ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2025, № 1, С. 149–162

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

УДК 004.8

МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ КОРЕФЕРЕНЦИИ И ПОИСКА ИМЕННЫХ ГРУПП В ЕСТЕСТВЕННЫХ ЯЗЫКАХ

© 2025 г. А. А. Козлова[a], *, И. Д. Кудинов[a], **, Д. В. Лемтюжникова[a], ***

[a]Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия

**e-mail: sankamoro@mail.ru*

***e-mail: ilja@kdsli.ru*

****e-mail: darabbt@gmail.com*

Поступила в редакцию 13.11.2024 г.
После доработки 01.12.2024 г.
Принята к публикации 13.01.2025 г.

Кореференция – это задача области обработки естественных языков, направленная на связывание слов и фраз в тексте, которые указывают на один и тот же объект реального мира. Она применима при суммаризация текста, ответах на вопросы, информационном поиске и диалоговых системах. Приводится разбор существующих методов решения задачи кореференции, а также предлагается способ, основанный на применении двухэтапной модели машинного обучения. Языковая модель преобразует токены текста в векторные представления. Далее для каждой пары токенов на основе их векторных представлений вычисляется оценка вероятности нахождения этих токенов либо в одной именной группе, либо в двух кореферентных именных группах. Таким образом, метод одновременно производит поиск именных групп и предсказывает кореферентную связь между ними.

Ключевые слова: кореференция, обработка естественного языка, машинное обучение, языковые модели

DOI: 10.31857/S0002338825010122 **EDN:** AIJIND

METHODS OF SOLVING THE PROBLEM OF COREFERENCE AND SEARCHING FOR NOUN PHRASES IN NATURAL LANGUAGES

A. A. Kozlova[a], *, I. D. Kudinov[a], **, D. V. Lemtyuzhnikova[a], ***

[a]V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Russia

**e-mail: sankamoro@mail.ru*

***e-mail: ilja@kdsli.ru*

****e-mail: darabbt@gmail.com*

Coreference is a task in the field of natural language processing aimed at linking words and phrases in a text that point to the same extra-linguistic object or referent. It is applicable in text summarization, question answering, information retrieval and dialog systems. In this paper, existing methods for solving the co-referencing problem are dissected and a method based on the application of a two-stage machine learning model is proposed. The language model converts text tokens into vector representations. Then, for each pair of tokens, based on their vector representations, an estimate of the probability of finding these tokens either in one name group or in two coreference name groups is computed. Thus, the method simultaneously searches for name groups and predicts the coreference relation between them.

Keywords: coreference, natural language processing, machine learning, language models

Введение. Все ускоряющийся рост объема хранимой человечеством информации, записанной текстом, уже не позволяет работать с нетрадиционными методами, в которых для работы с текстом неизбежно нужны умеющие читать и писать люди. В наши дни необходимо уметь извлекать информацию и менять ее в текстовых документах разного стиля: от промышленной документации до произведений художественной литературы, не прибегая к помощи человека.

Естественные языки возникли как средство общения между людьми и никогда не предназначались для автоматической обработки. Они даже не представляют из себя строгой системы правил, которую можно было бы превратить в математическую модель, они имеют скорее статистическую природу [1]. Один и тот же текст, предложение или слово могут нести разный смысл, значение или эмоциональную окраску для разных людей, культур и времен. Все это делает задачи, связанные с семантикой, невероятно трудными, но поскольку актуальность полного или частичного решения этих задач с развитием человечества только растет, предлагаются различные методы их решения.

Одним из направлений математической лингвистики является работа по формализации модели грамматичных текстов на естественных языках. Большой популярностью пользуется теория универсальной грамматики, родоначальником которой считается американский лингвист Н. Хомский [2]. Универсальная грамматика имеет общие черты, присущие всем известным естественным языкам. Так, тексты на всех языках могут быть выделены в абстрактное синтаксическое дерево по принципу синтаксического подчинения одного слова другому. Каждое слово в этом дереве обладает собственной синтаксической ролью, самые универсальные из которых – подлежащее, сказуемое и прямое дополнение. Универсальная грамматика также отмечает общие для всех языков понятия падежей, времени, лица, числа, рода, наклонения, способа словообразования при помощи аффиксов. Такая теория позволяет дать ответ для текста, где для каждого слова известны его грамматические признаки, будет ли он грамматичным или нет.

Задача кореференции возникла как один из способов определения смысла текста. В отличие от задач суммаризации или ответов на вопросы по тексту, решение задачи кореференции дает возможность точечно получить информацию из текста по какому-то объекту. В представленной работе предлагаются методы ее решения для русскоязычных текстов с использованием систем правил и машинного обучения.

Современные языковые модели решают задачу кореференции неявным образом, учитывая при вычислении внутреннего представления каждого слова контекст величиной в 4–10 тыс. слов, стоящих перед текущим. Однако задача кореференции не теряет актуальности по причине того, что она является одной из самых легко формализуемых задач среди тех, которые оперируют “смыслом” слов. Это позволяет изучать свойства языковых моделей на примере решения задачи кореференции. Наши дальнейшие работы посвящены этой теме.

1. Задача кореференции. 1.1. Именная группа. Введем основные понятия из области обработки естественного языка, необходимые для решения задачи кореференции.

Определение 1. Именная группа (noun phrase) – словосочетание в тексте, ссылающееся на объект реального мира.

Определение 2. Референт именной группы – объект реального мира, на который ссылается эта именная группа.

Именная группа является основным объектом в семантике текста, любой текст – описанием действий или состояний объектов, выраженных в нем именными группами. На рис. 1 представлено три предложения с выделенными в них именными группами.

Простой способ определить именную группу вручную – заменить ее полностью на местоимение. Например, предложения с рис. 1 при подстановке в них местоимений вместо именных групп преобразуются в предложения на рис. 2 соответственно.

Согласно подходу грамматики зависимостей, которой следует теория универсальной грамматики, множество слов и знаков препинания в тексте образуют ориентированное синтаксическое дерево. Ребра дерева всегда направлены в сторону от корня и выражают синтаксическое подчинение одного слова другому.

Определение 3. Синтаксически подчиненным называют слово x, которое связано с другим словом y так, что x задает вопрос y. Синтаксические связи классифицируются по синтаксическим ролям: подлежащее, дополнение, определение, обстоятельство и др.

В качестве системы синтаксических ролей часто используется проект UD (universal dependencies) [3], содержащий в том числе универсальную систему синтаксических ролей, которая подходит для более чем 80 естественных языков. Корнем дерева всегда является сказуемое. Пример синтаксического дерева приведен на рис. 3. Именные группы выделены

Предложение содержит именную **группу**

На **доске** были написаны разные **слова**

Лежащее рядом **яблоко** казалось **свежим** и **влажным**

Рис. 1. Предложения с выделенными в них именными группами.

