

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

Распознавание 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

1	Введение	3
2	Обзор литературы	4
2.1	Стандартные подходы к распознаванию именованных сущностей	4
2.1.1	Low-Resource Named Entity Recognition with Cross-Lingual, Character-Level Neural Conditional Random Fields	4
2.1.2	A Neural Layered Model for Nested Named Entity Recognition	4
2.1.3	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	5
2.1.4	Huggingface Transformers	5
2.2	Стандартные подходы к разметке данных	6
2.3	Работы, связанные с распознаванием именованных сущностей в малоресурсных языках	6
2.3.1	Datasets and Baselines for Named Entity Recognition in Armenian Texts	6
2.4	Работы, связанные с распознаванием именованных сущностей в татарском языке	6
2.4.1	Developing Corpus Management System: Architecture of System and Database	7
2.4.2	Named Entity Recognition in Tatar: Corpus-Based Algorithm	7
2.5	Выводы	8
3	Методология	9
4	Получение и разметка данных	10
4.1	Туган Тел	10
4.2	Татарская Википедия	11
4.3	Разметка данных для обучения	14
4.4	Разметка данных для оценивания	14
4.5	Проблемы с разметкой данных	14
5	Обучение и тюнинг моделей	15
5.1	BiLSTM-CRF	15
5.2	BERT	16
6	Воспроизведение алгоритма из статьи Невзоровой	16
7	Демонстрация полученных результатов	17

8	Сравнение результатов	19
9	Выводы	20
10	Заключение	21
	Список литературы	21

Abstract

In this work, I try to solve the problem of Tatar named entity recognition, which connected with problems of NER in low resourced languages. I present a list with named entities and two automatically annotated corpora from Tugan Tel and Wikipedia. Corpora contain labels such as PER, LOC, ORG and MISC. I train a BiLSTM-CRF and BERT models, which provide 0.47 average F-score.

Keywords— Named Entity Recognition, NER, tatar language, low resourced languages

Аннотация

В данной работе я попыталась решить задачу распознавания именованных сущностей в татарском языке, который является языком с малыми ресурсами и для которого ранее не решалась данная проблема. По результатам работы я предоставляю список именованных сущностей и два размеченных с помощью алгоритма корпуса на основе корпуса Туган Тел и Википедии. Корпусы содержат теги PER, LOC, ORG и MISC. Я обучила модели BiLSTM-CRF и BERT, которые показали средний результат метрики f-score 0.47.

1 Введение

Распознавание именованных сущностей (Named entity recognition, NER) это одна из задач обработки естественного языка; задача обнаружения и классификации слов в тексте на несколько заранее определённых категорий, таких как, например, люди, места, организации и т.д. Распознавание именованных сущностей имеет множество применений, используется в автоматическом разделении на категории текстов, рекомендательных системах, системах извлечения информации. Как задача распознавание именованных сущностей была сформулирована ещё в прошлом веке, однако широкое распространение получила только в последнем десятилетии. Развитие глубоких нейронных сетей дало значительный толчок развитию обработке естественных языков, и, как следствие, задаче распознавания именованных сущностей. Были изобретены более эффективные и точные модели, которые показывают хорошие результаты. Однако существуют и серьёзные проблемы, связанные с данной задачей. Во-первых, упомянутые выше модели с хорошими результатами существуют только для широко распространённых языков, для которых имеются размеченные корпуса, а языки, которые не входят в «топ-10» по числу носителей, оказываются за бортом. Во-вторых, у компаний и исследователей нет причины вкладываться в задачу распознавания 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 (как и было сказано ранее). В то же время у меня есть причина: татарский язык является родным языком для меня, и я стараюсь сохранять его и продвигать его значимость, в том числе и с помощью такой работы.

Помимо патриотических мотивов существуют и мотивы прагматические: развитие распознавание 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 скорее всего будет иметь неплохие шансы и для других языков данной группы. К сожалению, это не сработает для турецкого языка, во-первых, потому что он относится к огузской группе, во-вторых (и это главная причина): там используется другой алфавит. Тюркская ветвь включает себя языки с различной письменностью, что усложняет возможность экстраполяции модели на «похожие» языки.

Также на тему конкретно татарского языка существует работа [1] исследователей из Академии наук Республики Татарстан. 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 текущими имеющимися в этом поле. Были сделаны обзор литературы, получение и разметка данных, обучение текущих лучших моделей.

TODO Помогите я не знаю как написать о своих целях и задачах.

