Федеральное государственное автономное образовательное
учреждение высшего образования
Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

Выпускная квалификационная работа
на тему

Извлечение именованных сущностей для языков с малыми ресурсами

Выполнила студентка группы БПМИ151, 4 курса,
Закирова Ксения Игоревна
Научный руководитель:
Доцент, кандидат технических наук,
Артемова Екатерина Леонидовна

Москва, 2020

Содержание

Abstract

In this work, I investigate the approaches to the problem of named entity recognition in the Tatar language. The Tatar language is low-resource, so I tackled both initial data collection and modelling. I automatically annotated corpora Tugan Tel and Wikipedia and I present a list of named entities. Corpora contain labels such as PER, LOC, ORG and MISC in BIO notation. I trained BiLSTM-CRF and BERT models. The BERT-based model achieves 0.47 average F-score.

Keywords— Named Entity Recognition, NER, Tatar language, low-resource languages

Аннотация

В данной работе я рассмотрела задачу извлечения именованных сущностей в татарском языке, собрала данные для корпуса Википедии и обучила машинную модель. Татарский является малоресурсным языком, для которого нет доступных решений в литературе. Результатом работы является список именованных сущностей и размеченный корпус на основе Википедии, который я предоставляю в открытый доступ. Корпус содержат теги PER, LOC, ORG и MISC в нотации BIO. Я обучила модели BiLSTM-CRF и BERT. BERT показал средний результат метрики f-score 0.47 на тестовом наборе, который был размечен вручную.

Keywords— Извлечение именованных сущностей, татарский язык, малоресурсные языки

1 Введение

Извлечение именованных сущностей (Named entity recognition, NER) это одна из задач обработки естественного языка. Словосочетания, однозначно определяющие некоторого человека, организацию, географический или другой объект, называются именованными сущностями. Задача обнаружения и классификации таких словосочетаний в тексте на несколько заранее определённых категорий и есть задача извлечения именованных сущностей. На вход подаётся текст (предложение), на выходе — массив из меток для каждой словоформы (словоформы – слова, знаки препинания, числа которые есть в тексте).

B-PER	I-PER	O	O	B-ORG	I-ORG	I-ORG	O	B-TIM	O	O
Иван	Петров	преподает	в	Высшей	Школе	Экономики	с	2014	года	.

Таблица 1.1: Пример размеченных данных, использована нотация BIO

Извлечение именованных сущностей имеет множество применений: в автоматическом разделении на категории текстов, в рекомендательных системах, в системах извлечения информации. Как задача извлечение именованных сущностей была сформулирована ещё в 1996 году [1], однако широкое распространение получила только в последнем десятилетии [2]. Развитие глубоких нейронных сетей дало значительный толчок развитию обработки естественных языков, и, как следствие, задаче извлечения именованных сущностей. Были изобретены более эффективные и точные модели, которые показывают хорошие результаты. Однако остаются и нерешенные проблемы, связанные с данной задачей. Во-первых, упомянутые выше модели с хорошими результатами существуют только для широко распространённых языков, для которых имеются

размеченные корпусы, а языки, не входящие в «топ-10» по числу носителей, оказываются вне внимания исследователей. Во-вторых, у компаний и исследователей недостаточно причин прикладывать усилия к задаче извлечения именованных сущностей для языков с малыми ресурсами, так как, скорее всего, это не сможет принести большой выгоды в дальнейшем из-за сравнительно небольшого числа носителей. У меня есть причина личного характера: татарский язык является родным языком для меня, и я стараюсь сохранять и развивать его, в том числе и с помощью этой работы.

Помимо патриотических мотивов есть и прагматические мотивы: решение задачи извлечения именованных сущностей для татарского языка может быть использовано для всей кыпчакской группы тюркской ветви языков (татарский, башкирский, карачаево-балкарский, казахский, киргизский и др.). Это связано с тем, что языки тюркской ветви достаточно похожи между собой, как грамматически, так и лексически. Как следствие, решение задачи для одного языка может быть и для других языков данной группы.

В тюркской языковой семье языком с наиболее богатыми ресурсами является турецкий язык, поскольку он является национальным языком. Остаётся открытым вопрос, насколько ресурсы турецкого языка могли бы помочь для решения задачи извлечения именованных сущностей на татарском языке. В данной работе я отказалась от данного подхода по следующим причинам:

- турецкий язык относится к другой группе языков: огузской;
- турецкий язык использует латинский алфавит, в то время как татарский язык преимущественно записывается на кириллице

Работа [3] исследователей из Академии наук Республики Татарстан даёт рекомендации к разметке корпуса. Далее я более подробно рассмотрю их работу в своем исследовании.

Целью работы является получить размеченный корпус и обученную модель, распознающую именованные сущности и сравнить полученные результаты с ранее имеющимися в этом поле. Работа содержит в себе обзор литературы, получение и разметку данных, выбор двух лучших известных моделей, обучение моделей и экспериментальную оценку. Рассматриваемые модели решают задачу, сформулированную следующим образом: на вход модели подаётся произвольный текст (предложение), а на выходе получается текст с тегами для каждой именованной сущности, встречающейся в тексте.

2 Обзор литературы

2.1 Стандартные подходы к извлечению именованных сущностей

В настоящий момент лучшей моделью на классическом датасете CoNLL 2003 [4] по оценке сайта paperswithcode.com является Delta [5] (модель на основе BERT), также высокие места занимают такие модели как CNN [6], GCDT [7], I-DARTS + Flair [8], LSTM-CRF [9], рис. **??**.

Для более глубокого погружения в тему рекомендуется ознакомиться со всеми моделями, приведенными выше; я же проведу краткий обзор моделей, которые были использованы в работе.

2.1.1 Low-Resource Named Entity Recognition with Cross-Lingual, Character-Level Neural Conditional Random Fields

Ryan Cotterell и Kevin Duh [10] предлагают модель условных случайных полей (Conditional Random Fields, CRF) и представляют способы её улучшения с помощью обучения модели на языках с большими ресурсами и последующего применения её к языку из того же семейства, но с меньшими ресурсами. В данной работе рассматривались семья индоевропейских языков, ветви: романская, германская, славянская, индоарийская; и семья автронезийских языков, ветвь: филиппинская.

Таблица 2.1: Пример предложения из датасета CoNLL

Word	POS tag	syntactic chunk tag	named entity tag
U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

2.1.2 A Neural Layered Model for Nested Named Entity Recognition

Meizhi Ju, Makoto Miwa и Sophia Ananiadou [11] представляют модель Layered-BiLSTM-CRF, которая работает с «наслоенными» именованными сущностями, т.е. когда одна именованная сущность частично или полностью входит в другую именованную сущность (рис. 2.2). Используются последовательно идущие плоские слои, где каждый слой состоит из модели двунаправленной долгой краткосрочной памяти (Bidirectional long short-term memory, BiLSTM) [12] и поверх неё модели условных случайных полей (Conditional Random Fields, CRF) [13].

2.1.3 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Данная статья исследователей из Google AI Language[14], которая на момент опубликования является лучшей моделью практически на всех популярных бенчмарках обработки естественного языка. Для моей задачи очень хорошо подходит претренированная модель bert-base-multilingual-cased, в обучающие данные которой входила также и Википедия на татарском языке.

BERT — это очень большая модель, которая не решает задачи извлечения именованных сущностей сама по себе, поэтому я воспользовалась библиотекой Transformers фирмы Huggingface [15]

2.1.4 Huggingface Transformers

Данная библиотека предоставляет возможность использовать различные модели (не только BERT, но и многие другие) для решения различных задач обработки естественного языка, в том числе и извлечения именованных сущностей. Их репозиторий на github содержит множество примеров для удобного использования их библиотеки.