Оно содержит **это**

На **ней** были написаны **они** (**Они** были написаны на **ней**)

Оно казалось **таким** и **таким**

Рис. 2. Предложения с рис. 1, все именные группы в которых заменены на местоимения.

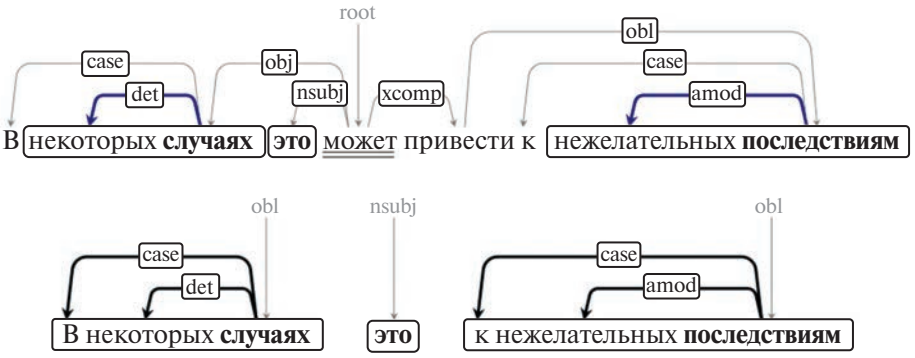

Рис. 3. Синтаксическое дерево предложения.

прямоугольниками, слова-вершины именных групп и стрелки синтаксических связей от вершин – жирным шрифтом. Каждая синтаксическая связь снабжена меткой соответствующей синтаксической роли, согласно UD [3].

В некоторых случаях вершина дерева может не соотноситься ни с одним словом или частью речи в тексте. Например, когда подлежащее или сказуемое не заданы явно. Слова-члены именной группы также входят в синтаксическое дерево. Именная группа определяется словом-вершиной.

Определение 4. Вершина именной группы (head of noun phrase) – это существительное или местоимение, которому подчинены все остальные слова именной группы. На рис. 1 вершины соответствующих именных групп выделены жирным шрифтом.

Каждая именная группа имеет вершину (head of noun phrase) – это существительное или местоимение, которому подчинены все остальные слова именной группы. Вершина определяет именную группу, а все остальные слова группы, выступающие чаще всего прилагательными или причастиями, лишь уточняют смысл вершины. Некоторые или все слова именной группы, не являющиеся вершиной, могут быть убраны без потери грамматичности текста. Без вершины именная группа не может ссылаться на референт и теряет свой смысл – прилагательные сами по себе ничего не значат.

Определение 5. Субстантивация – это лингвистическое явление, при котором слова других частей речи (например, прилагательные, глаголы, числительные) приобретают функции существительных и могут выступать в роли вершины именной группы. Примерами субстантивации служат:

а) “больной поправился” – прилагательное;
б) “командующий отдал приказ” – причастие;
в) “подали на второе” – порядковое числительное;
г) “двое на качелях” – собирательное числительное;
д) “наше завтра” – наречие;

Рис. 4. Случаи вложенных именных групп.

Рис. 5. Именная группа, состоящая из однородных членов.

е) “услышал мерное тик-так” – звукоподражательные слова;

ж) “для посмеяться” – глагол-инфинитив.

Из этого следует, что опираться на определение принадлежности слов к той или иной части речи для обозначения именных групп невозможно. Все слова текста, относящиеся к самостоятельным частям речи, кроме глаголов, наречий и деепричастий, так или иначе входят в какую-либо именную группу: существительные и местоимения создают их, а остальные (прилагательные, причастия и слова других частей речи) описывают, уточняют их.

Именные группы обладают следующими свойствами:

1. Именная группа не может содержать слова, находящиеся в разных предложениях, потому что между словами в разных предложениях не может существовать синтаксической связи.

2. Именные группы представляют собой непрерывную последовательность слов в предложении.

3. Именные группы могут быть вложены друг в друга, если вершина одной группы подчинена слову из другой именной группы. При этом каждая из именных групп ссылается на разные референты. Например, для второго предложения на рис. 4 выделяются три входящие друг в друга именные группы:

а) “выпущенная из моего лука стрела”;

б) “моего лука”;

в) “моего».

На рис. 4 именные группы выделены прямоугольниками, вершины соответствующих групп – жирным шрифтом, стрелки между двумя вершинами двух именных групп – черным цветом. Вершины именных групп играют роль подлежащего или дополнения. Если синтаксические роли каждого слова текста известны, таким простым алгоритмом могут быть однозначно найдены все вершины именных групп. По определению каждое существительное и местоимение создает свою именную группу.

Ряд однородных членов предложения также является цельной именной группой. Пример такого случая продемонстрирован на рис. 5, на котором отдельные именные группы обведены в прямоугольники.

1.2. Кореференция.

Определение 6. Кореферентные именные группы – это именные группы, которые ссылаются на один и тот же референт.

Определение 7. Задача кореференции заключается в разбиении множества именных групп на непересекающиеся классы, в которых именные группы будут кореферентными.

Демонстрирующие трудности этой задачи примеры изображены на рис. 6. В первом примере две именные группы “Маша” и “кашу” в первом предложении являются кандидатами на кореферентность с именной группой ”Она” во втором предложении. Согласно грамматическим признаками, “ей” могут быть как Маша, так и каша, однако продолжение второго предложения дает характеристику “вкусная” референту “ей”, что больше, но не однозначно соответствует каше. Второй пример демонстрирует случай кореференции между именными группами, не являющимися местоимениями. На рис. 6 именные группы выделены прямоугольниками. Стрелками обозначены кореферентные связи.

Теоретически любые две именные группы в тексте могут оказаться кореферентными вне зависимости от части речи или грамматических признаков вершин групп. Иногда кореферентные именные группы могут быть не согласованы по грамматическому роду или числу: “Редколлегия отклонила мою заявку. Они написали...”, “Сегодня я был у врача. Она сказала...”. Поэтому наличие кореференции между двумя группами в общем случае может быть определено исключительно по их семантике – по статистике применения каждой из двух вер-

Рис. 6. Примеры кореференции.

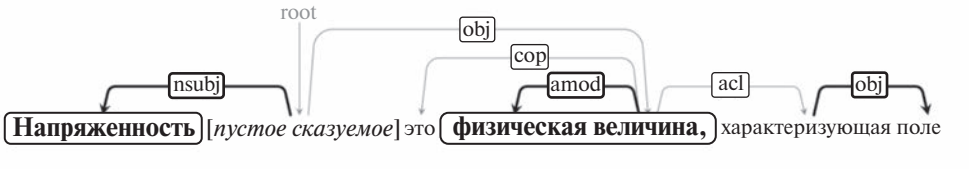

Рис. 7. Синтаксическое дерево предложения, в котором сказуемое не выражено явно.

