УДК 004.93+004.852

Замуруев Александр Владимирович, аспирант, Российский Университет Транспорта (МИИТ), г. Москва
Мымрин Павел Михайлович, аспирант, Российский Университет Транспорта (МИИТ), г. Москва

АРХИТЕКТУРА ТРАНСФОРМЕРНОЙ МОДЕЛИ ДЛЯ РАСПОЗНАВАНИЯ ИМЕНОВАННЫХ СУЩНОСТЕЙ В ТЕКСТЕ

Аннотация

В статье рассматривается архитектура трансформерной модели для распознавания именованных сущностей в тексте. Определены основные архитектурные параметры модели и установлены диапазоны их значений: число трансформерных блоков, число голов внимания, размерность одной головы внимания, размер блока для блочного механизма внимания, размер окна для оконного механизма внимания. Проведен экспериментальный анализ влияния указанных параметров на качество распознавания именованных сущностей. Полученные результаты позволяют определить область значений параметров, обеспечивающую более высокое качество распознавания именованных сущностей в тексте в рамках рассматриваемой архитектуры.

Annotation

The paper examines the architecture of a transformer model for named entity recognition in text. The main architectural parameters of the model are identified and the ranges of their values are established: the number of transformer blocks, the number of attention heads, the dimensionality of a single attention head, the block size for the block attention mechanism, and the window size for the window attention mechanism. An experimental analysis of the influence of these parameters on the quality of named entity recognition is conducted. The obtained results make

it possible to determine the range of parameter values that ensures higher quality of named entity recognition in text within the framework of the considered architecture.

Ключевые слова: распознавание именованных сущностей, трансформер, классификация, трансформер, механизм внимания, архитектурные параметры, нейронная сеть.

Keywords: named entity recognition, transformer, classification, attention mechanism, architectural parameters, neural network

ВВЕДЕНИЕ

Распознавание именованных сущностей (Named Entity Recognition, NER) является одной из фундаментальных задач обработки естественного языка, направленной на выделение и классификацию упоминаний объектов в тексте. Например, в текстах, полученных из систем автоматического распознавания речи, распознавание именованных сущностей может использоваться для выявления персональных данных, в обращениях пользователей – для выделения названий продуктов и организаций, в юридических документах – для поиска сторон договора, адресов, дат и реквизитов.

Классические подходы к распознаванию именованных сущностей могут быть недостаточно эффективными при работе с разнородными текстовыми данными. Методы, основанные на правилах, требуют трудоемкого ручного описания всевозможных вариантов написания сущностей и контекстов, в котором они встречаются. Ранние нейросетевые модели обладают ограниченной способностью к анализу широкого контекста и часто склонны к заучиванию текста обучающей выборки.

Современные нейросетевые подходы к распознаванию именованных сущностей основаны на токенизации текста и последующим формированием векторных представлений токенов. Для учета контекста в таких представлениях широко используется архитектура трансформера, ключевым

компонентом которой является механизм внимания (самовнимания), позволяющий модели сопоставлять токены между собой.

Целью данной работы является формирование архитектуры трансформерной модели для задачи распознавания именованных сущностей в тексте и определение допустимых диапазонов параметров модели.

1 АРХИТЕКТУРА ТРАНСФОРМЕРНОЙ МОДЕЛИ

В основе работы лежит архитектура модели на базе трансформера, предназначенная для задачи распознавания именованных сущностей. Каждому токену входной последовательности текста необходимо сопоставить метку, определяющую принадлежность токена к определенному типу именованной сущности. Пусть входная последовательность текста после токенизации имеет вид

$$X = (x_1, x_2, \dots, x_n), \quad (1)$$

где x_i – i-й токен, n – длина последовательности. Для каждого токена требуется определить метку из множества меток именованных сущностей

$$Y = \{y_1, y_2, \dots, y_K\}, \quad (2)$$

где y_j – j-ая метка именованной сущности, K – общее число меток именованных сущностей. Общая схема модели может быть представлена как

$$X \to E(X) \to T_1, \dots, T_L \to H \to A(H) \to Z \to C(Z), \quad (3)$$

где X – входная последовательность токенов, $E(X)$ – векторное представление токенов, $T_1, \dots, T_L$ – последовательность трансформерных блоков, H – промежуточные скрытые состояния, $A(H)$ – дополнительные слои, Z – итоговые скрытые состояния, $C(Z)$ – классификационный слой меток именованных сущностей для каждого токена.

Базовая архитектура модели приведена на рисунке 1.

Рисунок 1 – Архитектура модели

В данной работе архитектура рассматривается как последовательность нескольких основных этапов: формирование входного представления текста, обработка последовательности трансформерными блоками, применение дополнительных слоев, классификация каждого токена.