2 Обзор литературы

2.1 Стандартные подходы к распознаванию именованных сущностей

Тут про модели, которые люди используют: BERT, CRF, LSTM и прочие.

На текущий момент лучшими моделями на классическом датасете CoNLL 2003 по оценке сайта paperswithcode.com является Delta [2] (модель BERT), также высокие места занимают такие модели как CNN [3], GCDT [4], I-DARTS + Flair [5], LSTM-CRF [6].

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

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

[7]

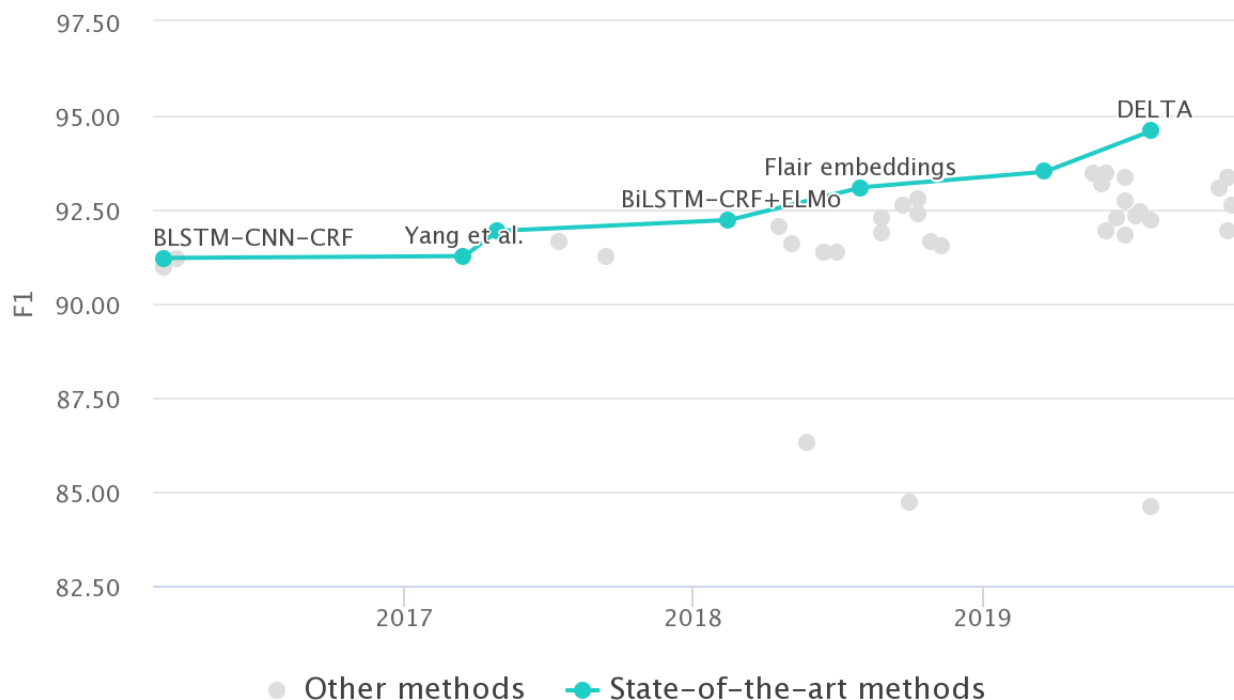
Статья от Ryan Cotterell и Kevin Duh, где представлена модель Conditional Random Fields (CRF) и представлены способы её улучшения с помощью таких надстроек, как обучение модели на языках с большими ресурсами, а потом применение её к языку из того же семейства, но с меньшими ресурсами. В данной работе рассматривались семья индоевропейских, ветви: романская, германская, славянская, индоарийская; и семья австронезийских, ветвь: филиппинская.

2.1.2 A Neural Layered Model for Nested Named Entity Recognition

[8]

Статья от Meizhi Ju, Makoto Miwa и Sophia Ananiadou. В данной работе представляется модель Layered-BiLSTM-CRF, которую я использовала в своей работе. Исследователи представляют модель, которая работает с «наслоенными» именованными сущностями, т.е. когда одна

Рис. 2.1: Лучшие модели для задачи распознавания именованных сущностей на датасете CoNLL 2003



именованная сущность частично или полностью входит в другую именованную сущность. Используются последовательно идущие плоские слои, слой состоит из BiLSTM и поверх него один слой CRF.