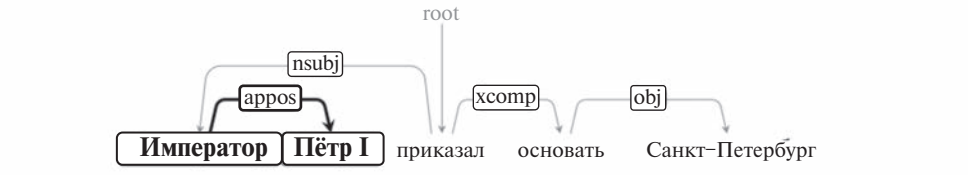

Рис. 8. Синтаксическое дерево предложения с подлежащим и его модификатором.

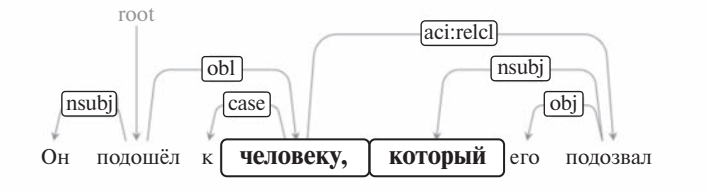

Рис. 9. Синтаксическое дерево сложного предложения.

шин. Однако для русского языка можно выделить несколько общих правил, которые могут либо полностью исключить, либо однозначно установить возможность кореференции между двумя группами:

1. Местоимения могут быть кореферентными только с подлежащими или прямыми дополнениями, если они стоят в разных предложениях.

2. Если сказуемое предложения отсутствует в явном виде, то оно подразумевает эквивалентность подлежащего и прямого дополнения предложений – вместо "пустого сказуемого" могут быть подставлены глаголы "является", "равен" и т.п., как продемонстрировано на рис. 7. На нем две выделенные именные группы кореферентны. Каждая синтаксическая связь снабжена меткой, соответствующей синтаксической роли, согласно UD [2]: роль nsubj – подлежащее, роль obj – прямое дополнение.

3. Именная группа и ее модификатор кореферентны. Пример такого случая приведен на рис. 8. На нем две выделенные именные группы кореферентны, а каждая синтаксическая связь снабжена меткой, соответствующей синтаксической роли, согласно UD [2]: роль appos – модификатор.

4. В сложноподчиненном предложении именная группа кореферентна союзному слову, стоящему прямо после него. Пример такого случая приведен на рис. 9. Две выделенные именные группы кореферентны, каждая синтаксическая связь снабжена меткой, соответствующей синтаксической роли, согласно UD [2]: роль acl:relcl задает связь между корнем подчиненной клаузой "который его подвел" и словом "человеку", которому клауза подчиняется.

5. Большое расстояние между именными группами практически всегда исключает их непосредственную кореферентность.

Необходимо отметить, что данные правила верны только для грамматичного текста.

Определение 8. Кореферентная цепочка – последовательность кореферентных именных групп в тексте.

Поиск кореферентных цепочек в тексте является конечной целью задачи кореференции. Многие модели рассматривают задачу кореференции как задачу оптимальной разметки множества именных групп на набор непересекающихся подмножеств, соответствующих той или иной цепочке-референту [4].

2. Обзор существующих методов решения задачи кореференции**.** Решение задач обработки естественного языка часто ориентировано на тексты конкретного стиля – научно-технические или публицистические, написанные грамотно и без ошибок, поскольку именно с ними чаще всего работают на практике. В данной работе задача кореференции рассматривается на примере таких корректных научно-технических текстов.

В обзорной статье [4] представлен современный взгляд на подходы к решению задачи кореференции, основные термины и методы, а также достигнутые результаты. Авторы приводят классификацию кореференции в английском языке, анализируют влияние грамматических признаков на кореференцию, правила, представленные на рис. 7–9, и методы оценки и решения задачи.

2.1. Методы, основанные на системах правил. Этот метод использует систему фиксированных правил для выявления кореферентных связей между именными группами. Сначала определяются грамматические признаки и синтаксические связи слов, а затем на основе этих данных оценивается наличие кореференции для каждой пары именных групп.

Часто такие модели применяют так называемые семантические классы слов, объединяющие слова по темам, к которым они относятся. Например, слова “пирог”, “запеканка”, “мясо” могут относиться к нескольким классам: от класса “еда” до класса “физический объект” в зависимости от точности системы классов. Необходимость в ручном составлении словарей семантических классов со временем свела на нет практику их использования.

В работе [5] описана модель, применяемая к английским текстам: она выделяет именные группы *i* и *j* и анализирует их на кореференцию по ряду признаков:

а) расстояние между *i* и *j* в предложениях;

б) является ли *i, j* местоимением;

в) совпадают ли *i* и *j*, если убрать у них артикли и указательные местоимения; г) начинается ли *i, j* с указательного местоимения “*the*” или нет;

д) согласованы ли *i* и *j* в числе, роде;

е) равны ли семантические классы *i* и *j*;

ж) выступают ли *i* и *j* именами собственными;

и) является ли *i* записанным по-другому *j* или наоборот; например, *j* может быть аббревиатурой *i* или отличаться от него дополнительной приставкой (преноминалом) как, например, в случае фамилий или организаций: “*Mr. Simpson*” и “*Bent Simpson*”;

к) будет ли *j* приложением *i*; в английском языке приложения отделены от остального предложения запятыми и находятся по соседству с выражением, к которому они относятся: “*My sister, Alice Smith, likes jelly beans*” и “*Alice Smith, my sister, likes jelly beans*”.

Подобные модели требуют тщательной настройки списка признаков и их весов, поскольку они не учитывают контекст и семантику. Аналогичный подход для русского языка представлен в работе [6]. В модели для каждой именной группы подбираются предшествующие кандидаты на кореференцию с последующим выбором наиболее вероятной связи. Включение семантической информации о контексте заметно улучшает результаты.

В работе [7] предлагается модель, решающая задачу кореференции путем анализа синтаксического дерева набором правил, подобных тем, что изображены на рис. 7–9.

2.2. Методы, основанные на машинном обучении. В идеальном решении для задачи кореференции учитывается практика языка, на которой базируется использование слов и выражений, однако ручные системы правил, как правило, игнорируют этот аспект, что делает их решения неточными. В отличие от них, модели машинного обучения могут извлекать скрытые зависимости, обучаясь на больших наборах данных. Они показали заметные успехи в решении этой задачи, так как выявляют связи, которые часто упускаются при ручной разработке списков признаков и правил.