Библиотека Huggingface Transformers содержит реализацию многоклассового классификатора словоформ, использующего векторное представление слов, которое выдаёт на выходе модель BERT выдаёт Поскольку модель BERT дополнительно разбивает словоформы на токены (отдельные знаки или часто встречающиеся комбинации знаков), то классификация производится для каждого токена независимо. В качестве конечного результата модели используются только теги, предсказанные для первого токена в каждом слове, остальные теги игнорируются. Следствием выбранной архитектуры классификатора является то, что модель не содержит внутри

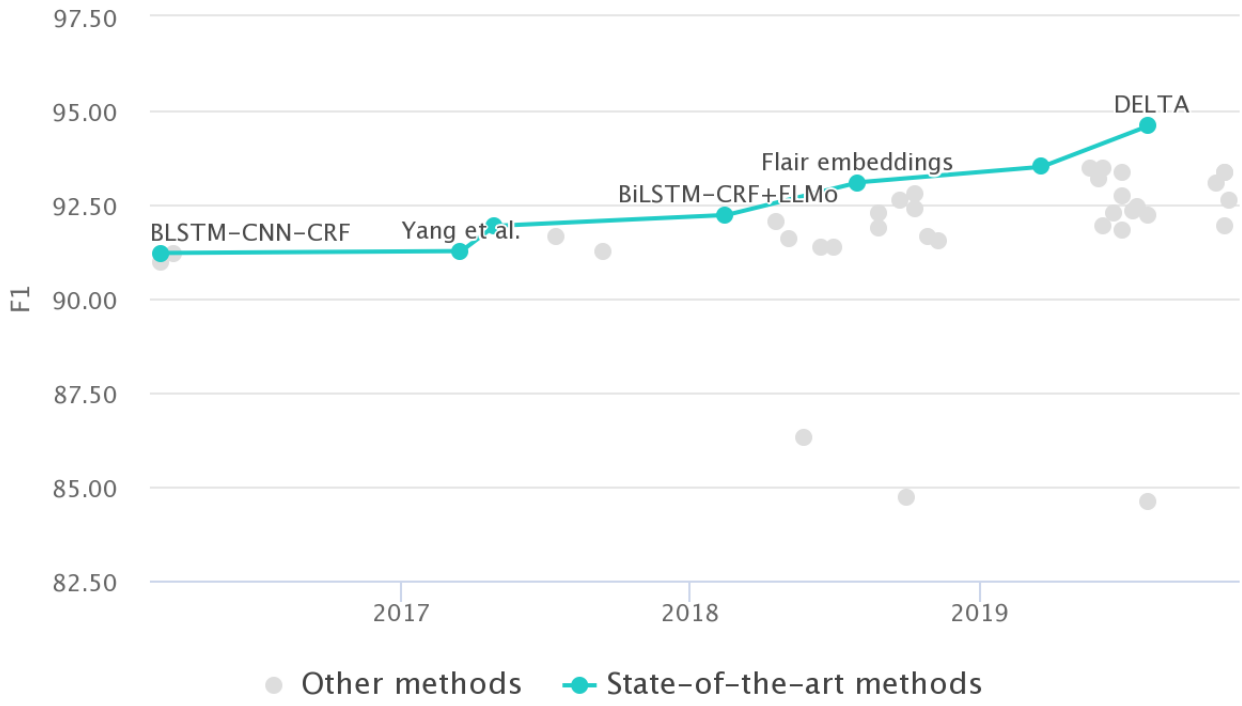
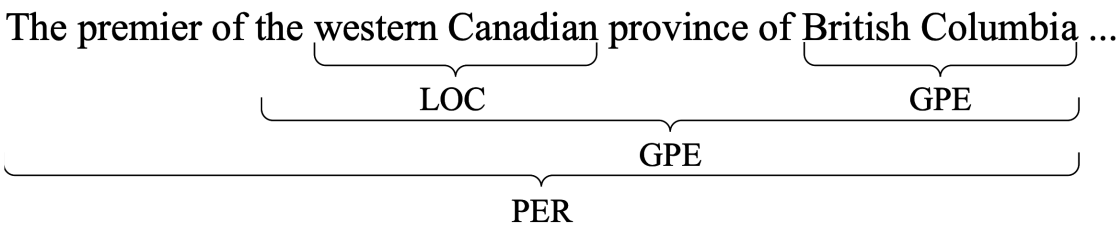


Рис. 2.1: Лучшие модели для задачи извлечения именованных сущностей на датасете CoNLL 2003, f-мера считается по чанкам

Рис. 2.2: Пример «наслоенных» именованных сущностей



себя ограничений на последовательность тегов, и иногда может предсказывать последовательности тегов, не являющиеся допустимыми по правилам разметки BIO, например, тег I-PER сразу после тега O.

2.2 Стандартные подходы к разметке данных

Текст делится на словоформы, т.е. каждый элемент в предложении отделяется от другого. Словоформами являются не только слова, но и знаки препинания, числа, эмодзи.

Стандартным (самой распространённым) форматом разметки для корпусов текста для задачи извлечения именованных сущностей является разметка IOB (сокр. от Inside–outside–beginning), иногда также называется BIO. Она была представлена в работе Text Chunking using Transformation-Based Learning [16].

Разметка BIO предназначена для выделения в тексте словосочетаний, несущих определённую смысловую нагрузку, например, именованных сущностей. Пометки (теги) ставятся на отдельные словоформы. Все словоформы, не входящие в состав именованных сущностей, получают метку 'O', а словосочетания помечаются с помощью меток с префиксами 'B-' или 'I-', приписанным к

тегу именованной сущности. Таким образом, используется три вида меток:

- Метки с префиксом 'B-' указывают, что словоформа является первой в словосочетании, обозначающем именованную сущность;
- Метки с префиксом 'I-' указыают, что словоформа входит в словосочетание, обозначающее именованную сущность, но не является первой, то есть является продолжением или окончанием словосочетания;
- Метка 'O' указывает, что словоформа не относится к именованной сущности.

Теги могут быть различными; устанавливаются на усмотрение исследователя, примеры тегов: PER (персона), LOC (географический объект), ORG (организация), TIM (время) и другие. Разметка в тексте выглядит следующим образом:

B-PER	I-PER	O	O	B-LOC	I-LOC	O
Василий	Иванов	проживает	в	Российской	Федерации	.

Как можно видеть из примера выше, теги могут относиться не только к одной отдельной словоформе, но и к целым фразам, тогда они переносятся на слова с помощью меток $B-, I-$.

Одной из практических проблем, возникающих при решении задачи извлечения именованных сущностей, является то, что определение именованной сущности достаточно субъективно, и иногда разные люди могут не сходиться во мнении, является ли некоторое словосочетание именованной сущностью или нет. Рассмотрим следующий пример:

B-PER или O?	O	O	B-LOC	O
Внук	Ахмета	живёт	в	Москве	.

По поводу слова "Москве"разногласий не возникает, однако возможны различные мнения про словосочетание "внук Ахмета". В одной интерпретации именованной сущностью является имя "Ахмета которое обозначает названного человека. В другой интерпретации, именованной сущностью является словосочетание "внук Ахмета поскольку из контекста ясно, что речь идёт об определённом человеке, который является внуком Ахмета и живёт в Москве.

Meizhi Ju, Makoto Miwa и Sophia Ananiadou в своей работе «A Neural Layered Model for Nested Named Entity Recognition» [11] приводят следующий пример 2.2, где «Mary's husband» является именованной сущностью. «Многослойное» моделирование именованных сущностей позволяет разрешить данное несогласие, объявив оба словосочетания именованными сущностями, но на разных уровнях разметки.

