична зв'язність тексту досягає мінімального значення. Ці дільнки є головними кандидатами на зміну теми. Однією з ранніх моделей, в якій застосовується даний підхід, є модель TextTiling [3], результати якої досі часто використовуються як основний. У цій моделі діалог токенізується, стеммінгується та поділяється на вікна фіксованої довжини. Кожне «вікно» представляється як вектор, у якому значеннями є частоти кожного токена. Далі обчислюються косинусні відстані між векторами «вікон» та вектора, між якими ці відстані досягають максимуму, є гіпотетичними межами тем. Для свого часу цей підхід був досить ефективним, але з часом, очевидно, з'явилися більш нові методи.

У 2003 році було запропоновано модель під назвою LCSeg, яка суттєво вплинула на подальші дослідження в цій галузі. В основі моделі лежить концепція лексичних ланцюжків – повторюваних слововжитків однієї й тієї ж лексеми. За задумом, більшість цих ланцюжків буде розриватися при зміні теми. При цьому найбільшу вагу отримують ланцюжки найчастіших лексем, а також довгі ланцюжки. Зрештою метрики розраховуються так само, як і в попередньому методі – шляхом обчислення косинусних відстаней між ланцюжками у сусідніх вікнах. Особливість цього в тому, що він показав гарні результати саме для діалогів кількох учасників.

Після поширення мовних моделей, зокрема BERT, стало можливим їх застосування для завдання сегментації тексту [4]. Останні дослідження в даній галузі використовують векторні уявлення слів та речень для обчислення подібності між дільницями діалогу.

Кластерізація за схожістю. Цей підхід є повною протилежністю попереднього. Замість того, щоб шукати ділянки з мінімальними зв'язками, можна навпаки шукати сегменти, всередині яких зв'язність досягає максимального значення. Після кластеризації речень тексту за таким принципом виходить готова відповідь на завдання. Кластеризація може бути агломеративною – у цьому випадку початковими центрами кластерів оголошуються пари висловлювань з максимальною подібністю, потім кластери розширюються. Однак для цього завдання більш ефективною вважається дивізивна кластеризація – спочатку весь діалог приймається за один кластер, після чого він починає дробитися на менші частини.

Прикладом застосування такого підходу виступає метод «побудови за точками», де весь діалог подається у вигляді двовимірної матриці, в якій рядки та стовпці є словами, розташованими за порядком. Матриця заповнюється одиницями (точками) на перетині однакових слів. Очевидно, що головна діагональ матриці гарантовано буде заповнена одиницями, однак, одиниці також зустрінуться і в інших місцях. Найбільш оптимальною кластеризацією вважається таке розбиття, при якому щільність точок всередині кластерів максимальна, і навпаки, щільність точок поза кластерами мінімальна. Тобто, таким чином виділяються теми, які максимально зв'язуються всередині діалогу і при цьому максимально відрізняються від суміжних тем. На основі цього методу було запропоновано й інші методи з тією самою стратегією. Наприклад, один із таких методів використовує речення замість слів.

Пошук відмінних ознак. Цей підхід кардинально відрізняється від двох попередніх. У даному випадку робиться спроба виділити деякі ознаки у діалозі, які можуть означати наявність кордону між темами. Одним із прикладів таких ознак є конкретні слова або фрази, якими говорячи зазвичай супроводжують зміни теми. Для української це, наприклад, такі фрази як «отже», «перейдемо до», тощо. Для англійської мови у цьому контексті часто згадуються слова «so» та «anyway» [5]. Також про зміну теми може сигналізувати тривала пауза в розмові або зміна того, хто говорить (хоча ці ознаки не гарантують зміну теми) [6].

Мета статті – створення повноцінного відкритого інструменту, що дозволяє виконувати сегментацію діалогів.

Виклад основного матеріалу. Основною моделлю сегментації текстів без вчителя вважається алгоритм, заснований на застосуванні ембедінгів BERT. В її основі лежить підхід знаходження в діалозі ділянок, у яких є мінімальна лексична зв'язність. Здатність BERT ефективно створювати контекстуальні семантичні уявлення слів та цілих речень забезпечує даній моделі хороші результати [7]. Немає досить великого обсягу даних на навчання моделей, оскільки більшість текстів діалогів залишається у закритому доступі. Тому на перший план виходить саме навчання без вчителя, а існуючий невеликий обсяг даних може використовуватися як тестова вибірка для оцінки ефективності різних параметрів самої моделі. Відомі спроби навчання моделей з учителем на текстових даних іншого домену з вільного доступу, проте такі моделі програють за рахунок того, що наявні у вільному доступі дані дуже відрізняються від реальних діалогів.

Запропонована модель складається з двох основних компонентів. Перший компонент відповідає за обчислення векторного подання речень та визначення їхньої семантичної близькості. Другий компонент безпосередньо визначає межі тем, використовуючи дані про зміну семантичної близькості між реченнями за часом. В якості моделі, що обчислює векторні уявлення слів, автори використовують дві: RoBERTa та Sentence-BERT. За допомогою RoBERTa вектори обчислюються шляхом вибору максимальних значень елементів передостаннього шару. Це обгрунтовується тим, що максимальний обсяг семантичної інформації зберігається саме у останніх шарах моделі. Вибір максимальних значень автори пояснюють тим, що таким чином відсікаються токени, що не несуть значущої інформації (стоп-слова й фрази) [8]. При використанні Sentence-BERT вектори обчислюються шляхом усереднення всіх результуючих векторів.