Модель из работы [8] решает задачу для произвольного языка с помощью дополнительной контекстной информации. Весь текст делится на *n*-граммы – последовательности из *n* слов, и между ними определяются возможные кореферентные связи. Модель с долговременной и краткосрочной памятью (long short-term memory (LSTM)) анализирует последовательности

слов, “запоминая” контекст, что позволяет учитывать предыдущие слова для лучшего прогноза кореференции [9, 10]. Чтобы уменьшить вычислительные затраты, отбрасываются n-граммы, не удовлетворяющие минимальному значению оценки кореференции. Однако такие подходы первого порядка, когда рассматриваются только пары именных групп, могут приводить к локальным согласованиям, но нарушать глобальную совместимость. В работе [11] это решается с помощью последовательного пересмотра прогнозов на следующих итерациях, что позволяет корректировать несовместимые группы.

Работа [12] улучшает точность за счет представления текста как матрицы вероятностей кореференции между предложениями. Три типа представлений слов – независимое, контекстуальное и символьное – объединяются, и их контекстуализация через LSTM помогает лучше учитывать общую семантику текста. Векторные представления слов проходят через LSTM-сеть, позволяя учитывать расположение предложений и их значимость, а матрица кореференций составляется с использованием механизма из публикации [13]. В работе [14] представлен алгоритм с “неконтролируемой” долговременной памятью для кореферентных связей, особенно полезный, когда требуется объемная информация об объектах. Для русского языка подходы к задаче кореференции исследовались в работе [15], где тестировались уже существующие решения, разработанные для других языков.

На конференции по компьютерной лингвистике “Диалог” в 2014 г. также проходило соревнование по решению задач анафоры и кореференции на русскоязычных текстах [16]. Метод из работы [17], основанный на технологии ABBYY Compreno для синтаксического анализа из [18], смог правильно решить задачу кореференции в 64% случаев. Модель из работы [19] комбинировала нейросетевые методы с правилами, что обеспечило решение в 51–59%, однако результаты сильно зависели от типа текста. Метод из работы [20] был направлен на анафору местоимений, где перебирались возможные антецеденты с оценкой их релевантности через нейросеть, что позволило достичь решения в 75% случаев. Модель из работы [21] строила кореферентные пары на базе выделения существительных и соседних слов по системе правил, обеспечивая точность 52%. В последующих соревнованиях по кореференции точность оставалась в пределах 50–75%, что отражает сложность задачи для русского языка [22].

2.3. Метрики. Метрики задачи кореференции делятся на три типа:

а) основанные на связях (link based): здесь оценки рассматривают кореферентные цепочки как множество связей между именными группами, но игнорируют референты с одним упоминанием, что иногда завышает результат;

б) базирующиеся на упоминаниях (mention based): оценки рассматривают кореферентные цепочки как множество именных групп;

в) основанные на поиске оптимального отображения (mapping-based) из найденного ответа на настоящий.

Данная классификация позволяет комплексно учитывать разные аспекты работы модели. Ее схематичное изображение приведено на рис. 10.

Метрики, основанные на связях (link-based), такие как моделирование неопределенности в корреляции (modeling uncertainty in correlation (MUC)) [23], рассматривают кореферентные цепочки как совокупность связей между именными группами. MUC определяет минимальное число изменений, требуемых для приведения к истинному разбиению, но игнорирует референты с одним упоминанием, что иногда завышает результат.

Чтобы преодолеть ограничения MUC, была предложена метрика B^3 [24], базирующаяся на упоминаниях, которая анализирует каждую кореферентную цепочку отдельно, что позволяет точнее учитывать именные группы в тексте, хотя они могут некорректно учитывать одно упоминание в нескольких цепочках.

Для более точного анализа метрики на основе отображений (mapping-based), такие как CEAF [25], строят оптимальное отображение найденных пар на истинные значения, а также оптимальное отображение между полученными цепочками и истинными. CEAF оценивает точность и полноту на базе совпадения цепочек, но игнорирует ошибки в несвязанных группах.

Для комбинированной оценки применяется CoNLL-метрика [26], которая усредняет F-меру, используемую в задачах классификации, MUC, B^3 и CEAF, предлагая сбалансированный взгляд на результативность алгоритма, но сглаживает уникальные свойства каждой метрики. LEA учитывает значимость каждого референта и различает важные и менее значимые ошибки, что делает ее более точной и детализированной по сравнению с другими методами [27].

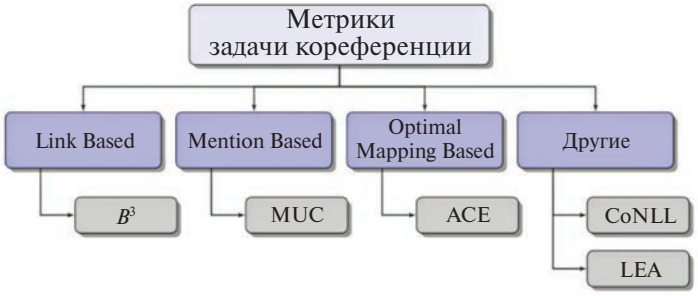

Рис. 10. Классификация метрик задачи кореференции, согласно [4].

3. Применение языковой модели к решению задачи кореференции**.** 3.1. Постановка задачи. Естественный язык позволяет описывать события, опыт и действия, что подразумевает наличие объектов контекста $s \in S$ и реакции $o \in O$. Эти объекты рассматриваются как тексты.

Определение 9. Текст является последовательностью токенов из множества V (словарь), где размер словаря записывается как $|V|$, элемент V – токен w, который может быть отдельным символом, группой символов или целым словом. Конечная последовательность токенов обозначается как $t = (w_1^n)$.

Множество всех возможных текстов T представляет языковую модель с распределением вероятности $p(t)$. Контекст $s \in T$ описывается условным распределением $p_{.|s}(t)$ над множеством текстов T_s, которое может включать тексты, начинающиеся или заканчивающиеся на s, или содержащие его как подстроку. В соответствии с новой постановкой дистрибутивная гипотеза пересматривается так: соответствующее тексту $s \in T$ распределение вероятности $p_{.|s}$ над множеством текстов из T_s. Основная цель моделей – получить условное распределение $p_{.|s}$ текстов по известному контексту s.

3.2. Токенизация. Перед обработкой любая языковая модель требует токенизации – преобразования текста в последовательность токенов. Наиболее распространенный подход к токенизации – токенизация по подсловам (subword tokenization), которая сочетает преимущества пословной и посимвольной токенизации.

Алгоритм парного кодирования (byte pair encoding (BPE)) используется для создания словаря подслов путем комбинирования часто встречающихся последовательностей символов в токены, что позволяет эффективно кодировать корни и аффиксы слов. В словарь токенов V включаются специальные токены, такие как:

а) начало и конец предложения;

б) начало и конец абзаца’

в) “неизвестный токен” для отсутствующих последовательностей.

3.3. Векторизация. В работе [28] был предложен метод *Word2Vec*, породивший направление так называемых векторных представлений. Метод использует алгоритм skig-gram для построения условного распределения текста вокруг токена w_s. Контекстом служат n соседних токенов, как изображено на рис. 11, при $n = 2$, а искомое распределение $p(w_{i+1}, w_{i-1}|w_i)$ задается как произведение $p(w|w_i)$.