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

[9]

Данная статья от Google AI Language, которая на текущий момент является лучшей текущей моделью практически на всех популярных бенчмарках обработки естественного языка. Она включает в себя очень много возможностей для различных задач обработки естественных языков, которые на данный момент пользуются большой популярностью. Для конкретно моей задачи очень хорошо подходит претренированная модель bert-base-multilingual-cased, в обучающие данные которой входила также и Википедия на татарском языке.

Несмотря на популярность BERT, даже дообучать данную модель довольно сложно из-за её размеров. Спасибо Высшей Школе Экономики за возможность использования кластера для обучения моделей для выпускной квалификационной работы.

BERT — это очень большая модель, которая, однако, не решает задачи распознавания именованных сущностей сама по себе, поэтому я воспользовалась библиотекой transformers от huggingface [10]

2.1.4 Huggingface Transformers

Данная библиотека предоставляет возможность использовать различные лучшие на данный момент модели (не только BERT, но и многие другие) для решения различных задач обработки естественного языка, в том числе и распознавания именованных сущностей, что и является моей

задачей. Их [репозиторий на github](#) содержит множество [примеров](#) для удобного использования их библиотеки.

2.2 Стандартные подходы к разметке данных

Стандартным (самой распространённым) форматом разметки для корпусов текста для задачи распознавания именованных сущностей является разметка IOB (сокр. от Inside–outside–beginning), она же BIO. Она была представлена в работе Text Chunking using Transformation-Based Learning [11]. Данный формат имеет три префикса:

- В префикс перед тегом указывает, что тег находится в начале чанка (в нашем случае именованной сущности)
- I префикс перед тегом указывает, что тег находится в продолжении чанка.
- O префикс указывает, что данное слово не относится ни к какому чанку.

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

Иван Петров проживает в Российской Федерации .

Что-то я не знаю, что ещё сюда написать.

2.3 Работы, связанные с распознаванием 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 языкам малой языковой группы.

В отличие от моего случая, где существует релевантная работа, поднимавшая раньше тему моей работы, 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 распознавания 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 распознаванием именованных сущностей в татарском языке

При поиске корпусов на татарском языке я нашла корпус Туган Тел — работу Невзоровой и др. [12]. К сожалению, других релевантных данных найдено не было.

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 в татарском языке, 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 существует проект разработки электронного корпуса, который также включает в себя автоматическую разметку 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, снабжены автоматической морфологической разметкой, которая включает в себя информацию о части речи и грамматической характеристики словоформы. Морфологическая разметка текстов корпуса выполняется автоматически с использованием модуля двухуровневого морфологического анализа татарского языка, реализованного в программном инструментарии РС-КИММО, с чем связан ряд проблем в использовании данного корпуса, о которых я скажу в основной части работы. На декабрь 2019 года в корпусе 194 млн. словоформ.

Проведя обзор литературы я нашла только одну статью, релевантную к моей теме.

2.4.2 Named Entity Recognition in Tatar: Corpus-Based Algorithm

Самая близкая к моей работе это статья «Named Entity Recognition in Tatar: Corpus-Based Algorithm» от 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» [12], используя следующие категории: книги, рестораны, фильмы, журналы, компании, аэропорты, корпорации, языки, колледжи, университеты, школы, магазины, музеи и больницы.

1. Используемые данные

Исследователи использовали корпус Туган тел [12], о котором я говорила выше.

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, причём количество итераций определяется пользователем.

Нулевым шагом алгоритма является выборка по поисковому запросу. Запрос может представлять собой форму слова, лемму, фразу или поиск по морфологическим параметрам. Выборка представляет собой набор биграмм и их количество вхождений в текст. В биграмме одно слово является запросом, а второе ищется по корпусу с (опционально) морфологическими параметрами. Далее полученный список из запроса просматривается глазами и из него убирается мусор. 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; добавляется слово может слева или справа, данный параметр выбирается пользователем. Полученный список триграмм отсортировывается по частоте вхождений в корпус и в выборке остаются только самые частотные (например, первые 95%, в статье этот параметр обычно был равен 80%). Порог отсечения (в статье он называется

«индекс покрытия», «covering index») более частотных вхождений также выбирается пользователем. 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 (четвертое слово может добавляться справа или слева), 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.

3. Окончание алгоритма:

Существует такое понятие как «точность сравнения» («accuracy of matching») P , которое задаётся пользователем в процентах. Если частота n -граммы меньше P от количества найденных $(n + 1)$ -грамм, то алгоритм прекращает увеличивать длину именованной сущности, иначе алгоритм переходит на следующую итерацию. Таким образом, в финальный результат входят самые стабильные n -граммы разной длины, включая результаты поиска изначального поискового запроса.