В основі сегментації лежить та сама ідея, що й у згаданій вище моделі TextTiling. Але у випадку з BERT семантична схожість обчислюється на основі ембедінгів таким чином:

1) для кожного речення у діалозі обчислюється його векторне уявлення;

2) весь текст діалогу ділиться на блоки (оптимальний розмір блоку було обчислено у процесі оцінки як гіперпараметр), для кожного блоку обчислюється один вектор шляхом вибору максимальних компонентів серед відповідних компонентів векторів пропозицій;

3) обчислюється косинусна відстань між блоками;

4) межі тем визначаються як ділянки, у яких косинусна відстань між блоками перевищує певний заданий поріг.

Поріг є настроюваним значенням і відсутність навчальної вибірки визначається емпіричним шляхом. Чим вище цей поріг, тим грубішим буде розбиття на теми, і навпаки, чим менше значення порога, тим більше тематик виділятиме модель. Наприклад, якщо нормалізувати всі косинусні відстані так, щоб вони набували значення від 0 до 1, то при значенні порога 0 кожен блок вважатиметься окремою темою, а при значенні порога 1 весь діалог вважатиметься однією великою темою.

При ретельному тестуванні цієї моделі та порівнянні готової розмітки зі зразком було виявлено, що модель, яка використовує лише ембедінги BERT, неспроможна розпізнати ознаки, що є очевидними (для людини) сигналами зміни теми. Одним із таких прикладів є, наприклад, привітання, коли до діалогу вступає новий учасник. BERT-модель не звертає на них уваги (це знаходиться за межами її призначення). Також було виявлено цікаві приклади помилкової розмітки, коли у промові одного з учасників з'являється фразеологічний оборот. Як правило, у фразеологічних оборотах використовується лексика, що ніяк не перетинається з лексикою загальної теми дискусії, через це модель вирішує, що вектор обговорення значно змінився і ставить на цій ділянці межу теми. Усі ці недоліки не означають, що модель не працює. Вона показує хороші результати, що перевершують багато інших аналогів, проте її можна поліпшити, якщо використовувати гібридний підхід і доповнити її аналізом лінгвістичних ознак.

У 2003 році було запропоновано модель, що поєднує дискурсивний аналіз та аналіз ознак. Високі результати досягалися за рахунок того, що рішення про сегментацію приймали на основі двох показників, що знижувало ймовірність помилок. Об'єднання показань цих показників проводилося за рахунок імовірнісного класифікатора.

Модуль аналізу дискурсу так само, як і модель на основі BERT, працювала за принципом знаходження ділянок діалогу з найбільшою лексичною зв'язністю. Перед початком аналізу виконується стандартний препроцесинг тексту: токенізація, видалення стоп-слів, стемінг. Застосування стеммінгу в моделі є важливим кроком, оскільки на той момент потужних мовних моделей ще не існувало, і стеммінг допомагав провести паралелі між різними дільницями діалогу, в яких фігурували різні форми однієї й тієї ж лексеми.

Модуль дискурсивного аналізу оперує поняттям лексичного ланцюжка. Під лексичним ланцюжком розуміються повтори лем серед різних ділянок діалогу. При цьому модуль логічно поділяється на дві частини: знаходження лексичних ланцюжків і власне визначення потенційних меж тем на основі даних лексичних ланцюжків. Спочатку для кожної леми, що повторюється, виводиться по одному лексичному ланцюжку: від першого і до останнього слововживання. Потім ці ланцюжки дробляться у тих ділянках діалогу, у яких виявляються слововживання даної леми. Таким чином відбувається відсіювання слабко зв'язаних ланцюжків. Після побудови ланцюжків кожному з них присвоюється вага значущості: найбільшу вагу отримують ланцюжки, у яких лема повторюється частіше, і навіть компактніші (короткі) ланцюжки (з гіпотези, що короткі ланцюжки є сильною ознакою лексичної зв'язності).

Після обчислення ланцюжків застосовується той самий підхід, що і в оригінальному алгоритмі TextTiling, тобто деяке вікно переміщається по тексту діалогу і обчислює косинусні відстані між вагами ланцюжків, що покривають розриви між вікнами. Розриви, в яких косинусна відстань є максимальною, є потенційними межами сегментів.

Модуль аналізу ознак приймав рішення про наявність кордонів з урахуванням форми діалогу, а не його змісту. Виділялося кілька ознак, з урахуванням яких будувалися правила визначення кордонів із певною ймовірністю. У моделі використовуються наступні ознаки:

1. Сполучні фрази. У ранніх роботах з сегментації часто виявлялася кореляція між межами сегментів і особливими сполучними словами чи фразами, що передують початку нової теми. Для англійської такими словами є, наприклад, «now» і «well», в українській мові схожу роль можуть виконувати слова «отже», «далі», тощо. І хоча обсяг датасетів розмічених діалогів невеликий, його все ж таки достатньо для того, щоб витягти деякі такі фрази і скласти словник.

2. Паузи. Існує зв'язок між переходами до нової теми та суттєвими паузами в діалозі (набагато вищими, ніж у звичайній промові між репліками). Однак, не будь-яка пауза свідчить про наявність нової теми. Паузи, що розділяють репліки одного й того ж спікера, як правило, не пов'язані з переходом на нову тему, а скоріше, з деяким роздумом. Також слід особливо виділяти паузи після того, як один з тих, хто говорить, ставить питання (конкретному мовцю або всієї аудиторії). Паузи, пов'язані з переходом на нову тему, які не належать до жодного із спікерів, і тривають доти, доки один із учасників не перехопить ініціативу в діалозі. Таким чином, з формальної точки зору для визначення меж сегментів виділяються ділянки з підвищеною тривалістю пауз, причому по обидва боки від цих пауз повинні бути різні учасники, а перед паузою не повинно бути репліки запитання.