2 ФОРМИРОВАНИЕ ВХОДНОГО ПРЕДСТАВЛЕНИЯ ТЕКСТА

Первым этапом является преобразование исходного текста во входное признаковое пространство модели. Исходный текст разбивается на токены (токенизируется), каждому токену сопоставляется числовой идентификатор словаря токенизатора, после чего идентификатор преобразуется в векторные представления. Обозначим размерность вектора представления токена как d_m, тогда преобразование будет иметь вид

$$T \to X(T) = (x_1, x_2, \dots, x_n) \to Id(X) = (Id_1, Id_2, \dots, Id_n) \to E(Id) = (E_1, E_2, \dots, E_n), \quad (4)$$

где T – входная последовательность текста, $X(T)$ – последовательность токенов, $Id(X)$ – последовательность числовых идентификаторов словаря токенизатора, $E(Id)$ – последовательность векторов токенов размерностью d_m, n – длина последовательности.

3 ТРАНСФОРМЕРНЫЙ БЛОК И МЕХАНИЗМЫ ВНИМАНИЯ

Основной частью модели является последовательность трансформерных блоков. Каждый трансформерный блок содержит механизм многоголового внимания (самовнимания), позволяющий сопоставлять токены последовательности между собой. В данной работе были рассмотрены блочный, оконный, PaTH, PaTH-FoX механизмы внимания.

Для всех рассматриваемых вариантов внимания используются общие понятия запросов, ключей и значений. Каждый токен формирует запрос – какую информацию он ищет в других токенах, ключ – какую информацию он сам может предоставить. Взаимодействие запроса и ключа определяет вес внимания – степень важности одного токена для другого. Итоговое представление токена – взвешенная сумма значений всех токенов. Использование нескольких голов внимания позволяет модели одновременно учитывать различные типы зависимостей.

Пусть оценка внимания между токенами i и j задается выражением

$$s_{ij} = \frac{q_i^T k_j}{\sqrt{d_k}}, \quad (5)$$

где q_i – вектор запроса i-го токена, k_j – вектор ключа j-го токена, d_k – размерность проекции исходного скрытого представления токена в одной голове внимания.

Для токена i формируется вектор оценок внимания s_i ко всем токенам последовательности

$$s_i = (s_{i1}, s_{i2}, \dots, s_{in}), \quad (6)$$

где n – длина последовательности. Нормированный вес внимания $softmax(s_i)_j$ к токену j определяется как

$$softmax(s_i)_j = \alpha_{ij} = \frac{e^{s_{ij}}}{\sum_{t=1}^{n} e^{s_{it}}}. \quad (7)$$

Итоговое представление токена i вычисляется как

$$o_i = \sum_{j=1}^{n} \alpha_{ij} v_j, \quad (8)$$

где v_j – вектор значения j-го токена.

Базовая операция внимания в общем виде задается выражением

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (9)$$

где Q – матрица запросов, K – матрица ключей, V – матрица значений, K^T – транспонированная матрица ключей K, d_k – размерность проекции исходного скрытого представления токена в одной голове внимания.

В случае многоголового внимания вычисления выполняются несколькими независимыми головами внимания. Пусть число голов равно M. Для каждой головы формируются собственные матрицы запросов, ключей, значений

$$Q^{(m)} = HW_Q^{(m)}, K^{(m)} = HW_K^{(m)}, V^{(m)} = HW_V^{(m)}, \quad (10)$$

где H – матрица скрытых состояний токенов, $W_Q^{(m)}$, $W_K^{(m)}, W_V^{(m)}$ – обучаемые матрицы проекций для m-й головы. Выходным значением a_m m-й головы является

$$a_m = Attention\left(Q^{(m)}, K^{(m)}, V^{(m)}\right). \quad (11)$$

Результаты всех голов объединяются операцией конкатенации

$$concat(a_1, a_2, \dots, a_M) = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_M \end{bmatrix}. \quad (12)$$

Итоговый выход многоголового внимания вычисляется как

$$MHA(H) = concat(a_1, a_2, \dots, a_M)W_O, \quad (13)$$

где W_O – обучаемая матрица выходного преобразования.

Обозначим множество токенов, доступных для внимания токена i как

$$\Omega_i \subseteq \{1, \dots, n\}, \quad (14)$$

где n – длина входной последовательности [1].

Блочный механизм внимания реализует идею локального внимания с установленным размером блока. Входная последовательность разбивается на блоки размера B. Токен i взаимодействует только с токенами внутри того же блока. Оценка внимания в блочном механизме для головы m имеет вид

$$s_{ij}^{(m)} = \frac{\left(q_i^{(m)}\right)^T k_j^{(m)}}{\sqrt{d_k}} + b_{i-j}^{(m)}, j \in \Omega_i^{block}, \quad (15)$$

где $q_i^{(m)}$ – вектор запроса в m-й голове, $k_j^{(m)}$ – вектор ключа в m-й голове, $b_{i-j}^{(m)}$ – обучаемое позиционное смещение.