Стоит отметить, что все сущности, выделенные на нулевом шаге алгоритма, так или иначе считаются именованными сущностями; вопрос только в том, сколько слов справа или слева к этой именованной сущности 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[12], 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).

$$Q = (Q_1, Q_2, L, R, C, P)$$

4. Эксперименты:

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

Также в данной статье очень интересный способ оценки результатов. Стандартные accuracy, 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, «требуется дополнительной очистки, тогда станет именованной сущностью», «требуется 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 2.1. Данное оценивание не позволяет мне сравниваться с результатами Невзоровой и др., так как в моей работе поставлена другая задача.

TODO номер таблицы?

2.5 Выводы

В области распознавания именованных сущностей написано много статей и изобретено много моделей, показывающих хорошие результаты на распространённых языках. Существуют так же работы по теме распознавания именованных сущностей для малоресурсных языков. Академия

Class named entity	of	Correct	Require filtering	Require expansion	Correct names of subclasses	Names of subclasses that require filtering	Incorrect	Total
Names ministries	of	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 2.1: Таблица 3 из статьи [1]

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 сравнюсь с результатами алгоритма Невзоровой и др.

3 Методология

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

1. Получение и разметка данных
2. Обучение и тюнинг моделей
3. Сравнение результатов

Но в течение работы по нескольким причинам были внесены корректировки. Во-первых, как я упоминала ранее в обзоре литературы, с представленными результатами в статье Невзоровой невозможно сравниваться, поскольку цели моей и их работ различаются. Во-вторых, качество полученных данных оказалось не лучшим из возможных, а алгоритм Невзоровой, разработанный как раз для разметки данных, мог бы улучшить имеющийся корпус, используемый для обучения моделей. Как следствие, было принято решение воспроизвести алгоритм из статьи Невзоровой насколько это возможно и воспользоваться полученными результатами.