Обучение модели происходит через максимизацию правдоподобия, представленную произведением для каждого токена w_i в текстах длиной T. Переход к задаче минимизации выполняется через отрицательный логарифм. Определение вероятности $p(w_{i+j}|w_i)$ сводится к метрике между распределениями $p(x)$ и $p(x)$, отражающей семантическую близость токенов. Дистрибутивная семантика связывает каждому токену числовой вектор, кодирующий его семантическое значение. Если преобразование f сохраняет структуру расстояний между векторами, то сходство $x_1, x_2 \in X$ можно оценить по сходству $f(x_1), f(x_2) \in Y$. Это преобразование называется вложением (embedding). Векторное представление токена w соответствует числовому вектору v_w.

3.4. Механизм внимания. Обработка последовательностей, элементы которых зависят от предыдущих, долгое время оставалась проблемой для алгоритмов машинного обучения [29]. Рекуррентные модели, такие, как LSTM [30], страдают от быстрого “забывания” информации и ограничений в распараллеливании, что затрудняет обработку длинных текстов.

В качестве решения был предложен механизм внимания (attention) [31, 32]. Он позволяет моделям, принимающим и возвращающим последовательности токенов, обращать вни-

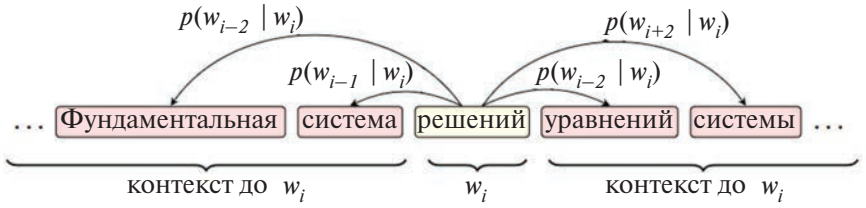

Рис. 11. Определение условного распределение текста вокруг токена w_i = “решений”.

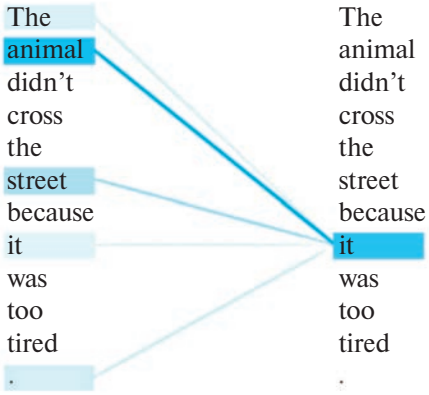

Рис. 12. Пример работы механизма внимания.

мание на определенные элементы входной последовательности. Механизм самовнимания использует одну и ту же последовательность в качестве входной и выходной. Пример работы механизма представлен на рис. 12: для токена “*it*” строится распределение, где более яркий цвет указывает на элементы, на которые стоит обращать внимание, такие как подлежащее “The animal” и соответствующее сказуемое.

3.5. Выводы. Языковые модели оперируют распределениями вероятностей над последовательностями токенов: минимальных элементов текста. Условная вероятность распределения контекста вокруг токена может интерпретироваться как семантическое значение, смысл данного токена.

Операции сравнения векторных представлений приближенно коррелируют с операциями сравнения соответствующих условных вероятностей, а значит, в совокупности: смыслов, семантических признаков, синтаксических ролей и положения в тексте соответствующих последовательностей токенов. Алгоритм решения задачи кореференции требует как можно большей информации о двух именных группах, для которых требуется определение наличия или отсутствия кореференции. Это делает сегодня привлечение языковых моделей для эффективного решения задачи кореференции необходимым.

4. Данные для обучения. В целях обучения моделей для решения задачи кореференции могут использоваться датасеты (таблица). Корпуса для задач кореференции охватывают различные языки и жанры, что помогает моделям адаптироваться к языковым и культурным особенностям текстов. Эти данные представлены на русском и английском языках и включают новости, научные статьи, литературу и диалоги, улучшая способность моделей обобщать информацию. Русскоязычные корпуса RuCoCo и Ru-eval-2019 предлагают хорошее покрытие для задач кореференции, но их объем уступает англоязычным, что может ограничивать эффективность обучения моделей для русского языка. Доступность данных также играет важную роль. RuCor и RuCoCo доступны бесплатно, что делает их полезными для исследователей. Некоторые корпуса, такие как SemEval и Ru-eval-2019, ориентированы на соревнования и предоставляют наборы данных для сравнения и тестирования моделей.

Данные корпуса формируют значительную базу для обучения моделей кореференции, однако разница в объеме и доступе подчеркивает необходимость создания более обширных открытых ресурсов, особенно для языков с меньшими объемами данных, таких как русский.

5. Решение задачи кореференции. 5.1. Модель. Современные универсальные языковые модели демонстрируют способность решения широкого класса задач обработки текста без

Таблица. Датасеты для обучения

Корпус	Состав	Структура	Применение	Доступ
RuCoCo	Русскоязычные новостные тексты; 3075 текстов, ~150 000 кореферентных цепочек	JSON-файлы: тексты с аннотацией кореферентных цепочек, вложенные и простые связи между именными группами	Обучение моделей кореференции на русском языке, задачи NLP	GitHub (https://github.com/dialogue-evaluation/RuCoCo-2023)
Ru-eval-2019	Русскоязычные тексты разных жанров: новости, научные тексты, диалоги; ~3000 текстов, ~100 000 кореферентных цепочек	Аннотированные цепочки кореференции, представляющие связи между упоминаниями одного референта	Тестирование моделей кореференции в задачах NLP и соревнованиях	Доступен по запросу или для участников соревнований Ru-eval (https://www.dialog-21.ru/media/4689/budnikovzverevamaximova2019evaluatinganaphoracoreferenceresolution.pdf)
RuCor	Русскоязычные тексты из разных жанров: новости, научные тексты, литература; ~ 2000 текстов	Тексты и аннотации кореферентных цепочек; связывают именные группы и местоимения с одним референтом	Для задач кореференции, тестирования и обучения моделей NLP на русском языке	RuCoref (http://rucoref.maimbava.net/)
AnCor	Русскоязычные тексты разных жанров; ~ 2500 текстов, ~ 75 000 кореферентных цепочек	Кореферентные цепочки, связывающие слова и фразы в тексте, которые относятся к одному объекту	Для обучения и тестирования моделей NLP, задач кореференции	Доступ по запросу (Russian National Corpus) (https://ruscorpora.ru/)
GUM	Многоязычные тексты: академические статьи, новости, литература, диалоги; ~ 800 текстов	Многоуровневая аннотация (кореференция, синтаксис, семантика); выделение именных групп и местоимений	Для обучения и тестирования моделей синтаксического и семантического анализа, выявления связей в тексте	GUM Corpus (https://paperswithcode.com/dataset/gum)
SemEval	Тексты различных жанров, включая новости и литературу; ~ 1000 текстов	Кореферентные цепочки с тренировочными и тестовыми наборами	Для оценки и тестирования моделей NLP, разработки алгоритмов кореференции	SemEval (https://www.researchgate.net/publication/262389039_SemEval-2010_Task_1_Coreference_Resolution_in_Multiple_Languages)

какой-либо предварительной обработки входного текста, кроме токенизации. Рассмотренные ранее методы решения кореференции опирались на предварительное решение задачи поиска именных групп. В свою очередь, демонстрируемая в данной работе модель пытается решить эти две задачи одновременно.