3. Перетин розмовляючих. Ця ознака схожа за своєю суттю на попередню, але працює у зворотний бік. Природно припустити, що у тих дільницях діалогу, де одночасно активні двоє чи більше учасників, ведеться якась жвава дискусія, що з меншою ймовірністю буде перервана зміною теми. Таким чином, ця ознака певною мірою послаблює інші: тобто наявність перетинів тих, хто говорить, знижує загальну ймовірність переходу, навіть якщо інші ознаки вказують на зміну теми.

4. Зміна того, хто говорить. Зміна теми часто супроводжується зміною спікера, проте не будь-яка зміна спікера вказує на зміну теми. Найбільший інтерес мають випадки, коли поточний учасник діалогу тривалий час утримував ініціативу у розмові, після чого слово перейшло до іншого учасника. Зворотні ситуації, коли розмовляючи часто змінюються, швидше свідчать про активне обговорення однієї теми, ніж про перехід від однієї до іншої.

У цілому, дана модель за оцінками авторів показує хороший результат, проте, по-перше, неможливо відтворити цей результат, оскільки немає коду чи готового інструменту, який реалізує даний метод, а по-друге, цей метод набагато краще справляється з сегментацією коли число сегментів відомо заздалегідь, а на практиці такі умови практично не зустрічаються.

Традиційними оцінками ефективності сегментації є метрики Pk та WindowDiff. Еталонна та оцінювана сегментація виражаються у вигляді послідовності нулів та одиниць, де одиниця означає початок нової теми (сегменту). Обидві метрики обчислюються шляхом переміщення «вікна» за документом та порівняння еталонної сегментації з оцінюваною. Результатом порівняння є визначення певної ймовірності помилки сегментації. Реалізація обох метрик включена до стандартного пакета NLTK Python та не вимагає доопрацювання.

Метрика Pk при кожному переміщенні «вікна» визначає, чи потрапляють кінці «вікна» в сегментації, що оцінюється, в ті ж сегменти, що і в еталонній розмітці [4]. У разі розбіжності збільшується певний лічильник. Підсумкова оцінка масштабується до діапазону від 0 до 1 з урахуванням кількості вимірів. Модель, яка успішно передбачила всі межі, отримує оцінку 0 (відсутність помилок сегментації). Метрика Pk проста, але має низку недоліків. По-перше, вона «карає» хибнонегативні помилки сильніше, ніж хибнопозитивні. По-друге, вона не бере до уваги кількість меж усередині вікна (якщо їх кілька, метрика цього «не помітить»). І по-третє, незначні промахи караються дуже сильно. Тобто, навіть якщо оцінювана та еталонна межа знаходяться поруч один з одним, це вважається такою ж серйозною помилкою, як і будь-який інший промах.

Метрика WindowDiff покликана усунути проблеми з метрикою Pk. Для кожної позиції «вікна» порівнюється кількість меж в еталонній розмітці та оцінці, що оцінюється. У цьому випадку хибнопозитивні та хибнонегативні помилки

караються однаково. Насправді застосовуються обидві ці метрики.

У процесі застосування цих метрик для оцінки ефективності різних моделей було помічено досить неприємну особливість. Обидві метрики покращувалися в міру збільшення порога зміни теми, і найкращі результати мали місце у виродженому випадку, коли поріг дорівнював 1 (тобто весь текст сприймався як один великий сегмент). Звичайно, цю проблему можна вирішити, порівнюючи моделі при однаковому значенні порога [9]. Однак, хотілося б мати можливість також порівнювати різні пороги, щоб зрозуміти, скільки сегментів найбільше схоже на еталонне. Таким чином, існує необхідність пошуку альтернативних метрик, які будуть позбавлені цього недоліку.

Загалом, практична частина дослідження відбувалася за наступним сценарієм:

3. Вибір та підготовка датасетів. На цьому етапі було проаналізовано датасети, пов'язані з темою сегментації. Серед них було обрано два датасети – AMI та ICSI. Корпус AMI є датасетом, що містить 100 годин розмов кількох учасників. Приблизно дві третини цих діалогів було створено штучно: учасники грали різні ролі у вигаданому робочому колективі та обговорювали створення деякого проекту з нуля протягом дня.

Третина діалогів, що залишилася, – реальні робочі телефонні розмови з різноманітних тематик. Корпус містить різні типи розмітки, зокрема, для кожного висловлювання вказуються дві тимчасові мітки (початку та кінця висловлювання) та позначається учасник. Також для діалогів існує еталонне розбиття на сегменти, яке використовується для оцінки ефективності методів.

Корпус ICSI загалом не сильно відрізняється від корпусу AMI. Обсяг даних у ICSI трохи менше – 70 годин записаних діалогів, але ці діалоги були записані у природній обстановці і стосувалися реальних питань зі сфери лінгвістики і мовлення. Середня тривалість діалогу в корпусі – 60 хвилин, у діалогах беруть участь у середньому 6-7 осіб. Розмітка містить ті самі дані, що і корпус AMI.