Такой механизм позволяет модели учитывать взаимное расположение токенов в пределах блока, однако взаимодействие между разными блоками отсутствует. Это ограничивает способность модели захватывать глобальный контекст, но позволяет эффективно обрабатывать короткие локальные зависимости, что может быть полезно для выделения сущностей, состоящих из нескольких смежных токенов. Идея ограничения области внимания локальными или блочными структурами используется в разреженных вариантах трансформеров, предназначенных для обработки длинных последовательностей [2].

Оконный механизм внимания предназначен для совмещения локальной обработки последовательности и передачи глобального контекста. Для токена i локальная оценка внимания к токену j внутри окна задается как

$$s_{ij}^{(local_m)} = \frac{\left(q_i^{(local_m)}\right)^T k_j^{(local_m)}}{\sqrt{d_k}}, j \in \Omega_i^{local}. \quad (16)$$

Локальное представление формируется как

$$o_i^{(local_m)} = \sum_{j \in \Omega_i^{local}} \alpha_{ij}^{(local_m)} v_j^{(local_m)}. \quad (17)$$

Для обеспечения обмена информацией между разными окнами вводятся G_n обучаемых глобальных токенов G

$$G = (g_1, g_2, \dots, g_G). \quad (18)$$

Для глобального токена l оценка внимания ко всем токенам j задается как

$$s_{lj}^{(global_m)} = \frac{\left(q_l^{(global_m)}\right)^T k_j^{(global_m)}}{\sqrt{d_k}}, j \in [1; n], \quad (19)$$

где n – длина последовательности. Глобальное представление формируется как

$$\tilde{g}_l^{(m)} = \sum_{j=1}^{n} \alpha_{lj}^{(global_m)} v_j^{(global_m)}. \quad (20)$$

Далее обновленные глобальные токены передают информацию локальным токенам последовательности

$$s_{il}^{(global \to local_m)} = \frac{\left(q_i^{(local_m)}\right)^T \tilde{k}_l^{(global_m)}}{\sqrt{d_k}}, l \in [1; G_n]. \quad (21)$$

Глобальная составляющая итогового представления локального токена i вычисляется как

$$o_i^{(global_m)} = \sum_{l=1}^{G_n} \alpha_{il}^{(global \to local_m)} \tilde{v}_l^{(global_m)}. \quad (22)$$

Итоговое представление локального токена в m-й голове формируется как

$$o_i^{(m)} = o_i^{(local_m)} + o_i^{(global_m)}. \quad (23)$$

Таким образом, оконный механизм внимания состоит из двух частей. Локальная часть отвечает за обработку зависимостей внутри окна, а

глобальные токены обеспечивают передачу информации между различными окнами, что позволяет учитывать сжатый глобальный контекст. Идея совмещения локального и глобального внимания также используется в архитектурах для обработки длинных последовательностей [3].

PaTH – механизм внимания, основанный на динамическом позиционном преобразовании. В отличие от блочного и оконного механизмов, где область взаимодействия задается локальной структурой, PaTH изменяет саму оценку внимания между токенами с учетом промежуточного контекста.

Пусть для каждого токена t в m-й голове строится матрица позиционного преобразования

$$R_t^{(m)} = \varphi^{(m)}(h_t), \quad (24)$$

где h_t – скрытое представление токена t, $\varphi^{(m)}$ – обучаемое преобразование для m-й головы внимания. Для пары токенов i и j рассматривается путь между ними. Если $i < j$, то путь между токенами можно представить как

$$i \to i+1 \to \cdots \to j. \quad (25)$$

Позиционное преобразование между токенами i и j получается путем накопления преобразований токенов на пути между i и j

$$P_{i\to j}^{(m)} = R_{i+1}^{(m)} R_{i+2}^{(m)} \ldots R_j^{(m)}. \quad (26)$$

Если $i > j$, произведение строится в обратном направлении. Оценка внимания в механизме PaTH имеет вид

$$s_{ij}^{(m)} = \frac{\left(q_i^{(m)}\right)^T P_{i\to j}^{(m)} k_j^{(m)}}{\sqrt{d_k}}. \quad (27)$$

Итоговое представление токена формируется как

$$o_i^{(m)} = \sum_{j\in\Omega_i} \alpha_{ij}^{(m)} v_j^{(m)}. \quad (28)$$

Таким образом, механизм внимания PaTH позволяет модели изменять оценку связи между токенами в зависимости от контекста промежуточного фрагмента.