Пусть w_1^n – входная последовательность токенов. План работы модели заключается в вычислении для каждого токена $w_i \in w_1^n$ набора оценок $s_{wi}(w) \in [0,1]$ для всех $w \in w_1^n$. Значение $s_{wi}(w_j)$ отражает вероятность того факта, что токены w_i и w_j входят в кореферентные группы. Как показано на рис. 13, в случае пары кореферентных именных групп, выраженных во входном тексте последовательностями токенов x_1^m и y_1^k, для каждого токена $w_i \in x_1^m \cup y_1^k$ значение $s_{w_i}(w_j)$ будет относительно высоким для токенов $w_j \in x_1^m \cup y_1^k$ и относительно низким для остальных токенов.

Для вычисления оценок $s_{wi}(w_j)$ в данной работе используется механизм внимания. Механизм внимания производит распределение вероятности $p_{wi}(w)$ над всей последовательностью w_1^n. Значение $p_{wi}(w_j)$ может быть получено из оценок s_{wi} следующим образом:

$$\frac{s_{w_i}\left(w_j\right)}{\sum_{k=1}^{n} s_{w_i}\left(w_k\right)} = \frac{s_{w_i}\left(w_j\right)}{C} = p_{w_i}\left(w_j\right) \Rightarrow s_{w_i}\left(w_j\right) = p_{w_i}\left(w_j\right) C, \quad C \in \mathbb{Z}.$$

В качестве константы C применяется размер входной последовательности $|w_1^n|$. Если $C < 1$ и $p_{w_i}(w_j) \in [0,1]$, то и $s_{w_i}(w_j) \in [0,1]$.

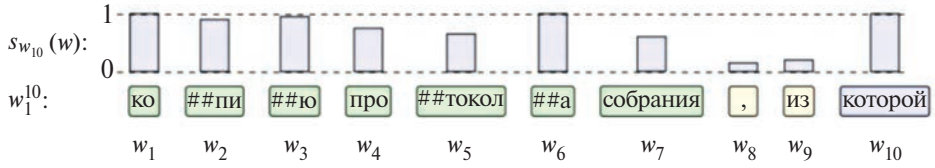

Рис. 13. Расчет оценок для токенов текста.

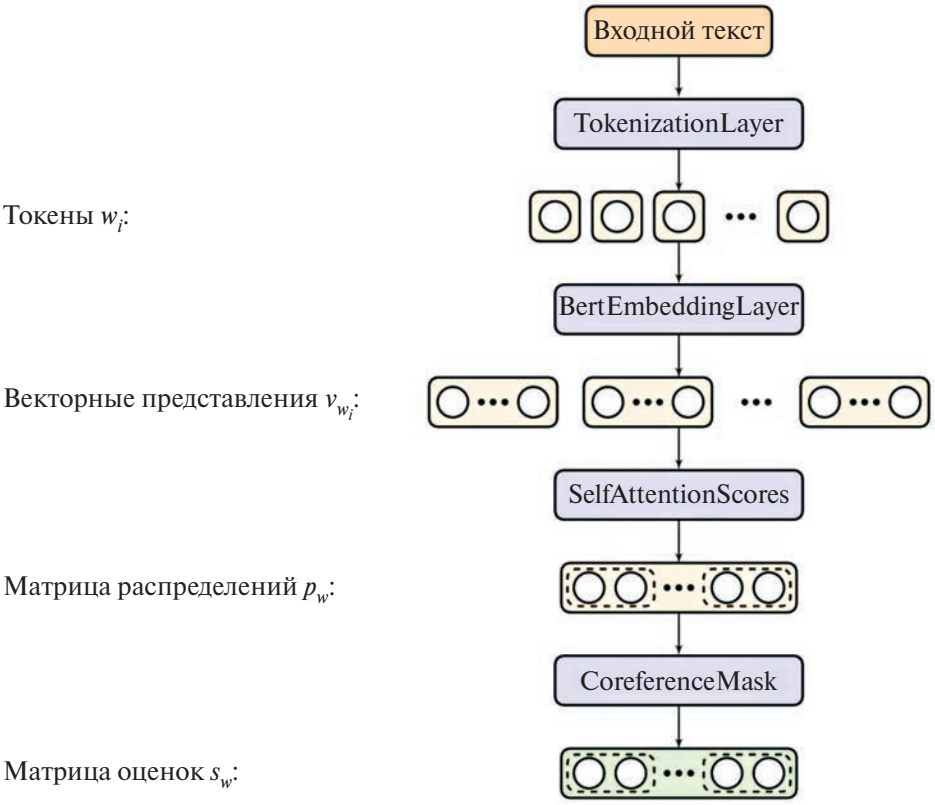

Рис. 14. Схема итоговой модели.

Внутренние представления токенов, используемые при вычислении $s_w\mathrm{i}(w_j)$, генерируются при помощи произвольной сторонней рабочей языковой модели. Предполагается, что генерируемые языковыми моделями общего назначения представления токенов содержат достаточно информации для определения кореференции описанным методом.

Поскольку механизм внимания вычисляет распределение p_w для каждого w независимо, в общем случае $s_{w_i}(w_j) \neq s_{w_j}(w_i)$ для любой пары токенов w_i и w_j. Именные группы могут быть вложены друг в друга, входя при этом в разные кореферентные цепочки. Таким образом, если w_i входит во вложенную именную группу, а w_j входит в другую именную группу и при этом они кореферентны, то значения оценок $s_{w_i}(w)$ и $s_{w_j}(w)$ могут различаться для всех w.