Спочатку обидва ці датасети знаходяться не в зовсім придатному для обробки вигляді. Крім еталонної сегментації датасети містять також іншу розмітку, наприклад, діалогові акти і короткі змісти діалогів. Ця інформація не потрібна для роботи нашої моделі, тому вихідні датасети були піддані препроцессингу. Крім цього, вихідні корпуси були представлені у форматі JSON, а для роботи моделі зручніше було уявити датасети в CSV-форматі, який добре сумісний з бібліотекою pandas.

Наприклад, у вихідному вигляді корпус AMI виглядав так:

[{"id": "ES2002a.B.dialog-act.dharshi.1", ***"speaker": "B", "starttime": "50.42"****, "startwordid": "ES2002a.B.words0",* ***"endtime": "50.99"****, "endwordid": "ES2002a.B.words0",* ***"text": "Okay"****, "label": "stl", "attributes": {"role": "PM", "participant": "FEE005"}}, {"id": "ES2002a.B.dialog-act.dharshi.2",* ***"speaker": "B", "starttime": "53.56"****, "startwordid": "ES2002a.B.words2",* ***"endtime": "53.96"****, "endwordid": "ES2002a.B.words2",* ***"text": "Right"****, "label": "stl", "attributes": {"role": "PM", "participant": "FEE005"}}, {"id": "ES2002a.B.dialog-act.dharshi.3",* ***"speaker": "B", "starttime": "55.415"****, "startwordid": "ES2002a.B.words4",* ***"endtime": "60.35"****, "endwordid": "ES2002a.B.words16",* ***"text": "<vocalsound> Umwell this is the kick-off meeting for our our project ."****, "label": "inf", "attributes": {"reflexivity": "true", "role": "PM", "participant": "FEE005"}}…*

Жирним шрифтом виділено інформацію, необхідну для роботи моделі. Підкреслено безпосередньо текст діалогу.

Видно, що значна частина інформації для вирішення задачі сегментації не є цікавою. Після препроцессингу дані були представлені в набагато зручнішому вигляді:

id,text,label,starttime,endtime,speaker
*ES2002a,**Okay**,0,50.42,50.99,B ES2002a,**Right**,0,53.56,53.96,B*
*ES2002a,**Um well this is the kick-off meeting for our our project.**,0,55.415,60.35,B*

4. Спроба відтворити результати статті про застосування BERT ембедінгів до завдання сегментації. Результати статті значно перевершували результати аналогічних досліджень, однак згодом, незважаючи на чисельні спроби, повторити ці результати за допомогою методу, описаного в статті, не вдалося.

5. Пошук та виправлення помилок у BERT-методі. При аналізі причин невідповідності результатів, описаних у статті, та реальних результатів, було помічено деякі помилки в коді методу, які згодом були виправлені.

6. Пошук методів вдосконалення BERT-метода. У рамках цього етапу було оцінено ефективність різних мовних моделей, покращено роботу з ембедінгами, а також були задіяні альтернативні метрики оцінки, що дозволяють з більшою ефективністю підібрати гіперпараметри методу.

7. Додавання лінгвістичних ознак у метод. На цьому етапі були вивчені ознаки, здатні поліпшити сегментацію. В результаті було знайдено оптимальні переваги, які показують значущість цих ознак при ухваленні остаточного рішення про сегментацію.

Недоліками описаної моделі BERT є неможливість швидкої перевірки результатів через звертання коду до внутрішньої закритої бази даних, застарілі мовні моделі та завантаження векторних вбудов.

Першим кроком у оцінці результатів було створення набору даних, придатного для обробки, після чого результати були обчислені за всіма параметрами, зазначеними авторами статті (табл. 1).

Таблиця 1. Відтворення результатів при стандартному використанні BERT-моделі

	PK AMI	WD AMI	PK ICSI	WD ICSI
Заявлений результат	0.339	0.334	0.336	0.349
Відтворений результат	0.462	0.474	0.482	0.511

Можна відзначити, що різниця в результатах є надзвичайно великою, і її неможливо звести до різниці у версіях бібліотек або помилок під час обробки набору даних. Проблеми з реплікацією результатів також виникали у інших дослідників, як це можна побачити у коментарях у репозиторії проєкту.

Тим не менше, сам алгоритм виглядав розумно, тому було прийнято рішення відійти від метрик та проаналізувати саму сегментацію, намагаючись зрозуміти, чому в ній виникають помилки. Вихідний код програми був відредагований таким чином, щоб він не лише генерував метрики і порівнював їх, але й створював зрозумілий звіт з розділенням на сегменти, щоб було можливо проаналізувати роботу алгоритму. У результаті в остаточній сегментації були помічені деякі підозрілі ділянки. Приведемо приклади таких ділянок:

D: "And um, you know, when I think about what they are now, it’s better, but actually it’s still kind of, I dunno, like a massive junky thing on the table."

B: Still feels quite primitive. "Maybe we could think about how, could be more, you know, streamlined. S Maybe like a touch screen or something?"

D: "Something like that, yeah." B: Okay.

D: Or whatever would be technologically reasonable.

B: "Uh-huh, okay." Well I guess that’s up to our industrial designer.

Segment 7

B: "‘Cause it could b it could it could be that f it could be that functionally that doesn’t make it any better, but that just the appeal of of not having <disfmarker> It looks better."

D: "You know, these days there’s a r pe things in people’s homes are becoming more and more like chic, you know."

B: Yeah. "Um, nicer materials and might be Okay." Okay. D: be worth exploring anyway.