В данной работе используются лишь однослойная разметка, которая придерживается первой интерпретации, то есть именованной сущностью объявляется наикратчайшее словосочетание, которое обозначает определённого человека, организацию и проч., кроме случаев, когда имя человека входит в название организации или географического объекта (например, "улица Фрунзе"является именованной сущностью категории LOC).

Когда люди размечают именованные сущности, они получают f-score равный 96.95% [17], что означает неточность даже при ручной разметке данных. «Золотыми» данными являются либо данные, которые исследователи отметили как правильные, либо выбранные «коллективно»: один и тот же текст даётся на разметку, например, 5 людям, и если 3 из 5 высказались, что это именованная сущность, тег оказывается в «золотых» данных.

John	B-PER	O	O
killed	O	O	O
Mary	B-PER	B-PER	O
's	O	I-PER	O
husband	O	I-PER	O
.	O	O	O

Таблица 2.2: Пример разметки из работы [11]

2.3 Работы, связанные с извлечением именованных сущностей в малоресурсных языках

2.3.1 Datasets and Baselines for Named Entity Recognition in Armenian Texts

Тема моей работы очень близка к теме работы данных исследователей, за исключением языка: у них, как понятно из названия, армянский язык, который так же относится к малоресурсным языкам.

В отличие от моего случая, где существует релевантная работа, Т. Гукасян, Г. Давтян, К. Аветисян и И. Андрианов стали, можно сказать, первопроходцами в своей области, поскольку никто не делал подобных работ для армянского языка. У них не было подобранного и размеченного корпуса текста, поэтому, помимо извлечения именованных сущностей, они занимались также и сбором и разметкой данных. Их модель включала в себя CRF, которую я использую и в своей работе, и рекомендую как хорошую модель для языков с малыми ресурсами.

В своей работе исследователи не использовали BERT, поскольку это относительно новая модель, а статья вышла в конце 2018 года.

2.4 Работы, связанные с извлечением именованных сущностей в татарском языке

При поиске корпусов на татарском языке я нашла корпус Туган Тел — работу Невзоровой и др. [18].

2.4.1 Developing Corpus Management System: Architecture of System and Database

Туган Тел – это корпус текстов на татарском языке, разработанный Институтом прикладной семиотики Академии наук Республики Татарстан. Корпус предназначен для широкого круга пользователей: лингвистов, специалистов по татарскому языку, преподавателей татарского и всем тем, кому может понадобиться набор текстов на татарском языке. Основными функциями корпуса являются: поиск по словоформе, лемме (лексеме), набору морфологических параметров. Существует система «корпус-менеджер», которая поддерживает данные функции. На данный момент ведётся разработка электронного корпуса, а также проект автоматической разметки корпуса. Корпус включает в себя татарские тексты различных жанров, такие как художественная литература, тексты СМИ, тексты официальных документов, учебная литература, научные публикации и др. Каждый документ имеет метаописание, включающее в себя автора и его пол, выходные данные, дату создания, жанр, части, главы и др. Тексты, включенные в корпус, снаб-

жены автоматической морфологической разметкой, которая включает в себя информацию о части речи и грамматическую характеристику словоформы. Морфологическая разметка текстов корпуса выполняется автоматически с использованием модуля двухуровневого морфологического анализа татарского языка, реализованного в программном инструментарии PC-KIMMO. На декабрь 2019 года в корпусе 194 млн. словоформ.

2.4.2 Named Entity Recognition in Tatar: Corpus-Based Algorithm

Самой близкой к моей работе является статья «Named Entity Recognition in Tatar: Corpus-Based Algorithm» О. Невзоровой, Д. Мухамедшина и А. Галиевой, Академия наук Республики Татарстан. В статье они предлагают алгоритм разметки корпусов, используя в качестве примера корпус «Туган Тел» [18] на следующие категории: книги, рестораны, фильмы, журналы, компании, аэропорты, корпорации, языки, колледжи, университеты, школы, магазины, музеи и больницы.

1. **Использованные данные**

   Исследователи использовали корпус Туган тел [18].

2. **Разбор алгоритма, предложенного в статье:**

   Представленный алгоритм основан на идее сравнения частотности n-грамм. Сравнение происходит на всём объёме корпуса, что увеличивает точность результата, заявляют авторы статьи. Алгоритм является итеративным, количество итераций (т.е. максимальная длина полученных n-грамм) определяется пользователем.

   С помощью нулевого шага алгоритма получается выборка по поисковому запросу. Запрос может представлять собой форму слова, лемму, фразу или поиск по морфологическим параметром. Выборка представляет собой набор биграмм. В биграмме одно слово является запросом, а второе получено в результате поиска по корпусу. Оба слова могут дополнительно ограничиваться морфологическими параметрами. Далее полученный список из запроса просматривается и из него вручную удаляются биграммы, не являющиеся именованными сущностями. Полученная «чистая» выборка используется для первого шага алгоритма.

   Полученный список биграмм ищется в корпусе и к нему добавляется третье слово, которое стоит с ним рядом в тексте; добавляться слово может слева или справа, данный параметр выбирается пользователем в начале и не меняется в ходе алгоритма. Полученный список триграмм отсортировывается по частоте вхождений в корпус и в выборке остаются только самые частотные. Порог отсечения (в статье он называется «индекс покрытия», «covering index») более частотных вхождений также выбирается пользователем (обычно 95%). Урезанный по порогу список триграмм используется как входные данные для второй итерации алгоритма: каждая триграмма ищется по корпусу как фраза и, аналогично первой итерации, составляются 4-граммы и их частоты. Точно так же выбираются самые частотные 4-граммы, список обрезается по пороговому значению и, при желании, алгоритм продолжается дальше, используя на вход уже список 4-грамм.

   Таким образом алгоритм использует n-граммы для поиска $(n + 1)$-грамм, некоторые из которых будут отсечены порогом, а остальные использованы в следующем шаге алгоритма или попадут в список итоговых именованных сущностей.

3. **Окончание алгоритма:**

   В качестве условия для завершения алгоритма используется величина, называемая «точность сравнения» («accuracy of matching») P, которая задаётся в процентах. Если частота $(n + 1)$-граммы меньше P от количества найденных n-грамм, то алгоритм прекращает увеличивать длину n-граммы, иначе алгоритм переходит на следующую итерацию. Таким

образом, в финальный результат входят самые стабильные n-граммы разной длины, включая результаты изначального поискового запроса.

Стоит отметить, что все словосочетания, выделенные на нулевом шаге алгоритма, так или иначе считаются именованными сущностями, именно поэтому необходима ручная фильтрация на нулевом этапе; вопрос только в том, сколько слов справа или слева к этой биграмме добавится. Если алгоритм перешёл от n-грамме к $n+1$-грамме, то n-грамма не входит в финальный результат.

Запрос извлечения именованных сущностей представляет собой кортеж (1), где Q_1 и Q_2 — запрос в корпус-менеджер Туган Тел[18], L, R это, соответственно, порог ограничения итераций добавления слов слева и справа, C — порог отсечения частотности на каждой итерации (covering index), P — порог для принятия решения о включении фразы в итоговый список именованных сущностей (accuracy of matching).

$$Q = (Q_1, Q_2, L, R, C, P) \tag{1}$$

4. **Эксперименты:**

   Исследователи перечисляют довольно много категорий, над которыми они экспериментировали, но результаты они показали на словах «министерство», «улица», «язык», «ресторан» и «корпорация».