1. Получение и разметка данных

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

Части речи <ul style="list-style-type: none"> <input type="checkbox"/> Существительное <input type="checkbox"/> Прилагательное <input type="checkbox"/> Глагол <input type="checkbox"/> Наречие <input type="checkbox"/> Числительное <input type="checkbox"/> Местоимение <input type="checkbox"/> Союз <input type="checkbox"/> Послелог <input type="checkbox"/> Междометие <input type="checkbox"/> Модальное слово <input type="checkbox"/> Звукоподражательное слово 	Падежи <ul style="list-style-type: none"> <input type="checkbox"/> Именительный <input type="checkbox"/> Родительный (генитив) <input type="checkbox"/> Направительный (директив) <input type="checkbox"/> Направительный с огранич. знач. <input type="checkbox"/> Винительный (аккузатив) <input type="checkbox"/> Исходный (аблатив) <input type="checkbox"/> Местно-временной (локатив) 	Залог <ul style="list-style-type: none"> <input type="checkbox"/> Действительный (основной) <input type="checkbox"/> Страдательный (пассив) <input type="checkbox"/> Возвратный (рефлексив) <input type="checkbox"/> Понудительный (каузатив) <input type="checkbox"/> Взаимно-совместный (реципрок) 	Формы императива <ul style="list-style-type: none"> <input type="checkbox"/> Императив 1 л. (гортатив) ед. ч. <input type="checkbox"/> Императив 1 л. (гортатив) мн. ч. <input type="checkbox"/> Императив 2 л. ед. ч. <input type="checkbox"/> Императив 2 л. мн. ч. <input type="checkbox"/> Императив 3 л. (юссив) ед. ч. <input type="checkbox"/> Императив 3 л. (юссив) мн. ч. <input type="checkbox"/> Просит. имп. (прекатив) на -чы <input type="checkbox"/> Просит. имп. (прекатив) на -сана
Время <ul style="list-style-type: none"> <input type="checkbox"/> Настоящее <input type="checkbox"/> Прош. категорич. <input type="checkbox"/> Прош. результативное (перфект) <input type="checkbox"/> Буд. категорич. <input type="checkbox"/> Буд. неопред. <input type="checkbox"/> Отриц. форма буд. неопред. 	Число <ul style="list-style-type: none"> <input type="checkbox"/> Единственное <input type="checkbox"/> Множественное 	Формы поссесива <ul style="list-style-type: none"> <input type="checkbox"/> 1 л., ед. ч. <input type="checkbox"/> 1 л., мн. ч. <input type="checkbox"/> 2 л., ед. ч. <input type="checkbox"/> 2 л., мн. ч. <input type="checkbox"/> 3 л., ед. ч. <input type="checkbox"/> 3 л., мн. ч. 	Разряды числительных <ul style="list-style-type: none"> <input type="checkbox"/> Собирательное <input type="checkbox"/> Порядковое <input type="checkbox"/> Разделительное <input type="checkbox"/> Приблизительного счета
Элементы словообразования <ul style="list-style-type: none"> <input type="checkbox"/> Уменьшит. форма <input type="checkbox"/> Ласкат. форма <input type="checkbox"/> Лицо деятеля по роду занятий <input type="checkbox"/> Абстрактное сущ. <input type="checkbox"/> Мера <input type="checkbox"/> Распределение 	Лицо <ul style="list-style-type: none"> <input type="checkbox"/> 1 л., ед. ч. <input type="checkbox"/> 1 л., мн. ч. <input type="checkbox"/> 2 л., ед. ч. <input type="checkbox"/> 2 л., мн. ч. <input type="checkbox"/> 3 л., ед. ч. <input type="checkbox"/> 3 л., мн. ч. 	Деепричастия <ul style="list-style-type: none"> <input type="checkbox"/> Сопутствующего действия <input type="checkbox"/> Сопутствующего действия (Отриц.) <input type="checkbox"/> Деепричастие на -гач <input type="checkbox"/> Деепричастие на -ганчы 	Общий вопрос <ul style="list-style-type: none"> <input type="checkbox"/> Вопросит., неопред. <input type="checkbox"/> Вопросит. формана-мыни <input type="checkbox"/> Вероятн., предположит. <input type="checkbox"/> Уподобление 1 <input type="checkbox"/> Уподобление 2 <input type="checkbox"/> Уподобление 3
Имена действия <ul style="list-style-type: none"> <input type="checkbox"/> Имя действия на -у <input type="checkbox"/> Имя действия на -ш (-ыш, -еш) 	Причастия <ul style="list-style-type: none"> <input type="checkbox"/> Настоящего времени <input type="checkbox"/> Прошедшего времени <input type="checkbox"/> Будущего времени <input type="checkbox"/> Регулярно совершаемого действия 	Модальные формы глаг. <ul style="list-style-type: none"> <input type="checkbox"/> Условная модальность (кондиционалис) <input type="checkbox"/> Необходимость <input type="checkbox"/> Возможность <input type="checkbox"/> Намерение <input type="checkbox"/> Предостережение 	Атрибутивные формы <ul style="list-style-type: none"> <input type="checkbox"/> Атрибутив на -лы (мунитатив) <input type="checkbox"/> Атрибутив на -сыз (Абессив) <input type="checkbox"/> Локативный атрибутив <input type="checkbox"/> Генитивный атрибутив
	Инфинитивы <ul style="list-style-type: none"> <input type="checkbox"/> Инфинитив на -ырга <input type="checkbox"/> Инфинитив на -мак 	Способы глаг. действия <ul style="list-style-type: none"> <input type="checkbox"/> на -гала <input type="checkbox"/> Раритив на -ыштыр 	Сравнит. степень <ul style="list-style-type: none"> <input type="checkbox"/> Сравнит. степень
	Аспект глагола <ul style="list-style-type: none"> <input type="checkbox"/> Отрицание 		

- Обучение и тюнинг моделей
- Воспроизведение статьи Невзоровой
- Разметка данных с помощью алгоритма Невзоровой
- Обучение и тюнинг моделей
- Сравнение результатов

4 Получение и разметка данных

4.1 Туган Тел

Обзор литературы показал, что существует корпус татарских текстов Туган Тел[12]. Данный корпус имеет также свою 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 нет.

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 корпусу.

Корпус представляет из себя .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

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 постоянно 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: Тахир меня зовут.

с помощью программного инструментария PC-KIMMO. Разметка выглядит следующим образом (см. рис 4.2). На нечетной строке написано слово, на следующей — разметка слова. Знаки препинания тоже являются «словами».

Поскольку я не использовала разметку никаким образом, кроме как для первой итерации выделения именованных сущностей, заострять внимание я не ней не буду.

TODO нужно ли вставлять описание тегов морфоанализатора?

В тегах морфоанализатора также присутствует тег PROP, который обозначает имя собственное. Сложно сказать, есть ли какая-то консистентность, но для первой итерации было решено применять этот тег в качестве именованной сущности. Как можно заметить в примере на рис. 4.3, в корпусе имена собственные иногда бывают с маленькой буквы, что говорит о том, что данные содержат в том числе и ошибки. Всего в текстах 30 753 824 слов, из них 534 514 это автоматически размеченные именованные сущности, что составляет 1,738% от всех слов.