C: Uh.

B: "Right, well um so just to wrap up the next meeting’s gonna be in thirty minutes." So that’s about um about ten to twelve by my watch.

A: "Um so inbetween now and then, um as the industrial designer, you’re gonna be working on you know the actual working design of it Yep."

B: so y you know what you’re doing there. "Um for user interface, technical functions, I guess that’s you know like what we’ve been talking about, what it’ll actually do." "Um and uh marketing executive, you’ll be just thinking about what it actually <disfmarker> what, you know, what requirements it has to has to fulfil and you’ll all get instructions emailed to you, I guess ."

D: Okay.

B: Um. "<vocalsound> Yeah, so it’s th the functional design stage is next, I guess." <vocalsound> And uh and that’s the end of the meeting.

"So I got that little message a lot sooner than I thought I would, so

<disfmarker> Um. <vocalsound> <vocalsound> Before we wrap up, just to make sure we’re all on the same page here, Mm-hmm." "um, do we

<disfmarker> We were given sort of an example of a coffee machine or something, Uh-huh, yeah."

У цьому прикладі видно, що початок 7-го сегмента був встановлений у досить несподіваному місці (уривок виділено курсивом). По-перше, межа теми розриває мову одного спікера, що само по собі виглядає дивно. По-друге, 7-ма частина починається зі слова "cause" (тому що), яке фактично вказує на нерозривний зв’язок попереднього висловлювання з останнім. Розмітка теми, фактично, руйнує цей "місток" між двома думками того ж спікера. З іншого боку, ми бачимо, що 7-ма частина, здається, служить завершенням всього діалогу. Виділено жирним шрифтом ділянки, які це підтверджують. У першому уривку міститься вказівка на те, що зустріч скоро завершиться, і у другому уривку міститься явне згадування того, що зустріч завершилася. Таким чином, можна зробити висновок про те, що алгоритм сприйняв загальну картину розподілу на сегменти, але, з якоїсь причини, змістив межі.

І ще один приклад:

B: so we don’t really need to consider that in the functionality of the <disfmarker> of the remote control. "Um they’ve also suggested that we um we only use the remote control to control the television, not the V_C_R_, D_V_D_ or anything else." *"I think the worry is that if the project becomes too complex then it’ll affect um how long it takes us to get it into into production, the time to market."*

D: Okay.

Segment 3

B: "So um, we’re just gonna keep it simple and it’ll just control the TV." And the other thing was that the company want the corporate colour and slogan to be implemented in the new design. Um I’m not entirely sure what the corporate colour is. "It might be yellow, because there seems to be a lot of yellow everywhere."

D: "And the slogan, like the actual written slogan, or just to embody

the idea of the slogan?" "Well that’s the thing, I’m I’m not sure um

<vocalsound> uh th because on the the company website, uh what does it say <disfmarker> ‘Bout putting the fashion in electronics."

A: "Uh something <disfmarker> Yeah, Mm yeah."

B: "I mean do they <disfmarker> <vocalsound> Is that something they want actually written on it, ‘cause it’s quite long." "Um or yeah, just the idea, but I’m not sure." So that’s something we can discuss as well. "So those are the three things, just not to worry about teletext, uh only control the T_V_, and um and uh incorporate the uh colour and slogan of the company." "Um so is everybody okay with any of that, or do you want me to recap at all?"

A: "Nope, we’re all set."

B: "Right um, time for presentations then." Who would like to go first?

C: <vocalsound> I’ll go first. D: Sure.

B: "Okay, cool." "Alright um, can I st steal this from the back of your laptop? Uh <disfmarker> Oh yeah, of course, yeah." G go on ahead. C: <vocalsound> <gap> so this is the technical functions design. "Um

<vocalsound> <disfmarker> Right <gap> to do the um <vocalsound> the design I have I’ve had a look online, I’ve had a look at the homepage, which has given us um some insp inspiration from previous products."

У цьому прикладі ми спостерігаємо подібну ситуацію, коли сегмент 3, хоч і знаходиться на межі висловлювань учасника D та учасника B, насправді розриває думку учасника B (цей уривок виділено курсивом). Проте, через кілька рядків після позначеної межі теми ми бачимо уривок, виділений жирним шрифтом, коли тема дійсно змінюється (є явна вказівка на те, що починається частина з презентаціями).

Схожі недоліки повторювалися і в інших прикладах, що призвело до висновку, що з якоїсь причини межі всіх сегментів зсунуті відносно їх правильного положення. Було вирішено провести аналіз вихідного коду з метою виявлення можливої помилки.

Після аналізу вихідного коду було виявлено значущий недолік. У початковій моделі для обчислення лексичної зв’язності використовувався метод «плаваючого вікна», де розмір вікна був заданий як гіперпараметр. Вікно переміщувалося по тексту, для кожного вікна обчислювалося загальне векторне представлення участка, який воно охоплювало. Потім між цими представленнями обчислювалась косинусна відстань. Обчислення проводились так, що косинусні відстані призначалися, починаючи з позиції i, де i – розмір вікна. Отже, маючи, наприклад, 100 речень і вибравши розмір вікна 10, можна було обчислити 80 косинусних відстаней (мінус 10 з початку і з кінця).