   Пример запроса со словом «министерство» (2):

$$Q = ((\text{wordform}, \text{ministrlygy}, «», \text{right}, 1, 10, \text{exact}), 7, 0, 95, 80) \tag{2}$$

   Также в данной статье очень интересный способ оценки результатов. Стандартные precision и recall (и производная от них F-score) в статье не упоминается, но оценивание полученных результатов производится. Происходит это следующим образом: вручную просматриваются все полученные n-граммы и классифицируются: на именованные сущности, «требует дополнительной очистки, тогда станет именованной сущностью», «требует расширения, тогда станет именованной сущностью», «это именованная сущность, но требует другой тег», «это именованная сущность, но требует дополнительной очистки и другой тег» и некорректные, см. таблицу 2.3. Данное оценивание не позволяет сравнивать результаты моей работы с результатами Невзоровой и др. напрямую, так как в моей работе поставлена другая задача. Однако, мы можем расширить их подход и применить к моей задаче на основе выделенных n-грамм.

2.5 Выводы

В области извлечения именованных сущностей написано много статей и изобретено много моделей, показывающих хорошие результаты на языках с большими ресурсами. Существуют так же работы по теме извлечения именованных сущностей для малоресурсных языков. Академия наук Республики Татарстан начала работу в данном направлении для татарского языка; я же, воспользовавшись их результатами, размечу корпус текстов на татарском языке, применю существующие модели к имеющимся данным и проведу сравнение с результатами алгоритма Невзоровой и др.

Class of named entity	Correct	Require filtering	Require expansion	Correct names of subclasses	Names of subclasses that require filtering	Incorrect	Total
Names of ministries	100%	0%	0%	0%	0%	0%	50
Street names	72%	12%	0%	0%	0%	16%	600
Language names	53.5%	0%	0%	0%	0%	46.5%	471 (2310)
Restaurant names	37.7%	18.3%	0%	13%	15.9%	15.1%	285
Corporation names	45.7%	19.6%	10.9%	21.7%	0%	2.2%	138

Таблица 2.3: Таблица 3 из статьи [3]

3 Методология

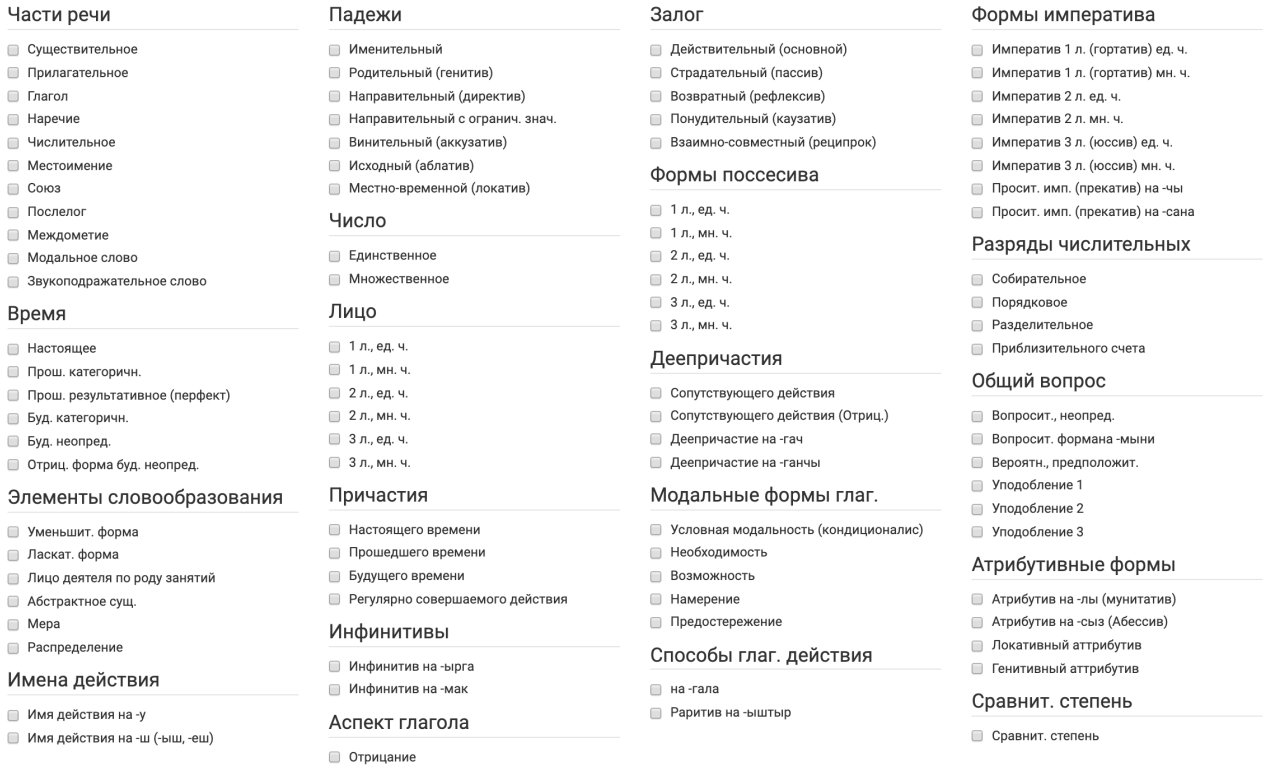
Целью работы было получить размеченный корпус и обученную модель, извлекающую именованные сущности. После обзора литературы были намечены задачи, и работа была предварительно разделена на несколько этапов.

1. Получение и разметка данных
2. Выбор, обучение и тюнинг моделей
3. Сравнение результатов

Но в ходе работы в план были внесены корректировки. Во-первых, как я упоминала ранее в обзоре литературы, представленные в статье Невзоровой результаты невозможно сравнивать с моими, поскольку постановка задач в моей и их работах различаются. Во-вторых, качество полученных данных оказалось не лучшим из возможных, а алгоритм Невзоровой, разработанный как раз для разметки данных, мог бы улучшить разметку корпуса, используемого для обучения моделей. Как следствие, было принято решение воспроизвести алгоритм из статьи Невзоровой и воспользоваться полученными результатами для разметки данных.

1. Получение и конвертирование данных в нужный формат.
2. Выбор, обучение и тюнинг моделей
3. Воспроизведение статьи Невзоровой
4. Разметка данных с помощью алгоритма Невзоровой
5. Обучение и тюнинг моделей
6. Сравнение результатов

Рис. 4.1: Параметры на сайте tugantel.tatar для поиска по корпусу



4 Получение и разметка данных

4.1 Туган Тел

Обзор литературы показал, что существует корпус татарских текстов Туган Тел[18]. Данный корпус имеет систему «корпус-менеджер», которая представлена в виде сайта. На этом сайте можно искать по словоформе или лемме с огромным количеством параметров [4.1], однако возможности просто скачать весь корпус не оказалось. Я предполагаю, что у Академии наук Республики Татарстан есть API для исполнения запросов на большом количестве данных и в каком-то более удобном формате, чем запрос на сайте, но у меня доступа к такому ресурсу не было.

Я связалась с Невзоровой по указанной в статье электронной почте, чтобы узнать подробности об их работе и попросить о сотрудничестве. Невзорова ответила на моё письмо и предоставила мне доступ к части корпуса, содержащей 30 млн словоформ.

Корпус представляет собой .zip файл, состоящий из 7557 .txt файлов, в общей сложности весом 1 183 023 978 Б. Как уже упоминалось ранее, корпус Туган Тел автоматически размечен с помощью программного инструментарии PC-KIMMO. Разметка выглядит следующим образом (см. рис 4.2). На нечетной строке написано слово, на следующей — разметка слова. Знаки препинания тоже являются «словоформами». Существует проблема с разделением текста на предложения, так как корпус не содержит никакой специальной разметки, обозначающей окончания предложений. Было принято решение разделять предложения по точкам, даже если это не самый точный способ разбиения.