PaTH-FoX расширяет механизм PaTH за счет адаптивного забывания. Для каждой пары токенов i и j в m-й голове вводится коэффициент забывания $f_{ij}^{(m)} \in (0;1]$, который определяет насколько сильно токен j должен учитываться при формировании представления токена i. С учетом забывания оценка внимания определяется как

$$s'^{(m)}_{ij} = s_{ij}^{(m)} + \ln\left(f_{ij}^{(m)} + \varepsilon\right), \quad (29)$$

где $s_{ij}^{(m)}$ – оценка внимания PaTH, ε – константа для численной устойчивости. Далее веса внимания определяются по модифицированным оценкам

$$\alpha_{ij}^{(m)} = softmax\left(s'^{(m)}_{i}\right)_j. \quad (30)$$

Итоговое представление токена имеет вид

$$o_i^{(m)} = \sum_{j\in\Omega_i} \alpha_{ij}^{(m)} v_j^{(m)}. \quad (31)$$

Так как $f_{ij}^{(m)} \leq 1$, добавление логарифмического слагаемого уменьшает оценку внимания для токенов с малым коэффициентом забывания. Таким образом, PaTH-FoX сохраняет динамическое позиционное преобразование PaTH и ослабляет влияние нерелевантных токенов [4].

После всех трансформерных блоков к промежуточным скрытым состояниям H могут применяться дополнительные слои $A(H)$, формирующие итоговые скрытые состояния Z.

4 ПАРАМЕТРЫ МОДЕЛИ

Для построения трансформерной модели с рассматриваемыми механизмами внимания необходимо определить исследуемый диапазон параметров, к которым относятся: размерность вектора представлений токена d_m, число трансформерных блоков L, число голов внимания M, размерность одной головы внимания d_k, размер блока для блочного механизма внимания B, размер окна для оконного механизма внимания W.

Распространенные варианты представления текста (такие как Multilingual BERT, XLM-RoBERTa) предоставляют векторы представления токенов размерностью $d_m = 768$. Использование этой размерности обеспечит возможность использования подобных предобученных векторизаторов.

При использовании многоголового внимания размерность одной головы определяется как

$$d_k = \frac{d_m}{M}, \quad (32)$$

где M – количество голов внимания. В качестве исследуемого диапазона значений количества голов был выбран $M \in \{12; 24; 48\}$, тогда размерность одной головы внимания составляет $d_k \in \{64; 32; 16\}$ соответственно. В статье «Attention Is All You Need» в качестве базовой конфигурации используется $d_k = 64$ [1]. В статье «PaTH Attention: Position Encoding via Accumulating Householder Transformations» также используется размерность головы внимания $d_k = 64$ в основных экспериментах [4]. Увеличение числа голов, в свою очередь, может снижать качество в результате уменьшения размерности одной головы. В статье «Low-Rank Bottleneck in Multi-head Attention Models» показано, что при увеличении числа голов M в модели BERTLARGE с 8 до 32, что соответствует уменьшению размерности головы со 128 до 32, значение SQuAD F1 снизилось с 90.89 до 90.45, SQuAD EM – с 84.10 до 83.48, а MNLI accuracy – с 85.0 до 84.4 [5]. Для рассматриваемой архитектуры число голов $M = 24$ соответствует размерности головы внимания $d_k = 32$, значение числа голов $M = 48$ позволяет дополнительно оценить влияние меньшей размерности головы внимания $d_k = 16$ на качество.

В статье «VEE-BERT: Accelerating BERT Inference for Named Entity Recognition via Vote Early Exiting», посвященной ускорению BERT моделей для задач распознавания именованных сущностей, наилучшее значение метрики качества micro-F1 достигалось на девятом трансформерном слое [6]. Таким образом, число трансформерных блоков рассматривалось в пределах

двенадцати $L \in \{1; 2; \dots; 11; 12\}$, в данной работе был рассмотрен диапазон трансформерных блоков $L \in \{2; 4; 6; 8; 10; 12\}$.

В реализованной архитектуре максимальная длина входной последовательности ограничена значением $n = 512$, что является стандартной верхней границей длины последовательности в BERT-подобных моделях. Диапазон размера блока для блочного механизма внимания был установлен $B \in \{16; 32; 64; 128; 256\}$, что покрывает масштабы $\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$ относительно установленной максимальной длины последовательности. Это позволяет оценить влияние масштаба контекста от области, сопоставимой с длиной короткого предложения или фрагмента текста до половины допустимой длины последовательности.

Диапазон размеров окон для оконного механизма внимания был также установлен в $W \in \{16; 32; 64; 128; 256\}$ по аналогичной логике.