Наступний крок: обчислення максимальної косинусної відстані, яка перевищувала певний заданий поріг. Таким чином, якщо, наприклад, перша косинусна відстань перевищувала заданий поріг, то модель передбачала, що в першому реченні відбувається зміна теми. Однак це було неправильним висновком, оскільки перша косинусна відстань фактично відповідала б 10-му реченню через коригування на розмір вікна. У початковому рішенні цього коригування не було, тому було прийнято рішення перерахувати результати, враховуючи це коригування. У прикладах можна побачити зміни, які відбулися після виправлення помилки.

У наступному прикладі видно, що початок 7-го сегмента збігається з оголошенням одного з учасників про те, що зустріч наближається до завершення. Незважаючи на те, що межа теми все ще розриває мову одного з учасників, тепер цей розрив виглядає логічним, оскільки виходить, що 7-а частина є заключенням, коли в діалозі підводяться підсумки зустрічі. Жовтим кольором позначена та межа сегмента, яку встановив попередній алгоритм.

D: "And um, you know, when I think about what they are now, it’s better, but actually it’s still kind of, I dunno, like a massive junky thing on the table."

B: Still feels quite primitive. "Maybe we could think about how, could be more, you know, streamlined. S Maybe like a touch screen or something?"

D: "Something like that, yeah." B: Okay.

D: Or whatever would be technologically reasonable.

B: "Uh-huh, okay." Well I guess that’s up to our industrial designer. ‘Cause it could b it could it could be that f it could be that functionally that doesn’t make it any better,

but that just the appeal of of not having <disfmarker> It looks better."

D: "You know, these days there’s a r pe things in people’s homes are becoming more and more like chic, you know."

B: Yeah. "Um, nicer materials and might be Okay." Okay. D: be worth exploring anyway.

C: Uh.

B: "Right, well um so just to wrap up the next meeting’s gonna be in thirty minutes." So that’s about um about ten to twelve by my watch. A: "Um so inbetween now and then, um as the industrial designer, you’re gonna be working on you know the actual working design of it Yep."

B: so y you know what you’re doing there. "Um for user interface, technical functions, I guess that’s you know like what we’ve been talking about, what it’ll actually do." "Um and uh marketing executive, you’ll be just thinking about what it actually <disfmarker> what, you know, what requirements it has to has to fulfil and you’ll all get instructions emailed to you, I guess."

D: Okay.

B: Um. "<vocalsound> Yeah, so it’s th the functional design stage is next, I guess."

Segment 7

B: <vocalsound> And uh and that’s the end of the meeting.

"So I got that little message a lot sooner than I thought I would, so

<disfmarker> Um. <vocalsound> <vocalsound> Before we wrap up, just to make sure we’re all on the same page here, Mm-hmm." "um, do we

<disfmarker> We were given sort of an example of a coffee machine or something, Uh-huh, yeah."

У наступному прикладі також видно, що межа теми зсунулася порівняно з попередньою версією (позначено жирним шрифтом) на більш логічний відрізок. Тепер вона приблизно відповідає початку презентацій у діалозі.

B: so we don't really need to consider that in the functionality of the <disfmarker> of the remote control. "Um they've also suggested that we um we only use the remote control to control the television, not the V_C_R_, D_V_D_ or anything else ." "I think the worry is that if the project becomes too complex then it'll affect um how long it takes us to get it into into production, the time to market."

D: Okay.

B: "So um, we're just gonna keep it simple and it'll just control the T_V."

And the other thing was that the company want the corporate colour and slogan to be implemented in the new design. Um I'm not entirely sure what the corporate colour is. "It might be yellow, because there seems to be a lot of yellow everywhere."

A: "Uh something <disfmarker> Yeah, Mm yeah."

B: "I mean do they <disfmarker> <vocalsound> Is that something they want actually written on it, 'cause it's quite long." "Um or yeah, just the idea, but I'm not sure." So that's something we can discuss as well. "So those are the three things, just not to worry about teletext, uh only control the T_V_, and um and uh incorporate the uh colour and slogan of the company." "Um so is everybody okay with any of that, or do you want me to recap at all?"

A: "Nope, we're all set."

B: "Right um, time for presentations then." Who would like to go first?

C: <vocalsound> I'll go first.

Segment 3

D: Sure.

B: "Okay, cool." "Alright um, can I st steal this from the back of your laptop? Uh <disfmarker> Oh yeah, of course, yeah." G go on ahead. C: <vocalsound> <gap> so this is the technical functions design. "Um

<vocalsound> <disfmarker> Right <gap> to do the um <vocalsound> the design I have I've had a look online, I've had a look at the homepage, which has given us um some insp inspiration from previous products."

Після аналізу метрик було помічено, що результати виправленої версії покращали (табл. 2).

Таблиця 2. Результати після введення виправлення на розмір вікна

	PK AMI	WD AMI	PK ICSI	WD ICSI
Заявлений результат	0.339	0.334	0.336	0.349
Відтворений результат	0.421	0.433	0.453	0.473

Класичні метрики Pk і WindowDiff під час тестування виявили один значущий недолік [10]: їх значення дуже сильно залежали від кількості сегментів, на які був розбитий текст. Оскільки загальна кількість сегментів зазвичай невідома наперед, виникає потреба в визначенні та подальшому налаштуванні порогу, що вказує на те, наскільки великим повинно бути відмінність у семантичній зв'язності, щоб вважати цю відмінність зміною теми (табл. 3).