4.2 Татарская Википедия

Кроме корпуса «Туган Тел» другого большого количества текстов, собранных в одном месте, найдено не было, поэтому было принято решение скачать википедию на татарском языке.

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,

позже статьи писались 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 1927-1939 — 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. На текущий 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: Статъя «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-тексты үзгәртү]

2006 елда Чулман елга бассейны нәм аның белән бәйлә жир 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 хөтлө
 - 10.01.01.013 — Ык башлангычыннан тамагыне хөтлө
 - 10.01.02.001 — Агыйдел (елга)сы башлангычыннан
 - 10.01.03 — Нократ^[2]
 - 10.01.03.001 — Чүпче башлангычыннан тамагыне хөтлө
 - 10.01.03.002 — Нократ башлангычыннан Вятка шөһәрәне хөтлө

Искәрмәләр [үзгәртү | вики-тексты үзгәртү]

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



Рис. 4.5: Пример сгенерированных статей из Википедии

×
Эзләү

Киңәйтелгән эзләү:

×

Эзләү:

×

Өлеге вики-проектта «Бассейны» исемле бит ясарга! Шулай ук, эзләү ярдәмендә табылган битләргә 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** нәм 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** нәм 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** нәм 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** нәм аның белән

1 Кб (77 сүз) - 2 апр 2013, 22:00

Также важно отметить, что википедия представляет из себя набор статей, написанных в академическом стиле, что не вполне соответствует реальной человеческой речи; в этом аспекте Туган Тел гораздо лучше. В Википедии в том числе существуют 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, в русской википедии они тоже сгенерированы автоматически.

4.3 Разметка данных для обучения

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

Во второй итерации использовался воспроизведенный алгоритм Невзоровой, в качестве «данных» для которого применялась Википедия. На данных из Википедии был получен список именованных сущностей по классам PER (персона), LOC (географический объект), ORG (организация) и MISC (всё остальное, но в основном названия языков). С помощью полученного списка была размечена Википедия, 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

Никакая автоматическая разметка не может быть настолько же идеальной, насколько ручная. Поэтому для оценивания воспроизведенного алгоритма Невзоровой и обученных моделей было принято решение разметить некоторое количество предложений самостоятельно, в силу имеющихся знаний татарского языка. Предложения выбирались 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 датасет golden-bio.txt, основанный на предложениях из корпуса Туган Тел (350 предложений).

4.5 Проблемы с разметкой данных

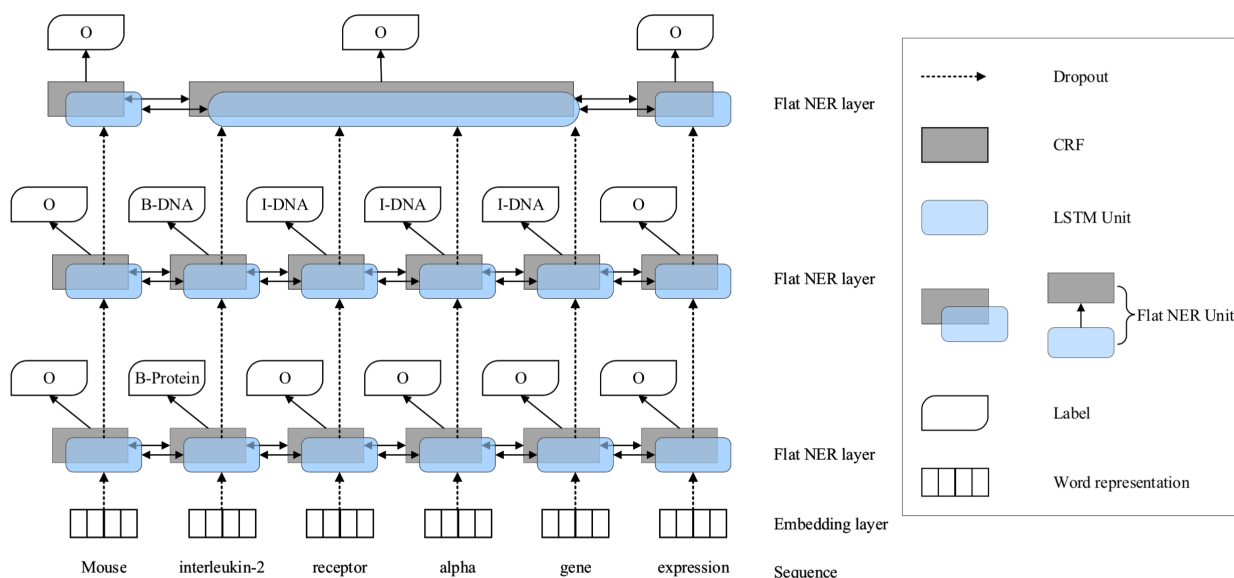
Как было описано и в статье Невзоровой, где исследователи глазами просматривали полученные результаты, отсеивали некорректные и улучшали свой алгоритм с помощью фильтров, так и в моей работе разметка данных происходила итеративно. Например, генерация статей про бассейны в Википедии была как раз выявлена 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 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.