5 ОПИСАНИЕ НАБОРА ДАННЫХ

Для обучения и оценки использовался русскоязычный набор данных Collection5, размеченный именованными сущностями в формате BIO (Begin, Inside, Outside) [7]. В формате BIO каждое слово в тексте помечается тегом, указывающим на принадлежность к определенному классу сущности (метке). Тег состоит из двух частей: одна указывает на тип сущности, а другая – на положение слова внутри сущности. Тег «B-» используется для первого слова сущности, тег «I-» – для всех последующих слов сущности, тег «O» – для всех слов, не относящихся к сущности.

Для набора данных Collection5 общее количество символов – 1708776, общее количество слов – 220791. Распределение длин текстов в символах приведено на рисунке 2.

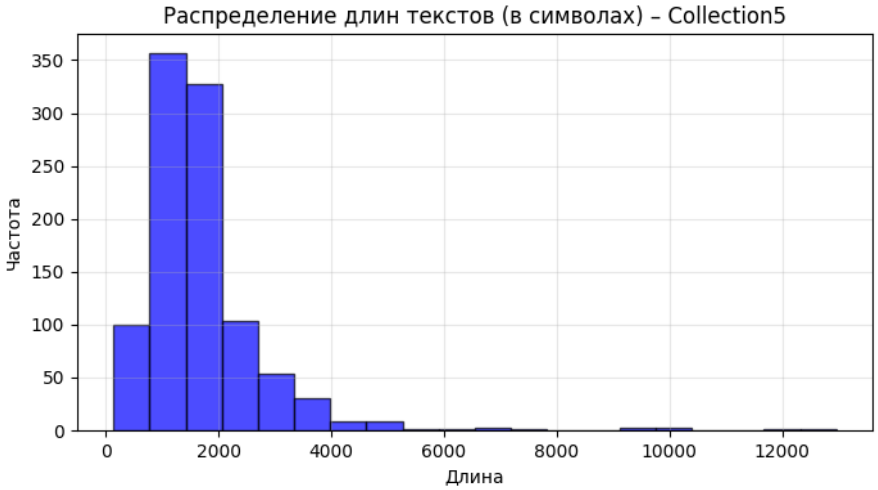

Рисунок 2 – Распределение длин текстов в символах для Collection5

В таблице 1 приведено количество и наименование тегов для набора данных Collection5.

Таблица 1 – Количество тегов в Collection5.

Тег	Количество
O	182600
B-GEOPOLIT	3945
B-LOC	2906
B-MEDIA	1061
B-ORG	5346
B-PER	10187
I-GEOPOLIT	361
I-LOC	1528
I-MEDIA	782
I-ORG	5083
I-PER	6992

6 ОБУЧЕНИЕ

Все модели разработаны на языке программирования Python с использованием библиотеки Torch. Обучение выполнялось с использованием графического процессора NVIDIA GeForce RTX 4070 с размером батча 16. Обучающая и проверяющая выборки были сформированы в соотношении 80/20.

Для оценки качества моделей использовались метрики Accuracy (доля правильных предсказаний) и F1-мера. Accuracy отражает долю токенов, для которых предсказанная метка совпала с истинной. Для задачи распознавания именованных сущностей метрика Accuracy не является достаточной, так как большая часть токенов, как правило, не являются именованными сущностями. Для более содержательной оценки использовалась F1-мера. Для каждой метки y вычислялись precision и recall

$$precision_y = \frac{TP_y}{TP_y + FP_y}, \quad (31)$$

$$recall_y = \frac{TP_y}{TP_y + FN_y}, \quad (32)$$

где TP_y – число токенов, правильно отнесенных к метке y; FP_y – число токенов, ошибочно отнесенных к метке y; FN_y – число токенов, истинно принадлежащих к метке y, но ошибочно отнесенных к другой метке. F1-мера для метки y определяется как гармоническое среднее precision и recall

$$F1_y = \frac{2 \cdot precision_y \cdot recall_y}{precision_y + recall_y}. \quad (33)$$

В работе использовалась взвешенная F1-мера, учитывающая различную частоту меток в данных

$$F1_{weighted} = \sum_{y=1}^{K} w_y F1_y, \quad (34)$$

где

$$w_y = \frac{N_y}{N}, \quad (35)$$

где N_y – количество токенов с меткой y, N – общее количество учитываемых токенов.

После определения исследуемых диапазонов архитектурных параметров был проведен предварительный экспериментальный анализ. Его целью было оценить, как изменение числа трансформерных блоков, числа голов внимания, размера блока и размера окна отражается на качестве распознавания именованных сущностей.

Эксперименты проводились на наборе данных Collection5 с использованием Multilingual BERT в качестве варианта формирования входных признаков. В качестве дополнительного слоя в анализируемых конфигурациях использовался DilatedConv – слой на основе нескольких параллельных одномерных свёрток, каждая из которых учитывает токены на разном расстоянии.