Таблиця 3. Залежність метрик PK та WindowDiff від порога зміни теми

	thr=0.3	thr=0.4	thr=0.5	thr=0.6	thr=0.7	thr=0.8	thr=0.9	thr=1
PK AMI	0.511	0.49	0.467	0.451	0.439	0.429	0.422	0.393
WD AMI	0.571	0.531	0.495	0.469	0.451	0.437	0.428	0.393
PK ICSI	0.58	0.522	0.469	0.423	0.415	0.406	0.39	0.331
WD ICSI	0.714	0.604	0.52	0.453	0.432	0.415	0.395	0.331

Можна відмітити, що чим вище значення порогу, тим кращі результати метрик (іншими словами, менше ймовірність помилки). Отже, якість метрик прямо залежить від кількості тем, які виділяє алгоритм. Певною мірою це зрозуміло, оскільки метрики PK і WindowDiff відображають ймовірність помилки сегментації. Очевидно, що чим менше тем виділяє алгоритм, тим менше помилок він допускає.

Підсумком дослідження стало створення інструменту, що реалізує описаний вище метод. Головне призначення цього інструменту – використання як складовий блок конвеєрів з обробки даних. Оскільки програма вирішує завдання сегментації діалогів кількох учасників, вхідні дані мають бути відповідним чином розмічені. Докладніше про розмітку – в описі вхідних параметрів.

Інструмент є консольною програмою, на вхід якої подається розмічений текст (і низка параметрів, деякі з яких необов'язкові), а на виході JSON файл, в якому вказані тимчасові мітки отриманих сегментів. Також існує можливість згенерувати розмітку, де весь текст буде розбитий на сегменти. Так можна легко проконтролювати роботу інструменту.

Для оцінки точності визначення сегментів тексту було проаналізовано саму сегментацію аби по-перше, визначити помилки, і по-друге, виправити помилки для вірної сегментації тексту. У ході експериментів було проаналізовано роботу інструмента на двох моделях, які запропоновані у роботі. Було використано декілька варіантів сегментації, які включали в себе використання лінгвістичних ознак (f), використання препроцесингу (p), зміна значення порога (t – 0; 1; 0.6).

Для оцінки точності роботи інструмента було використано формулу точності:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN},$$

де TP – кількість правильно визначених сегментів (True Positives),

TN – кількість правильно визначених відсутностей сегментів (True Negatives),

FP – кількість неправильно визначених сегментів (False Positives),

FN – кількість неправильно пропущених сегментів (False Negatives).

Оскільки значення TN (кількість правильно визначених відсутностей сегментів) та FN (кількість неправильно пропущених сегментів) не спостерігалась, їх значення взято за 0.

При тестувані було отримано різні результати:

1) 7 сегментів мають результат 100 % ;

2) 4 сегменти мають результати від 90% до 60 %;

3) 4 сегменти мають результат 50 % .

Помилки містяться на початку нового сегмента, які становлять наявність від 1 до 4 реплік, які відносяться до попереднього сегмента, що і призвело до таких результатів (табл. 4).

Таблиця 4. Результати оцінки точності моделі paraphrase-multilingual-mpnet-base-v2 (sbert)

Значення при сегментації	Кількість виведених сегментів	Кількість правильних сегментів	Результат у відсотках
t-0.6; f	4	3	75%
t-0; f	4	2	50%
t-1; f	1	1	100%
t-0; p	7	5	71%
t-1; p	1	1	100%
t-0	7	6	86%
t-1	1	1	100%

Оскільки результати отримані різні і у великій кількості, було прийнято рішення підрахувати середню оцінку кожної моделі. Отримані результати наступні: середня оцінка моделі paraphrase-multilingual-mpnet-base-v2 (sbert) становить 83 %, середня оцінка моделі roberta-base (mbert) становить 73 %.

Обидві моделі показали гарні результати, але модель sbert працює трохи краще, про що свідчать результати оцінки точності. Помилки присутні при виводі тексту з використанням обох моделей, але модель paraphrase-multilingual-mpnet-base-v2 (sbert) містить їх трохи менше.

Для оцінки повноти визначення (сегментів тексту) було проаналізовано безпосередньо текст діалогу. У ході експериментів було проаналізовано роботу інструмента на двох запропонованих моделях. Було використано декілька варіантів сегментації, які включали в себе використання лінгвістичних ознак (f), використання препроцесингу (p), зміна значення порога (t – 0, 1, 0.6). Результати експериментів наведено у табл. 5.

Таблиця 5. Результати оцінки повноти моделі paraphrase-multilingual-mpnet-base-v2 (sbert)

Значення при сегментації	Кількість правильних сегментів	Результат у відсотках
t-0.6; f	3	100%
t-0; f	2	100%
t-1; f	1	100%
t-0; p	5	100%
t-1; p	1	100%
t-0	6	100%
t-1	1	100%

При тестувані було отримано результати усіх текстів у 100%. Це свідчить про те, що текст діалогу виводиться коректно, усі репліки відповідають спікерам, які їх вимовляють.

Висновки. Підсумком дослідження став готовий інструмент для сегментації, заснований на роботі з ембедінгами BERT й характерних ознаках діалогів кількох учасників, розроблений на мові програмування Python. Це відповідає меті роботи – створенню повноцінного відкритого інструменту, що дозволяє виконувати сегментацію діалогів. Інструмент показує хороші результати, які свідчать про те, що цей підхід є перспективним напрямом, який можна розвивати та вдосконалювати. Працюючи над цією проблемою, були проаналізовані існуючі підходи до її вирішення та проаналізовано основні моделі сегментації. Для вирішення проблеми сегментації було також проаналізовано відомі датасети та обрано найбільш підходящі для роботи.