Принять окончательное решение о наличии кореферентной связи между w_i и w_j можно путем сравнения $s_{w_i}(w_j)$ с некоторой константой $l \in [0, 1]$ при помощи оператора d_l:

$$d_l(x) = \begin{cases} x < l \;\Rightarrow\; \text{не кореферентны}, \\ x \geq l \;\Rightarrow\; \text{кореферентны}. \end{cases}$$

Пусть

$$s_{w_i} = \begin{pmatrix} s_{w_i}(w_1) \\ s_{w_i}(w_2) \\ \vdots \\ s_{w_i}(w_n) \end{pmatrix}, \quad p_{w_i} = \begin{pmatrix} p_{w_i}(w_1) \\ p_{w_i}(w_2) \\ \vdots \\ p_{w_i}(w_n) \end{pmatrix}, \quad \forall j\, w_j \in w_1^n.$$

Полученное от механизма внимания распределение p_{w_i} выражено вектором. Тогда найденное от механизма внимания распределение p_{w_i}, выраженное вектором p_{w_i}, может быть преобразовано в вектор оценок s_{w_i} следующим образом:

$$s_{w_i} = \frac{p_{w_i}}{128}.$$

Для построения и обучения модели используется фреймворк TensorFlow и включенный в него Keras [33]. Для токенизации и векторизации – определения векторных представлений токенов – применяются сторонние модели. Модель bert-base-ru-cased [34] семейства BERT используется для токенизации и векторизации.

Последовательность слоев токенизации, векторизации и механизма внимания формирует модель, изображенную на рис. 14. Часть модели, состоящей из слоя механизма внимания, в данной работе называется внутренней моделью inner_model. Внутренняя модель проходит обучение, после чего встраивается в основную модель, слои которой не требуют обучения. Внутренняя модель принимает тензор из вещественных значений размера $128 \times \mathrm{emb}_s$, где emb_s – размер векторного представления, и возвращает матрицу 128×128 вещественных значений.

Модель отличается небольшим размером. Слой механизма внимания имеет 49 тыс. скрытых параметров. Наибольшей вычислительной сложностью обладает предобученный слой векторизации.

5.2. Конвейер входных данных. В статье используется корпус RuCoCo. Он состоит из 3075 текстов новостей и включает около 150 тыс. именных групп. Корпус представлен в формате JSON, где каждый файл содержит текст и информацию о кореферентных связях. JSON-структура имеет следующие ключевые свойства:

1) text: строка с текстом;

2) entities: список кореферентных цепочек, каждая из которых включает пары индексов, указывающих на начало и конец именных групп в тексте;

3) includes: список номеров дочерних цепочек для каждой основной цепочки.

Фреймворк TensorFlow предоставляет удобный способ представления корпусов, называемый конвейером данных. Исходные данные в корпусе RuCoCo хранятся в виде json-файлов, когда все модели работают только с числовыми векторами и матрицами. Конвейер данных задает последовательность предварительных преобразований файлов из корпуса в обучающий набор примеров.

Приведенный формат файлов в корпусе не удобен для модели. Большинство математических вычислений в ходе обучения и использования моделей машинного обучения стремятся представить в виде операций над тензорами. Это обусловлено следующим:

а) операции над тензорами: сложение, умножение, могут выполнятся параллельно;

б) обучение и применение моделей может проходить на видеокарте. Архитектура видеокарт заточена под операции с тензорами, используемых в трехмерной графике, и выполняют их быстрее центрального процессора;

в) модель может обрабатывать не один пример, а несколько, объединенных в пакет.

Например, если обычно модель принимает на вход вектор из 10 значений, n входных значений могут быть собраны в тензор размера $n \times 10$ и проводимые с этим тензором расчеты в рамках вычислительного графа будут аналогичны тем, которые бы проводились на простом векторе размером 10. Существенным ограничением такого подхода, особо заметным на задачах обработки текстов, является трудность обработки данных произвольной длины. Входной текст может содержать различное число токенов. Потому данные из файлов корпуса необходимо представить в другой форме. Текст должен быть токенизирован, векторизирован и разбит на блоки по 128 токенов. Набор кореферентных цепочек нужно преобразовать в набор векторов оценок, составляющих двухмерный тензор размером 128×128.

5.3. Обучение и тестирование. Входной датасет делится на три части: обучающий, тестирующий и проверяющий. Поскольку все три набора получаются из одного источника, элементы каждого из них можно считать распределенными согласно одному порождающему распределению. При каждом вызове функция перемешивает элементы и компонует их в пакеты для последующего использования. Обучающий датасет составляет 64% от общего

числа примеров, тестирующий – 20%, проверяющий – 16%. Процесс обучения модели применяет функцию ошибки в виде бинарной кросс-энтропии. Ее значение в ходе обучения достигло 0.6.

Оценка качества модели в задаче кореференции представляет определенные сложности. Метрики, чувствительные к частичному определению именных групп, могут зафиксировать ошибки, когда только часть правильных токенов была идентифицирована. Тем не менее строгое определение всех элементов именной группы не всегда критично. Основным результатом является определение контекста вокруг кореферентных именных групп, позволяющего извлекать информацию об объектах.

В полученной модели, использующей метрику MUC для оценки кореференции, полнота составляет всего 8%, что указывает на высокую вероятность ошибки первого рода (92%). Следовательно, модель имеет тенденцию пропускать значительное количество действительных кореферентных связей, т.е. она не связывает между собой те именные группы, которые действительно относятся к одному и тому же референту. Несмотря на это, точность модели равна 98%, что свидетельствует о низкой вероятности ошибки второго рода (2%). Когда модель решает установить кореферентные связи, она делает это с высокой степенью уверенности, но зачастую ошибается в пропуске реальных связей, не ошибаясь в определении неправильных.

Такой результат демонстрирует, что модель крайне редко ошибается, когда связывает именные группы, однако в то же время имеет низкую способность выявлять все возможные кореферентные связи, что может указывать на недостаточную чувствительность модели к разнообразию контекста или особенностей языка.

Заключение. Современная обработка текста базируется на том обстоятельстве, что смысл слов и словосочетаний заключается в распределении вероятности вокруг них в тексте. Для работы с такими распределениями современные языковые модели применяют механизм кодировщика для преобразования слов и фраз в вектора многомерного вещественного пространства, называемые векторными представлениями токенов.

В данной работе предлагается методика использования векторных представлений токенов, полученных от сторонней генеративной языковой модели. Для каждой пары (w_1, w_2) токенов методами машинного обучения вычисляется оценка вероятности $\mu(w_1, w_2)$ того, что оба токена входят в кореферентные именные группы на основе их векторных представлений. Далее применяется пороговая функция с параметром-порогом $M \in [0, 1]$, которая отсекает маловероятные пары.

Таким образом, с помощью современных методов машинного обучения в задаче кореференции значительно улучшается эффективность анализа текстов и поиск информации, сосредоточенной вокруг конкретных сущностей и объектов. В дальнейшем планируется исследовать возможности повышения полноты модели путем дообучения на более разнообразных корпусах данных или использования дополнительных языковых моделей.