Для того, чтобы оценить влияние числа трансформерных блоков и числа голов внимания на качество распознавания именованных сущностей, была рассмотрена модель, реализующая в себе механизмы внимания PaTH и PaTH-FoX с дополнительным слоем DilatedConv.

На рисунке 3 приведены полученные метрики F1-меры для модели, реализующей в себе механизм внимания PaTH с дополнительным слоем DilatedConv.

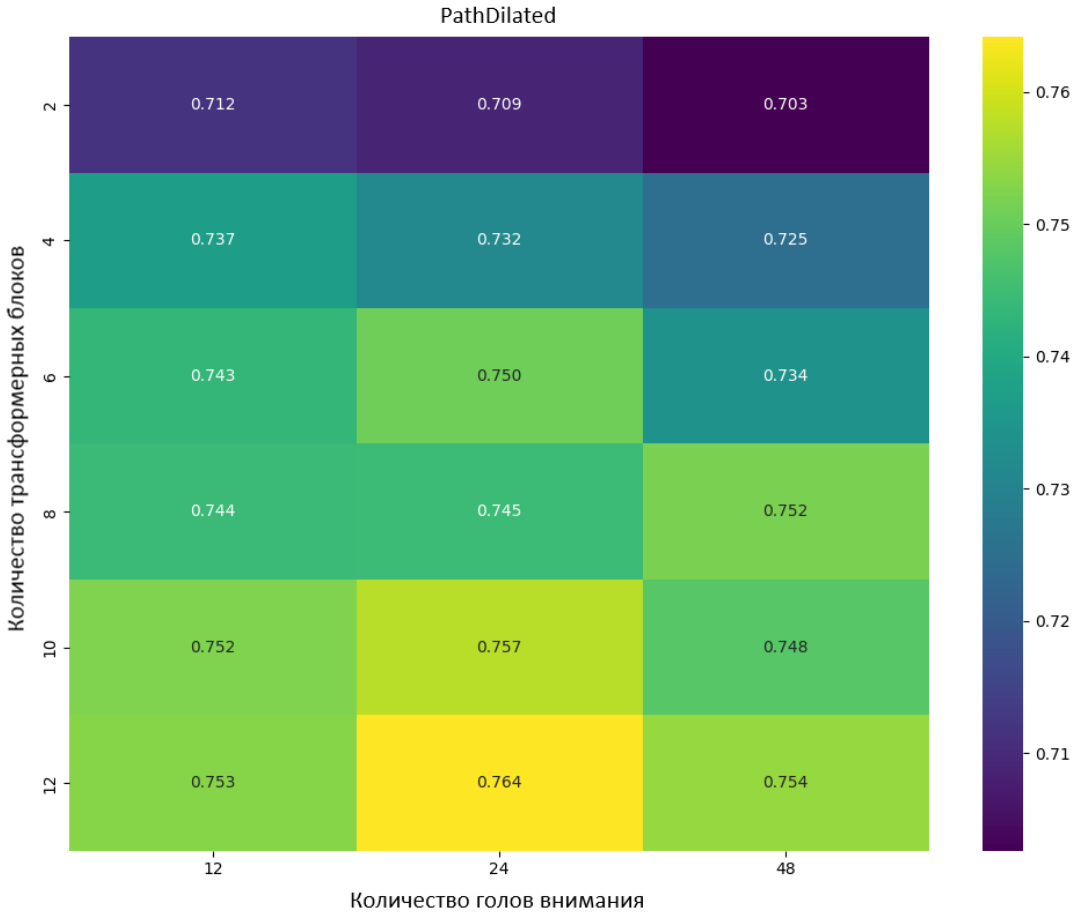

Рисунок 3 – Метрики F1-меры для модели с механизмом внимания PaTH со слоем DilatedConv

Полученные результаты показывают, что для механизма PaTH увеличение числа трансформерных блоков в среднем положительно влияет на качество распознавания именованных сущностей. Среднее значение F1-меры на проверяющей выборке возрастает от 0.708 при двух блоках до 0.757 при двенадцати блоках. Разница между минимальным и максимальным значением F1-меры среди всех рассмотренных конфигураций на проверяющей выборке составляет 0.061, что указывает на существенное влияние глубины модели.

При небольшом числе блоков лучшие результаты достигаются при меньшем числе голов, что может быть связано с большей размерностью отдельной головы внимания. Увеличение количества голов ведет к уменьшению размерности проекции, используемой одной головой, что уменьшает объем информации, доступный каждой отдельной голове, поэтому

зависимость качества распознавания от числа голов имеет нелинейный характер.

Близкие высокие результаты наблюдаются для конфигураций с 10 – 12 блоками, что позволяет говорить об области высокого качества.

На рисунке 4 приведены полученные метрики F1-меры для модели, реализующей в себе механизм внимания PaTH-FoX с дополнительным слоем DilatedConv.