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» [8]. Она использует разметку BIO, как и многие другие модели для распознавания именованных сущностей. Архитектура модели изображена на рис. ??

Рис. 5.1: Архитектура модели BiLSTM-CRF, рис. из статьи [8]



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

Первая итерация на данных Туган Тел, где тег PROP стал соответствовать тегу B-PER.

Результат обучения, в данных два тега: O и B-PER.

TODO Убрать последние 3 столбца?

Category	Precision	Recall	F-score	Predicts	Golds	Correct
PER	99.768	90.727	95.033	2589	2847	2583

Была сделана демонстрация работы модели в виде сайта, запускающегося на локальной машине, но самые простые примеры не из тестового набора выявили большие несовершенства данной модели, поэтому в итоге было принято решение в целом отказаться от её использования. Как можно понять, цифры выше показывают возможность 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.

Далее сегодня я обучу модель на размеченной с помощью алгоритма Невзоровой Википедии и можно будет сравниться со всеми остальными моделями.

5.2 BERT

[9] 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 [10] и претренированная модель bert-base-multilingual-cased, которая обучена на чувствительных к регистру данных из 104-х крупнейших 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).

Опять возникла проблема с тем, что корпус слишком большой (вот уж иронично, что язык считается малоресурсным), но не с тем, что локальная машина не тянула обучение, а с тем, что библиотека torch использует библиотеку pickle для сохранения признаков, а библиотека pickle не сохраняет данных больше, чем на 4GB, а у меня, по моим прикидкам, признаков должно было получиться на 13GB. В итоге пришлось ограничиться 1/3 от всех данных.

Первая итерация на данных Туган Тел, где тег PROP стал соответствовать тегу B-PER.

Результат обучения, в данных два тега: O и B-PER.

Precision	Recall	F-score
97.447	94.585	95.995

Один в один повторение истории с BiLSTM-CRF — сами по себе цифры очень многообещающие, модель обучилась прекрасно, но на деле они показывают всего лишь затренированность модели на плохих начальных данных, которая в итоге не работает на простых примерах, придуманных из головы. Было принято решение от этой модели отказаться.

Вторая итерация была на википедии, размеченной с помощью списка именованных сущностей, полученных с помощью алгоритма Невзоровой. В корпусе википедии 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

Сделать метрики на всех категориях!!!

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 распознавать именованные сущности, необходимо проверить предсказания на «золотых» данных, размеченных вручную, про которые мы точно знаем, что они правильные.

6 Воспроизведение алгоритма из статьи Невзоровой

Было принято решение воспроизвести алгоритм из статьи Невзоровой и др. [1] по двум причинам.

1. Чтобы была возможность сравниваться с результатами предыдущих работ.
2. Чтобы разметить данные эффективно и, по возможности, хорошо.

Обе цели были достигнуты и теперь я имею возможность рассказать вам о проблемах, с которыми я столкнулась во время работы и результатах, полученных в итоге.

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 менеджер-системе Туган Тел [12], поэтому воспроизвести алгоритм в точности не удалось. Попытка приблизиться к результату состояла в том, чтобы использовать в качестве начального слова все возможные словоформы слова, а не только слово само, т.е. частично сделать работу морфоанализатора. Однако делать запросы по каким-то специфичным параметром, конечно, в моём случае невозможно.