Со всеми тегами морфоанализатора можно ознакомиться на сайте tatmorphan.pythonanywhere.com.

В тегах морфоанализатора присутствует тег PROP, который обозначает имена собственные. Для первой итерации было решено считать имена собственные именованными сущностями. Как

Рис. 4.2: Пример случайного предложения из корпуса Туган Тел

```
Аның
аны+PN+POSS_2SG(ың)+Nom;аның+PN;ул+PN+GEN(ның);
дөньяга
дөнья+N+Sg+DIR(ГА);
күз
күз+N+Sg+Nom;
карашы
караш+N+Sg+POSS_3(Сы)+Nom;
хаман
хаман+Adv;
үзгәрми
үзгәр+V+NEG(мА)+PRES(Й);
.
Type1
```

Перевод: Его мировоззрение все ещё не меняется.

Рис. 4.3: Пример предложения из корпуса Туган Тел с атрибутом PROP

```
Type2
Исемем
исем+N+Sg+POSS_1SG(ым)+Nom;
тахир
тахир+PROP+Sg+Nom;
минем
мин+PN+GEN(ның);
.
Type1
```

Перевод: меня зовут Тахир.

можно заметить в примере на рис. 4.3, в корпусе имена собственные иногда написаны с маленькой буквы, что говорит о том, что данные содержат ошибки. Всего в текстах 30 753 824 слов, из них 534 514 это слова с атрибутом PROP, что составляет 1,7% от всех слов.

4.2 Татарская Википедия

На данный момент татарская Википедия содержит 89 252 статей, которые написаны как с помощью кириллической, так и с помощью латинской письменности. Данный раздел Википедии был открыт 15 сентября 2003 года и сначала функционировал исключительно на латинице, впоследствие статьи добавлялись с использованием обоих алфавитов; сейчас же достигнут консенсус об использовании единой системы категорий на кириллице, однако некоторые статьи до сих пор остаются латинизированными (примерно треть от всех имеющихся статей). Причин такой путаницы несколько.

Во-первых, проблема алфавита в татарском языке стояла ещё со времен Советского Союза, т.к. до 1927 года использовалась арабская письменность, с 1927 по 1939 — латинская письменность, а 5 мая 1939 года Президиум Верховного Совета Татарской АССР принял указ «О переводе татарской письменности с латинизированного алфавита на алфавит на основе русской график» и начал использоваться кириллический алфавит. Поскольку переход на другую письменность происходил принудительно, до сих пор ведутся дебаты о возвращении на латинский алфавит. В настоящий момент в республике Татарстан кириллица остаётся официальным алфавитом, однако

стало допустимым использование латиницы и арабицы при обращении граждан в государственные органы и латиницы при транслитерации. Существует официальное соответствие данных трёх алфавитов.

Во-вторых, в 2000-х годах существовала проблема с записью текстов на компьютере, вызванная отсутствием букв дополнительной кириллицы в стандартных раскладках.

В связи с этим статьи на латинице пришлось конвертировать в кириллицу и в то же время случайно не перевести английские названия (например, ссылки). Данная процедура была проведена с помощью автоматического скрипта, поэтому возможны артефакты в виде, например, слова «хттп».

Рис. 4.4: Статья «Камский бассейновый округ»

Чулман су бассейны округы

Чулман су бассейны округы — Русиядәге 20 су бассейны округларының берсе (Су кодексының 28-че статьясына ярашлы).

[үзгәртү | вики-текстны үзгәртү]

2006 елда Чулман елга бассейны һәм аның белән бәйле җир асты су объектларын махсус саклау максатында барлыкка килә.

Чулман су бассейны округы 10 коды белән билгеләнә.

- 10.01 — Чулман
  - 10.01.01 — Чулман[1]
    - 10.01.01.001 — Чулман башлангычыннан
    - 10.01.01.002 — Чулман
    - 10.01.01.012 — Иж башлангычыннан тамагыне хәтле
    - 10.01.01.013 — Ык башлангычыннан тамагыне хәтле
    - 10.01.02.001 — Агыйдел (елга)сы башлангычыннан
  - 10.01.03 — Нократ[2]
    - 10.01.03.001 — Чүпче башлангычыннан тамагыне хәтле
    - 10.01.03.002 — Нократ башлангычыннан Вятка шеһәренә хәтле

Искәрмәләр [үзгәртү | вики-текстны үзгәртү]

1. ↑ http://gis-lab.info/data/mp/gvr/s10.01.01.html
2. ↑ http://gis-lab.info/data/mp/gvr/s10.01.03.html

Рис. 4.5: Пример сгенерированных статей из Википедии

бассейны | **Эзләү**

Киңәйтелгән эзләү: Муафыйклык буенча тәртипләү ✕

Эзләү: (Төп) ✕

Әлеге вики-проектта «Бассейны» исемле бит ясарга! Шулай ук, эзләү ярдәмендә табылган битләрне карагыз.

Ука су **бассейны** округы

Ука су **бассейны** округы — Русиядәге 20 су **бассейны** округларының берсе (Су кодексының 28-че статьясына ярашлы). 2006 елда Ука елга **бассейны** һәм аның белән

7 Кб (106 сүз) - 21 мар 2018, 19:00

Чулман су **бассейны** округы

Чулман су **бассейны** округы — Русиядәге 20 су **бассейны** округларының берсе (Су кодексының 28-че статьясына ярашлы). 2006 елда Чулман елга **бассейны** һәм аның

6 Кб (127 сүз) - 16 авг 2013, 07:26

Югары Об су **бассейны** округы

Об су **бассейны** округы — Русиядәге 20 су **бассейны** округларының берсе (Су кодексының 28-че статьясына ярашлы). 2006 елда Югары Об елга **бассейны** һәм аның

6 Кб (154 сүз) - 7 май 2014, 22:24

Тын су **бассейны** округы

Тын су **бассейны** округы — Русиядәге 20 су **бассейны** округларының берсе (Су кодексының 28-че статьясына ярашлы). 2006 елда Тын елга **бассейны** һәм аның белән

1 Кб (77 сүз) - 2 апр 2013, 22:00

Также важно отметить, что Википедия представляет собой набор статей, написанных в академическом стиле, что не вполне отражает разнообразие языка в различных сферах употребления; в этом аспекте Туган Тел гораздо лучше. В Википедии присутствуют автоматически сгенерированные статьи, это ухудшает качество текстов как корпуса для обучения, так как некоторые фразы становятся частотными не из-за того, что они действительно часто используются в языке, а из-за множества сгенерированных статей. Например, статьи про бассейновые округа («бассейны» это не множественное число слова «бассейн», а принадлежность к третьему лицу). Статья про Камский бассейновый округ (рис. 4.4) и по тому же шаблону ещё много других статей про бассейновые округа (рис. 4.5).

Справедливости ради следует заметить, что в русской википедии они тоже сгенерированы автоматически. С другой стороны, Википедия имеет преимущество перед корпусом Туган Тел по количеству упоминаний географических названий.

4.3 Разметка данных для обучения

Первой итерацией было использование разметки PROP в корпусе Туган Тел, никакие другие теги морфоанализотора не использовались; Википедия не использовалась.