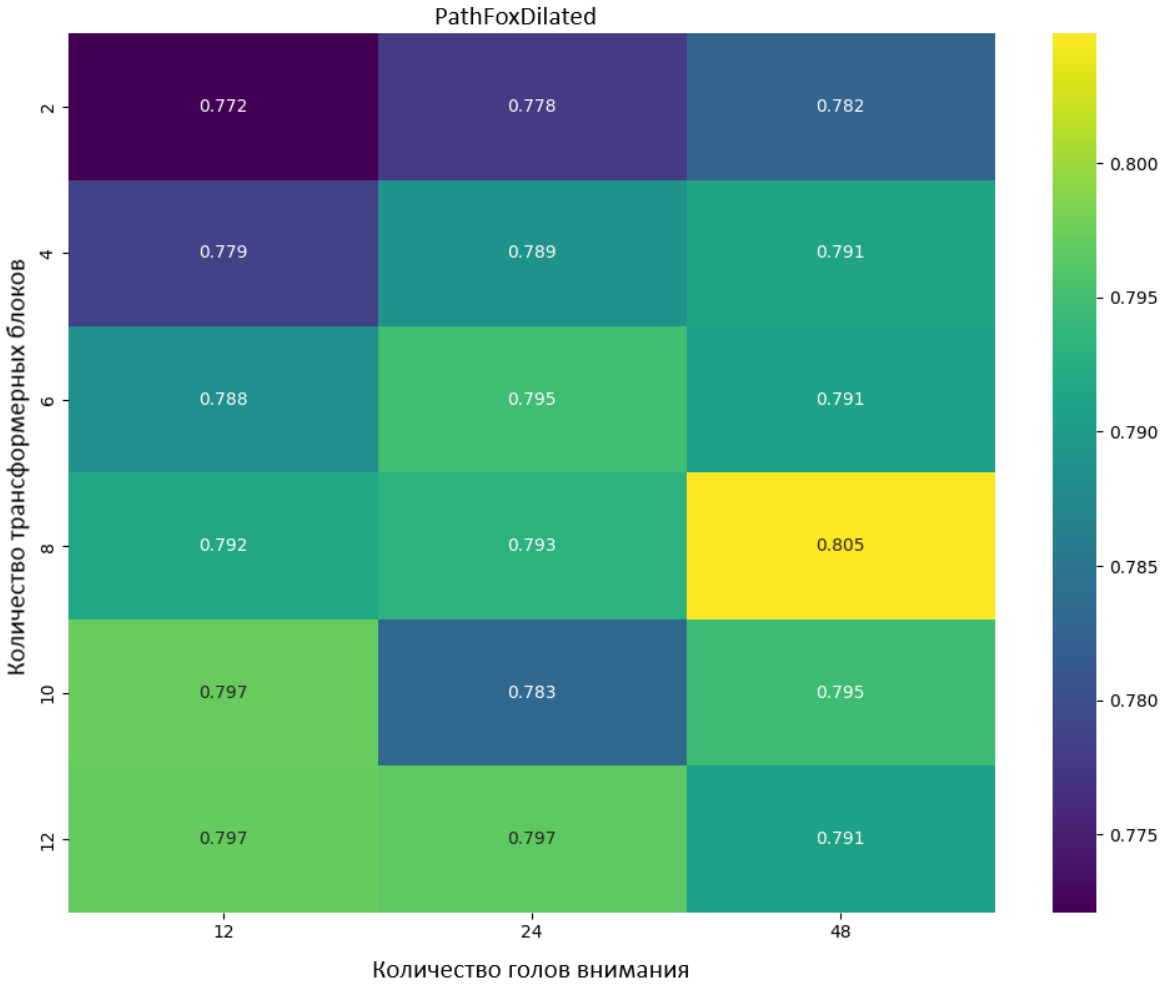

Рисунок 4 – Метрики F1-меры для модели с механизмом внимания PaTH-FoX со слоем DilatedConv

При использовании PaTH-FoX, за счёт механизма адаптивного забывания, модель быстрее выходит на область устойчивого качества. При использовании 12 голов F1-мера продолжает расти до диапазона 10 – 12

трансформерных блоков, при использовании 24 голов – до 8 трансформерных блоков, а при использовании 48 голов – до 4 трансформерных блоков.

Сопоставив результаты для PaTH и PaTH-FoX, можно прийти к выводу, что устойчивые значения качества наблюдаются для конфигураций, использующих 10 или 12 трансформерных блоков и 12, 24, 48 голов внимания.

Для дальнейших экспериментов была выбрана конфигурация, использующая 12 трансформерных блоков и 12 голов внимания.