При перевірці роботи алгоритму було виявлено декілька помилок, які порушували роботу інструмента. Під час тестування моделі були помічені приклади порушення коли лексика при обговорені могла змінитися, а тема лишалась незмінною. Тому до моделі BERT було вирішено додати лінгвістичні ознаки, які б допомогали модулю лексичної зв'язності приймати рішення, оскільки самих ембедінгів недостатньо для ефективної сегментаціїї. Були визначені наступні лінгвістичні ознаки: ознака лексичної зв'язності (BERT), ознака пауз у діалозі, ознака накладання мови говорячих, ознака зміни говорячого, ключові фрази та зглажування коеффіцієнтів.

Проте, слід зазначити, що у деяких із останніх робіт з теми сегментації діалогів автори працюють із іншими корпусами, наприклад, DialSeg і DocDial. Є роботи, у яких метрики перевершують ті, які були отримані у цій роботі, але експерименти проводилися за допомогою різних корпусів, тому достовірно порівнювати ці моделі не можна. Перевірку запропонованого у роботі алгоритму за допомогою інших корпусів для порівняння з іншими моделями можна розглядати як завдання на майбутнє.

Крім цього, ще одним напрямком розвитку вважатимуться адаптування алгоритму до сегментації нерозмічених діалогів. Теоретично модель можна застосувати до текстів діалогів, у яких немає вказівки на час виголошення тієї чи іншої репліки. У цьому випадку не буде працювати ознака наявності пауз, але всі інші ознаки можуть бути обчислені (за умови, що текст міститиме імена тих, хто говорить). Хоча ознака наявності пауз робить істотний внесок у сегментацію, і результати, ймовірно, будуть гіршими, але здатність інструменту працювати з нерозміченим текстом, безсумнівно, стане великим плюсом.

References

1. Zeng, Y., Li, J., Zhao, L., Kang, Y., Sun, C., Zhang, Q., & He, X. (2021). *Unsupervised dialogue summarization with topic-aware ranking and context modeling*. In Proc. of AAAI 2021, 35(16), 14674–14682.

2. Inan, H., Rungta, R., & Mehdad, Y. (2022). *Structured summarization: Unified text segmentation and segment labeling as a generation task*. In Findings of EACL 2022 (pp. 883–893). Association for Computational Linguistics.

3. Zhang, Y., Liu, Y., Xu, Y., Zhu, C., & Zeng, M. (2022). *DialogLM: Pre-trained model for long dialogue understanding and summarization*. In Proc. of AAAI 2022, 36(10), 10965–10973.

4. Xing, L., & Carenini, G. (2021). *Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring*. In Proc. of SIGDIAL 2021 (pp. 167–177). Association for Computational Linguistics.

5. Solbiati, A., Yvon, F., & Joly, A. (2021). *Dialogue topic segmentation for meeting transcripts using BERT embeddings*. In Proc. of IEEE SLT 2021 (pp. 757–764). IEEE.

6. Matsumoto, K., Sasayama, M., & Kirihara, T. (2022). *Topic break detection in interview dialogues using sentence embedding of utterance and speech intention based on multitask neural networks*. Sensors, 22(2), 694.

7. Nair, I., Garimella, A., Srinivasan, B. V., Modani, N., Chhaya, N., Karanam, S., & Shekhar, S. (2023). *A neural CRF-based hierarchical approach for linear text segmentation*. In Findings of ACL 2023 (pp. 883–893). Association for Computational Linguistics.

8. Wu, C.-S., Hoi, S., Socher, R., & Xiong, C. (2020). *TOD-BERT: Pre-trained natural language understanding for task-oriented dialog*. In Proc. of EMNLP 2020 (pp. 917–929). Association for Computational Linguistics.

9. Das, R., Goyal, P., Kale, D., & Hakkani-Tür, D. (2024). *Structured open-domain dialogue segmentation and state tracking*. (Preprint arXiv:2403.00027).

10. Feng, S., et al. (2021). *MultiDoc2Dial: Modeling dialogues grounded in multiple documents*. In Proc. of EMNLP 2021 (pp. 5847–5860). Association for Computational Linguistics.

Надійшла (received) 10.06.2025

Відомості про авторів / About the Authors

***Бабкова Надія Вікторівна (Nadiia Babkova)* –** Національний технічний університет «Харківський політехнічний інститут», кандидат технічних наук, доцент, завідувач кафедри інтелектуальних комп'ютерних систем; Харків, Україна; ORCID: https://orcid.org/0000-0002-2200-7794

***Гулієва Діна Олександрівна (Dina Huliieva)* –** Національний технічний університет «Харківський політехнічний інститут», кандидат філологічних наук, доцент, доцент кафедри інтелектуальних комп'ютерних систем; Харків, Україна; ORCID: https://orcid.org/0000-0001-8310-745X

***Кочуєва Зоя Анатоліївна (Zoia Kochuieva)* –** Національний технічний університет «Харківський політехнічний інститут», кандидат технічних наук, доцент, доцент кафедри інтелектуальних комп'ютерних систем; Харків, Україна; ORCID: https://orcid.org/0000-0002-4300-3370

***Угольнікова Наталія Сергіївна (Nataliia Ugolnikova)* –** Національний технічний університет «Харківський політехнічний інститут», кандидат філологічних наук, доцент кафедри інтелектуальних комп'ютерних систем; Харків, Україна; ORCID: https://orcid.org/0000-0003-2322-0922