Во второй итерации использовался воспроизведенный алгоритм Невзоровой, в качестве исходного корпуса для которого использовалась Википедия. На данных из Википедии был получен список n-грамм[1], обозначающих именованные сущности по классам PER (персона), LOC (географический объект), ORG (организация) и MISC (названия языков) С помощью полученного списка была размечена Википедия с помощью сравнения n-грамм на точное равенство, в то время как Туган Тел никак не использовался при данной разметке и был отложен в качестве данных для оценивания.

Для разметки BIO был написан небольшой скрипт, с помощью которого вы можете также получить размеченные данные на своей локальной машине.

4.4 Разметка данных для оценивания

Автоматическая разметка не подходит для оценки результатов. Поэтому для оценивания воспроизведенного алгоритма Невзоровой и обученных моделей было принято решение разметить некоторое количество предложений вручную, в силу имеющихся знаний татарского языка. Предложения были выбраны из корпуса Туган Тел из соображений наличия в них хотя бы одной именованной сущности; для этого был использован алгоритм Невзоровой и если он определил наличие именованной сущности, то предложение добавлялось в список кандидатов на ручную разметку. Из полученного списка кандидатов предложения выбирались случайным образом, разметка алгоритмом Невзоровой удалялась до начала ручной разметки, чтобы не влиять на конечный результат. Получился тестовый набор данных golden-bio.txt [2], основанный на предложениях из корпуса Туган Тел (369 предложений).

4.5 Проблемы с разметкой данных

Как было описано и в статье Невзоровой, где исследователи глазами просматривали полученные результаты, отсеивали некорректные и улучшали свой алгоритм с помощью фильтров, так и в моей работе разметка данных происходила итеративно. Например, генерация статей про бассейны в Википедии была как раз выявлена в просмотре полученных результатов после разметки. Также выявлялись пробелы в алгоритме, например, некоторые географические названия, которые не попадали в список, были добавлены позже вручную. Для разметки тегом PER

[1] Список n-грамм, обозначающих именованные сущности, предоставлен на github.com/ksemiya/NER_in_Tatar

[2] golden-bio.txt также предоставлен на github.com/ksemiya/NER_in_Tatar

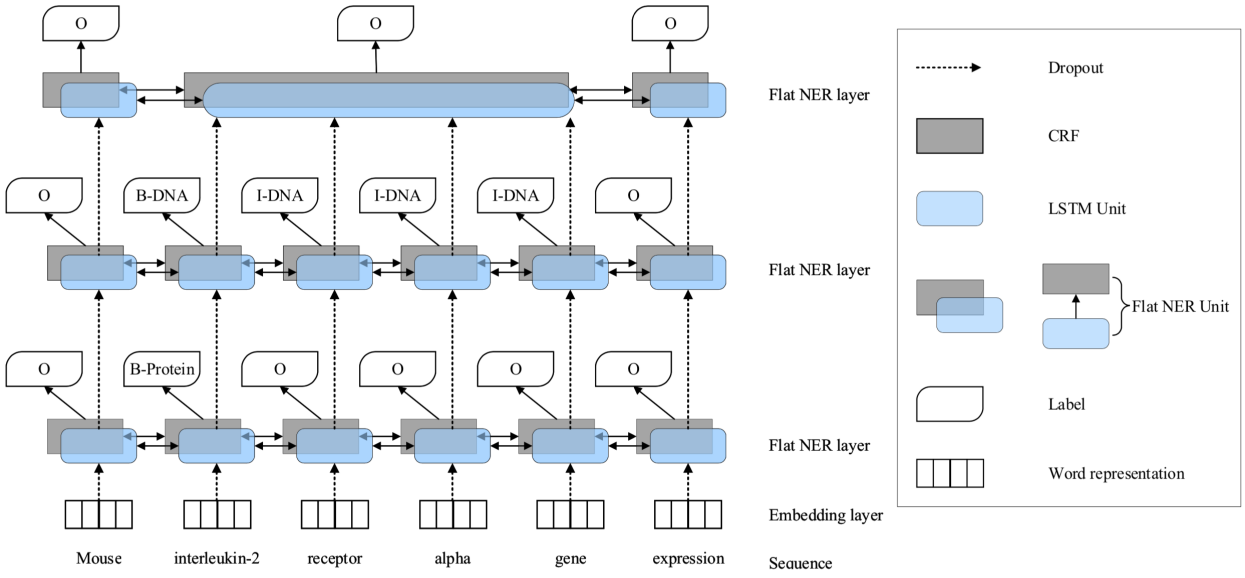
был использован справочник имён. Подводя итог, лучшей всё равно остаётся ручная разметка, а любая автоматическая разметка требует просмотра и последующей корректировки, возможно в несколько этапов, и это занимает очень много времени. С другой стороны, ручная разметка корпуса заняла бы ещё больше время.

5 Обучение и тюнинг моделей

5.1 BiLSTM-CRF

Была использована модель BiLSTM-CRF из статьи «A Neural Layered Model for Nested Named Entity Recognition» [11]. Она использует разметку BIO, как и многие другие модели для извлечения именованных сущностей. Архитектура модели изображена на рис. 5.1

Рис. 5.1: Архитектура модели BiLSTM-CRF, рис. из статьи [11]



Была возможность запустить модель только на локальной не очень мощной машине, поэтому пришлось обучать модель не на всех данных, а только на части.

Первая итерация на данных Туган Тел. Результат обучения, в данных два тега: O и PROP.

Precision	Recall	F-score
99.768	90.727	95.033

Таблица 5.1: Обучение BiLSTM-CRF на первых данных

Была сделана демонстрация работы модели в виде сайта, запускающегося на локальной машине, но самые простые примеры не из тестового набора выявили большие несовершенства дан-

ной модели — она не извлекала именованные сущности в самых простых предложениях, поэтому в итоге было принято решение отказаться от её использования. Как можно понять, цифры выше показывают способность модели обучаться на данных (и модель действительно обучается хорошо, выборка разделяется на тестовую и валидационную и всегда показывает хороший результат), но проблема в том, что сами данные очень низкого качества — и эту проблему модель, увы, исправить не может.

Вторая итерация обучения была произведена на Википедии, размеченной с помощью списка n-грамм, полученных с помощью алгоритма Невзоровой. В тренировочном наборе были оставлены только те предложения, где есть хотя бы одна именованная сущность.

Результат обучения, в данных 9 тегов: B-LOC B-MISC B-ORG B-PER I-LOC I-MISC I-ORG I-PER O

Precision	Recall	F-score
89.421	86.825	88.104

Таблица 5.2: Обучение BiLSTM-CRF на улучшенных данных

5.2 BERT

[14] Одна из самых известных моделей на сегодняшний день, показала лучшие результаты на классических данных CoNLL 2003 (см. обзор литературы).

Для решения моей задачи была использована библиотека Hugging face [15] и претренированная модель bert-base-multilingual-cased, которая обучена на чувствительных к регистру данных из 104-х наибольших Википедий. Данная модель включает в себя и татарский язык (беглый взгляд по токенам показал, что действительно есть как и кириллические, так и латинские токены на татарском языке).

При обучении модели возникли проблемы с размером корпуса данных. Из-за особенностей реализации обучения модели BERT в HuggingFace, библиотека не была способна сохранить все признаки полностью. Особенность была связана с реализацией загрузки данных в библиотеке. Ей нужно было сохранить все данные в виде признаков PyTorch. Обучение удалось произвести только лишь на 1/3 всех доступных данных.

Результат обучения, в данных два тега: O и PROP.

Precision	Recall	F-score
97.447	94.585	95.995

Таблица 5.3: Обучение BERT на первых данных

Один в один повторение истории с BiLSTM-CRF — сами по себе цифры очень многообеща-

ющие, модель обучилась прекрасно, но на деле они показывают всего лишь натренированность модели на плохих начальных данных. Модель в итоге не работает на простых примерах, придуманных из головы. Было принято решение от этой модели отказаться.