Для определения размера блока для механизма блочного внимания и размера окна для механизма оконного внимания были проведены эксперименты с моделями, реализующими блочный и оконный механизмы внимания с дополнительным слоем DilatedConv.

В таблице 2 приведены результаты обучения модели, реализующей блочный механизм внимания.

Таблица 2 – Результаты обучения модели, реализующей блочный механизм внимания.

Размер блока	F1-мера (обучающая выборка)	F1-мера (проверяющая выборка)
16	0.761	0.609
32	0.766	0.619
64	0.761	0.633
128	0.770	0.628
256	0.780	0.632

Полученные результаты показывают, что увеличение размера блока с 16 до 64 приводит к росту F1-меры на проверяющей выборке, что указывает на то, что слишком малый блок ограничивает способность модели учитывать локальные зависимости между соседними токенами. Максимальное значение F1-меры на проверяющей выборке достигается при размере блока 64 и при дальнейшем его увеличении устойчивого прироста качества не наблюдается.

В таблице 3 приведены результаты обучения модели, реализующей оконный механизм внимания.

Таблица 3 – Результаты обучения модели, реализующей оконный механизм внимания.

Размер окна	F1-мера (обучающая выборка)	F1-мера (проверяющая выборка)
16	0.744	0.622
32	0.733	0.595
64	0.727	0.612
128	0.731	0.586
256	0.732	0.618

Полученные результаты для оконного механизма внимания показывают, что зависимость F1-меры от размера окна имеет нелинейный характер. Наилучшее значение F1-меры на проверяющей выборке достигается при размере окна 16 и составляет 0.622. При увеличении размера окна до 32 качество снижается до 0.595, затем при размере окна 64 частично восстанавливается до 0.612, при размере окна 128 снова уменьшается до 0.586, а при размере окна 256 возрастает до 0.618. Такое поведение показывает, что увеличение размера окна само по себе не гарантирует улучшения качества модели. При увеличении размера окна в область локального внимания попадает больше токенов. С одной стороны, это расширяет доступный контекст, но с другой – увеличивает количество шумовых или слабосвязанных токенов. Наблюдаемая «волнообразная» зависимость качества от размера окна может быть связана с балансом между полезным локальным контекстом и количеством нерелевантной информации внутри окна.

ЗАКЛЮЧЕНИЕ

В настоящей работе была сформирована архитектура трансформерной модели для распознавания именованных сущностей и определены диапазоны параметров, необходимые для ее настройки и применения в рамках задачи распознавания именованных сущностей в тексте. Рассматривались параметры, задающие глубину модели и способ распределения скрытого представления между головами внимания, а также параметры локальных механизмов внимания, определяющие размер области взаимодействия токенов: число трансформерных блоков, число голов внимания, размерность одной головы внимания, размер блока для блочного механизма внимания и размер окна для оконного механизма внимания. Проведенный экспериментальный анализ позволил оценить влияние параметров модели на качество распознавания именованных сущностей в тексте. Установлено, что изменение архитектурных параметров может по-разному отражаться на качестве модели: увеличение глубины, числа голов внимания или размера локальной области не является достаточным условием повышения качества. На основе полученных результатов была определена область значений параметров, обеспечивающая более высокое качество распознавания именованных сущностей в тексте в рамках рассматриваемой архитектуры.

ЛИТЕРАТУРА

1) Attention Is All You Need [Электронный ресурс]. // arXiv: [сайт]. – URL: https://arxiv.org/abs/1706.03762 (дата обращения: 03.10.2025).
2) Big Bird: Transformers for Longer Sequences [Электронный ресурс]. // arXiv: [сайт]. – URL: https://arxiv.org/abs/2007.14062 (дата обращения: 17.10.2025).
3) Longformer: The Long-Document Transformer [Электронный ресурс]. // arXiv: [сайт]. – URL: https://arxiv.org/abs/2004.05150 (дата обращения: 03.11.2025).
4) PaTH Attention: Position Encoding via Accumulating Householder Transformations [Электронный ресурс]. // arXiv: [сайт]. – URL: https://arxiv.org/abs/2505.16381 (дата обращения: 25.11.2025).
5) Low-Rank Bottleneck in Multi-head Attention Models [Электронный ресурс]. // arXiv: [сайт]. – URL: https://arxiv.org/abs/2002.07028 (дата обращения: 27.11.2025).
6) VEE-BERT: Accelerating BERT Inference for Named Entity Recognition via Vote Early Exiting [Электронный ресурс]. // OpenReview: [сайт]. – URL: https://openreview.net/pdf?id=3ffY9B-oiAm (дата обращения: 27.11.2025).
7) Collection5 [Электронный ресурс]. // Лаборатория информационных исследований: [сайт]. – URL: https://www.labinform.ru/pub/named_entities/collection5.zip (дата обращения: 04.12.2025).