СПИСОК ЛИТЕРАТУРЫ

1. *Гирудкий А.А.* Введение в языкознание. Минск: Высш. шк., 2022. ISBN 978-985-06-3430-6.
2. *Chomsky N.* Aspects of the Theory of Syntax. Cambridge: MIT press, 2014. № 11.
3. *Nivre J., Zeman D., Ginter F., Tyers F.* Universal Dependencies // 15th Conf. of the European Chapter of the Association for Computational Linguistics. Valencia, 2017.
4. *Sukthanker R., Poria S., Cambria E., Thirunavukarasu R.* Anaphora and Coreference Resolution: A Review // Information Fusion. 2020. V. 59. P. 139–162;
   https://doi.org/10.1016/j.inffus.2020.01.010
5. *Soon W.M., Lim D.C.Y., Ng H.T.* A Machine Learning Approach to Coreference Resolution of Noun Phrases // Computational Linguistics. 2001. V. 27. № 4. P. 521–544;
   https://doi.org/10.1162/089120101753342653
6. *Toldova S., Ionov M.* Coreference Resolution for Russian: The Impact of Semantic Features // Computational Linguistics and Intellectual Technologies. 2017. V. 1. № 16. P. 339–348.
7. *Haghighi A., Klein D.* Simple Coreference Resolution with Rich Syntactic and Semanticfeatures // Conference on Empirical Methods in Natural Language Processing (EMNLP). Singapore, 2009. P. 1152–1161;
   https://doi.org/10.3115/1699648.1699661
8. *Le. K., He L., Lewis M., Zettlemoyer L.* End-to-end Neural Coreference Resolution // Conference on Empirical Methods in Natural Language Processing (EMNLP). Copenhagen, 2017. P. 188–197;
   https://doi.org/10.18653/v1/d17-1018

9. *Hochreiter S., Schmidhuber J.* Long Short-Term Memory // Neural Computation. 1997. V. 9. № 8. P. 1735–1780; https://doi.org/10.1162/neco.1997.9.8.1735.
10. *Olah C.* Understanding LSTM Networks. 2015. [Электронный ресурс] URL: http://colah. github.io/posts/2015-08-Understanding-LSTMs/
11. *Lee K., He L., Zettlemoyer L.* Higher-order Coreference Resolution with Coarse-to-fine Inference // Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT). 2018. V. 2. P. 687–692;
https://doi.org/10.18653/v1/n18-2108
12. *Le T.A., Petrov M.A., Kuratov Y.M., Burtsev M.S.* Sentence Level Representation and Language Models in the Task of Coreference Resolution for Russian // Computational Linguistics and Intellectual Technologies. 2019. V. 2. № 18. P. 364–373.
13. *Shen T., Zhou T., Long G., Jiang J., Pan S., Zhang C.* Disan: Directional Self-Attention Network for RnN/CNN-free Language Understanding // 32nd AAAI Conference on Artificial Intelligence (AAAI). 2018. P. 5446–5455.
14. *Peng H., Khashabi D., Roth D.* Solving Hard Coreference Problems // Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT). 2015. P. 809–819;
https://doi.org/10.3115/v1/n15-1082
15. *Sysoev A.A.* Coreference Resolution in Russian: State-of-the-Art // Approaches Application and Evolvement. 2017. V. 16. P. 327–347.
16. *Toldova S.Ju., Roytberg A., Ladygina A.A. et al.* RU-EVAL-2014: Evaluating Anaphora and Coreference Resolution for Russian // Computational Linguistics and Intellectual Technologies. 2014. № 13. P. 681–694.
17. *Bogdanov A.V., Dzhumaev S.S., Skorinkin D.A., Starostin A.S.* Anaphora Analysis Based on ABBYY Compreno Linguistic Technologies // Computational Linguistics and Intellectual Technologies. 2014;
https://doi.org/10.13140/2.1.2600.7688
18. *Anisimovich K.V., Druzhkin K.Y., Zuev K.A.* Syntactic and Semantic Parser Based on ABBYY Compreno Linguistic Technologies // Computational Linguistics and Intellectual Technologies. 2012. V. 11. № 18. P. 90–103.
19. *Ionov M., Kutuzov A.* The Impact of Morphology Processing Quality on Automated Anaphora Resolution for Russian. M., 2014. № 13. P. 232–241.
20. *Kamenskaya M., Khramoin I., Smirnov I. et al.* Data-driven Methods for Anaphora Resolution of Russian Texts // Computational Linguistics and Intellectual Technologies. 2014. P. 241–250.
21. *Protopopova E.V., Bodrova A.A., Volskaya S.A. et al.* Anaphoric Annotation and Corpus-based Anaphora Resolution: An Experiment // Computational Linguistics and Intellectual Technologies. 2014. № 13. P. 562–571.
22. *Budnikov A.E., Toldova S.Y., Zvereva D.S. et al.* Ru-eval-2019: Evaluating Anaphora and Coreference Resolution for Russian // Computational Linguistics and Intellectual Technologies. 2019.
23. *Vilain M., Burger J.D., Aberdeen J. et al.* A Model-Theoretic Coreference Scoring Scheme // Conference on Message Understanding. Columbia: Association for Computational Linguistics, 1995. P. 45–52;
https://doi.org/10.3115/1072399
24. *Bagga A., Baldwin B.* Algorithms for Scoring Coreference Chains // The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference. Citeseer. 1998. V. 1. P. 563–566.
25. *Luo X.* On Coreference Resolution Performance Metrics // Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language. Vancouver: Association for Computational Linguistics, 2005. P. 25–32;
https://doi.org/10.3115/1220575.1220579
26. *Pradhan S., Moschitti A., Xue N. et al.* CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes // Joint Conference on EMNLP and CoNLL-shared task. Jeju Island, 2012. P. 1–40.
27. *Moosavi N.S., Strube M.* Which Coreference Evaluation Metric Do You Trust? A Proposal for a Link-based Entity Aware Metric // Proc. 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. V. 1. P. 632–642;
https://doi.org/10.18653/v1/P16-1060
28. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient Estimation of Word Representations in Vector Space // ArXiv preprint arXiv:1301.3781. 2013.
29. *Olah C.* Understanding LSTM Networks. 2015. [Электронный ресурс] URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
30. *Hochreiter S., Schmidhuber J.* Long Short-term Memory // Neural computation. 1997. V. 9. P. 1735–1780;
https://doi.org/10.1162/neco.1997.9.8.1735
31. *Bahdanau D., Cho K., Bengio Y.* Neural Machine Translation by Jointly Learning to Align and Translate // ArXiv preprint arXiv:1409.0473. 2014.
32. *Luong M.-T., Pham H., Manning C.D.* Effective Approaches to Attention Based Neural Machine Translation // ArXiv preprint arXiv:1508.04025. 2015.
33. *Abadi M., Agarwal A., Barham et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. [Электронный ресурс] URL: https://www.tensorflow.org/
34. *Abdaoui A., Pradel C., Sigel G.* Load What You Need: Smaller Versions of Mutlilingual BERT // SustaiNLP / EMNLP. ArXiv:2010.05609. 2020.