Вторая итерация обучения была произведена на Википедии, размеченной с помощью списка n-грамм, полученных с помощью алгоритма Невзоровой. В тренировочном наборе были оставлены только те предложения, где есть хотя бы одна именованная сущность.

Результат обучения, в данных 9 тегов: B-LOC B-MISC B-ORG B-PER I-LOC I-MISC I-ORG I-PER O

Precision	Recall	F-score
88.853	92.377	90.581

Таблица 5.4: Обучение BERT на улучшенных данных

Как видно, результаты заметно ухудшились, что неудивительно, ведь классов стало в 4.5 раза больше.

Однако все эти результаты всего лишь показывают, как хорошо модель обучилась предсказывать результаты на имеющихся данных. Чтобы проверить её реальную способность извлекать именованные сущности, необходимо проверить предсказания на «золотых» данных, размеченных вручную, про которые мы с высокой точностью знаем, что они правильные.

6 Воспроизведение алгоритма из статьи Невзоровой

Было принято решение воспроизвести алгоритм из статьи Невзоровой и др. [3] по двум причинам.

1. Чтобы была возможность сравнивать результаты.
2. Чтобы разметить данные эффективно и, по возможности, хорошо.

Воспроизводить алгоритм по описанию в статье было нетривиально, поэтому, если будут желающие воспроизвести его ещё раз, то я рекомендую описание, которое я дала выше в разделе обзор литературы.

Как было сказано ранее, у меня не было доступа к системе «корпус-менеджер» Туган Тел [18], поэтому воспроизвести алгоритм в точности не удалось. Попытка приблизиться к результату состояла в том, чтобы использовать в качестве начального слова все возможные формы слова, а не только слово само, т.е. частично сделать работу морфоанализатора вручную. Однако возможность делать запросы по каким-то морфологическим параметром не была реализована.

Несмотря на все вышеописанные трудности, алгоритм сработал достаточно хорошо, хотя и, как и сказано в работе Невзоровой и др., требовалась значительная ручная чистка полученных данных.

Ручная чистка состояла в просмотре полученных именованных сущностей и исправление п возможности не точечно, а «широкими мазками» — добавлением новых начальных слов для поискового запроса или удалением всех очевидно некорректных n-грамм одним вызовом команды grep. Например, по слову «министрлыгы» (министерство) появилось множество n-грамм,

которые начинались с союза «һәм» (и). Это некорректные n-граммы, но удалить их все было достаточно легко.

Также помимо слов, представленных в работе Невзоровой и др. в качестве начальных, были использованы и другие слова, такие как географические объекты: «елга» (река), «шәһәр» (город), «авыл» (село), «өлкә» (область) и др., организации: «академияс» (академия), «идарә» (администрация), «институт» (институт) и др., персоны: «абый» (старший брат), «апа» (старшая сестра) и др. (в татарском языке принято называть родственников, например, Ильдар абый, Сания апа, поэтому это часто встречающийся шаблон).

В результате получился список из 30 тысяч именованных сущностей, который доступен на моём github. Он может пригодиться людям, которые будут продолжать работу в данном направлении для разметки собственных корпусов или извлечения именованных сущностей.

7 Демонстрация полученных результатов

Был сделан демонстрационный http-сервер, работающий локально на рабочей станции, с помощью которого можно проверить модель на работоспособность.

Рис. 7.1: Старший научный сотрудник историческго института имени Мержани

Всё распознано корректно, категория ORG

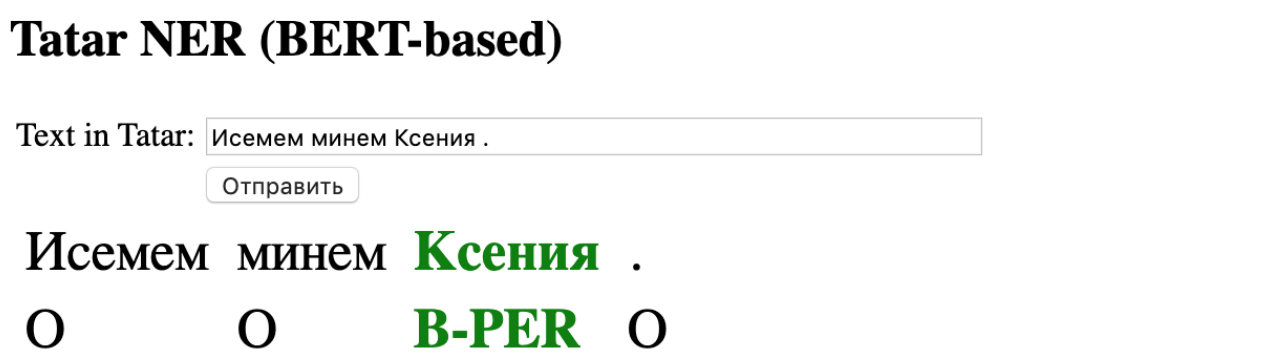


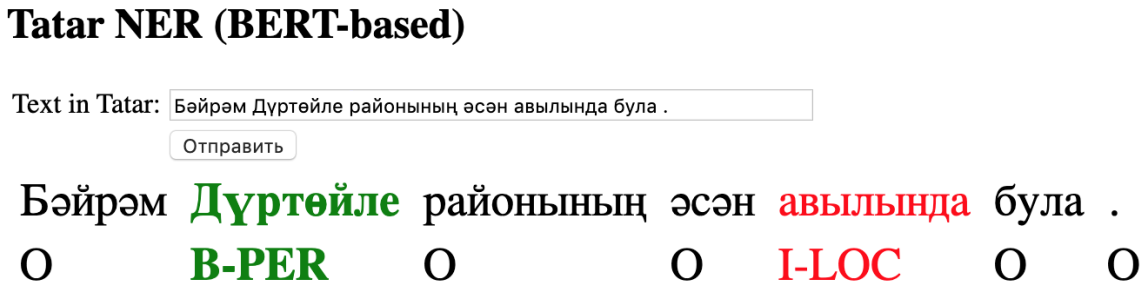
Рис. 7.2: Зовут меня Ксения

Всё распознано корректно, категория PER

Рис. 7.3: Он прочитал решение на русском языке

Всё распознано корректно, категория MISC

Рис. 7.4: Празднование состоится в селе Асан Дюртюлинского района



Не распознано имя села, хотя само понятие «село» распознано как категория LOC. «Дюртюлиский» ошибочно распознано как имя, «район» должно входить в название района.

8 Сравнение результатов

Отдельной задачей стоял вопрос, как сравнить полученные результаты с предыдущими результатами в данной области. Было принято решение разметить некоторое количество предложений вручную, чтобы иметь качественные «золотые» данные, про которые было бы известно, что вероятность ошибок в них гораздо ниже, чем в автоматически размеченных. Такая разметка была получена и следующим этапом воспроизведенный алгоритм Невзоровой, BiLSTM-CRF и BERT (последние два обученные на датасете, основанном на Википедии) были запущены на этих данных. Теперь полученные результаты допустимо сравнивать между собой, поскольку они «в равном положении» и выполняют одинаковую задачу. Но нужно заметить, что модели BiLSTM-CRF и BERT были обучены на данных, размеченных по методу Невзоровой и др., поэтому результаты моделей не полностью независимы.

Как видно из цифр таблиц [8.1], [8.2], [8.3], разметка данных с помощью алгоритма Невзоровой и др. далеко не идеальная, и модели выучили ровно столько, сколько данные могли им дать (однако у меня была надежда, что модели выучат больше, чем им дали на вход; для этого они обучались на данных, которые не содержали в себе «пустые» предложения, чтобы не учить их определять данные без меток).

category	precision	recall	**f1-score**	total
PER	0.53	0.63	**0.58**	374
LOC	0.50	0.05	**0.09**	78
ORG	0.40	0.10	**0.15**	21
MISC	0.83	0.62	**0.71**	8
macro avg	0.52	0.51	**0.48**	481

Таблица 8.1: Результаты алгоритма Невзоровой

category	precision	recall	**f1-score**	total
PER	0.53	0.60	**0.56**	374
LOC	0.50	0.05	**0.09**	78
ORG	0.33	0.10	**0.15**	21
MISC	0.56	0.62	**0.59**	8
macro avg	0.52	0.49	**0.47**	481

(a) Результаты модели BERT

category	precision	recall	**f1-score**	total
PER	0.52	0.44	**0.48**	374
LOC	1.00	0.03	**0.05**	78
ORG	0.33	0.10	**0.15**	21
MISC	0.44	0.50	**0.47**	8
macro avg	0.59	0.36	**0.39**	481

(b) Результаты модели Bi-LSTM-CRF

Таблица 8.2: Результаты нейросетевых моделей

model	precision	recall	**macro f1**
Алгоритм разметки данных	0.52	0.51	**0.48**
BERT	0.52	0.49	**0.47**
BiLSTM-CRF	0.59	0.36	**0.39**

Таблица 8.3: Сравнительная таблица, macro avg

9 Выводы

1. Воспроизведен алгоритм из статьи Невзоровой и др., он показал свою работоспособность и его можно использовать в дальнейшем.

2. Получен набор n-грамм именованных сущностей и размеченный корпус, с которыми можно будет работать в дальнейшем.

3. Получены модели, предсказывающие именованные сущности, но результаты этих моделей из-за качества данных оставляют желать лучшего.

4. Лучшие известные модели действительно работают хорошо и на татарском языке, обучаясь на данных настолько, насколько данные это позволяют, но увы, ничего сверх этого получить от них не удалось.

5. Всё зависит от того, насколько хорошо размечены данные, поэтому в будущих работах нужно лучше разметить данные, количество доступных данных было вполне достаточным для обучения моделей.

10 Заключение

Были поставлены задачи получить размеченный корпус и обученную модель, распознающую именованные сущности и сравнить полученные результаты с предыдущими работами в данной области. Задачи были выполнены в полном объёме. Были размечены данные с помощью воспроизведенного алгоритма Невзоровой, улучшены с помощью ручной доработки; небольшое количество данных размечено полностью вручную. Получен список n-грамм именованных сущностей, которые можно использовать в дальнейших работах по данной теме. Были обучены модели BiLSTM-CRF и BERT, которые показали результаты, сравнимые с предыдущей работой в данной области.

В дальнейшем можно будет сотрудничать с Академией наук Республики Татарстан и дальше продвигать направление извлечения именованных сущностей, пробовать новые модели не только извлечения, но также и разметки данных, поскольку с каждым годом корпус Туган Тел становится объемнее. Использовать в качестве признаков морфологические параметры и не только. Направлений для работы много и это хорошее поле для дальнейших исследований. Конкретные направления, которые хотелось бы выделить:

1. Улучшать разметку данных с помощью анализа текущих слабостей алгоритма Невзоровой, добавлением справочников и эвристик;

2. Разработать удобную систему для ручной разметки данных и привлекать к разметке больше людей, получая более точные результаты;

3. Попробовать другие лучшие известные модели, а именно сделать обёртку CRF над BERT, это скорее всего поможет решить проблему с I-тегами без B-тегов.

Код доступен по ссылке github.com/ksemiya/NER_in_Tatar

Выражаю благодарность О. Невзоровой за содействие в работе и предоставлении корпуса Туган Тел.

Список литературы

[1] Grishman R., Sundheim B. Message understanding conference-6: A brief history // Proceedings of the 16th Conference on Computational Linguistics - Volume 1. — COLING '96. — USA : Association for Computational Linguistics, 1996. — P. 466–471. — URL: https://doi.org/10.3115/992628.992709.

[2] A survey on deep learning for named entity recognition / Jing Li, Aixin Sun, Jianglei Han, Chenliang Li // CoRR. — 2018. — Vol. abs/1812.09449. — 1812.09449.

[3] Olga Nevzorova D. M., Galieva A. Named entity recognition in tatar: Corpus-based algorithm // semanticscholar. — 2018.

[4] Tjong Kim Sang E. F., De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition // Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. — 2003. — P. 142–147. — URL: https://www.aclweb.org/anthology/W03-0419.

[5] DELTA: A DEep learning based Language Technology plAtform / Han, Kun, Chen et al. // arXiv e-prints. — 2019. — URL: https://arxiv.org/abs/1908.01853.

[6] Cloze-driven Pretraining of Self-attention Networks / Alexei Baevski, Sergey Edunov, Yinhan Liu et al. // arXiv e-prints. — 2019. — URL: https://arxiv.org/abs/1903.07785v1.

[7] Gcdt: A global context enhanced deep transition architecture for sequence labeling / Yijin Liu, Fandong Meng, Jinchao Zhang et al. // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — 2019.

[8] Improved differentiable architecture search for language modeling and named entity recognition / Yufan Jiang, Chi Hu, Tong Xiao et al. // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). — Hong Kong, China : Association for Computational Linguistics, 2019. — nov. — P. 3585–3590. — URL: https://www.aclweb.org/anthology/D19-1367.

[9] Straková J., Straka M., Hajic J. Neural architectures for nested NER through linearization // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — Florence, Italy : Association for Computational Linguistics, 2019. — jul. — P. 5326–5331. — URL: https://www.aclweb.org/anthology/P19-1527.

[10] Cotterell R., Duh K. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields // Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers). — Taipei, Taiwan : Asian Federation of Natural Language Processing, 2017. — nov. — P. 91–96. — URL: https://www.aclweb.org/anthology/I17-2016.

[11] Ju M., Miwa M., Ananiadou S. A neural layered model for nested named entity recognition // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). — New Orleans, Louisiana : Association for Computational Linguistics, 2018. — jun. — P. 1446–1459. — URL: https://www.aclweb.org/anthology/N18-1131.

[12] LSTM: A search space odyssey / Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník et al. // CoRR. — 2015. — Vol. abs/1503.04069. — 1503.04069.

[13] Lafferty J., Mccallum A., Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data // Proceedings of the Eighteenth International Conference on Machine Learning. — 2001. — 01. — P. 282–289.

[14] BERT: pre-training of deep bidirectional transformers for language understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova // CoRR. — 2018. — Vol. abs/1810.04805. — 1810.04805.

[15] Huggingface's transformers: State-of-the-art natural language processing / Thomas Wolf, Lysandre Debut, Victor Sanh et al. // ArXiv. — 2019. — Vol. abs/1910.03771.

[16] Ramshaw L. A., Marcus M. P. Text chunking using transformation-based learning // CoRR. — 1995. — Vol. cmp-lg/9505040. — URL: http://arxiv.org/abs/cmp-lg/9505040.

[17] Marsh E., Perzanowski D. MUC-7 evaluation of IE technology: Overview of results // Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998. — 1998. — URL: https://www.aclweb.org/anthology/M98-1002.

[18] Nevzorova O. M. D., R. G. Developing corpus management system: Architecture of system and database // The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) / The Tatarstan Academy of Sciences, Kazan, Russia. — 2017. — P. 108–122.