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

Ручная чистка состояла в просмотре полученных именованных сущностей и по-возможности исправление не точечно, а «широкими мазками» — добавлением новых начальных слов для поискового запроса или удаление всех очевидно некорректных сущностей одним вызовом команды `gter`. Например, по слову «министрлыгы» (министерство) появилось множество сущностей, которые начинались с союза «Һәм» (и). Это некорректные сущности, но удалить их все 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), «авылы» (село), «өлкәсе» (область) и др., организации: «академиясы» (академия), «идарә» (администрация), «институты» (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 тысяч именованных сущностей, который доступен на моём github. Он пригодится людям, которые будут продолжать работу в данном направлении для разметки собственных корпусов или распознавания именованных сущностей.

7 Демонстрация полученных результатов

Была сделана html-страница, с помощью которой можно проверить модель на работоспособность.

Рис. 7.1: Старший научный сотрудник исторического института имени Мержани

Tatar NER (BERT-based)

Text in Tatar: Мәрҗани исемендәге тарих институтында өлкән фәнни хезмәткәр

Мәрҗани исемендәге тарих институтында өлкән фәнни хезмәткәр .

Some examples:

- [Исемем минем акбай](#)
- [Мәрҗани исемендәге тарих институтында өлкән фәнни хезмәткәр.](#)
- [госманов белән берлектә «Шура» журналының библиографик күрсәткече төзеп бастырды \(2000\).](#)
- [тинчурин ижатында символлар әһәмиятле урын алып тора.](#)

Всё распознано корректно, ORG

Рис. 7.2: Зовут меня Ксения

Tatar NER (BERT-based)

Text in Tatar:

Исемем минем **Ксения** .

Some examples:

- [Исемем минем акбай](#)
- [Мәржани исемендәге тарих институтында өлкән фәнни хезмәткәр.](#)
- [госманов белән берлектә «Шура» журналының библиографик күрсәткече төзеп бастырды \(2000\).](#)
- [тинчурин ижатында символлар эхәмиятле урын алып тора.](#)

Всё распознано корректно, PER

Рис. 7.3: Он решил изучать русский язык

Tatar NER (BERT-based)

Text in Tatar:

Ул карарны **рус телендә** укыды .

Some examples:

- [Исемем минем акбай](#)
- [Мәржани исемендәге тарих институтында өлкән фәнни хезмәткәр.](#)
- [госманов белән берлектә «Шура» журналының библиографик күрсәткече төзеп бастырды \(2000\).](#)
- [тинчурин ижатында символлар эхәмиятле урын алып тора.](#)

Всё распознано корректно, MISC

Рис. 7.4: Празднование состоится в селе Асан Дюртюлинского района

Tatar NER (BERT-based)

Text in Tatar:

Бәйрәм **Дүртөйле** районының әсән **авылында** була .

Some examples:

- [Исемем минем акбай](#)
- [Мәржани исемендәге тарих институтында өлкән фәнни хезмәткәр.](#)
- [госманов белән берлектә «Шура» журналының библиографик күрсәткече төзеп бастырды \(2000\).](#)
- [тинчурин ижатында символлар эхәмиятле урын алып тора.](#)

Не распознано имя села, хотя само понятие «село» распознано как LOC. «Дюртюлиский» ошибочно распознано как имя.

8 Сравнение результатов

Отдельной задачей стоял вопрос, как сравниться с предыдущими результатами в данной области. Было принято решение разметить некоторое количество предложений вручную, чтобы иметь «золотые» данные, про которые было бы известно, что шанс неправильных меток 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 запущен на этих данных. Теперь полученные результаты допустимо сравнивать между собой, поскольку они «в равном положении» и выполняют одинаковую задачу.

Результаты выполнения алгоритма Невзоровой на «золотых» данных приведены в таблице 8.1

TODO Выделить главное! В этой таблице столько цифр, что замыливается всё.

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
micro avg	0.53	0.51	0.52	481
macro avg	0.52	0.51	0.48	481

Таблица 8.1: Результаты алгоритма Невзоровой

Результаты тестирования модели BERT на «золотых» данных приведены в таблице 8.2

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
micro avg	0.53	0.49	0.51	481
macro avg	0.52	0.49	0.47	481

Таблица 8.2: Результаты модели BERT

Результаты тестирования модели BiLSTM-CRF на «золотых» данных приведены в таблице 8.3 пока таблицы нет, но скоро будет, и моё предположение таково, что он будет ещё хуже, чем BERT. Но даже если сравним, то точно не лучше, чем алгоритм Невзоровой.

Тут будет таблица

Таблица 8.3: Результаты модели BERT

Как видно из цифр выше, разметка данных с помощью алгоритма 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).

9 Выводы

1. Воспроизведен алгоритм из статьи Невзоровой и др., он показал свою работоспособность и его можно использовать в дальнейшем.
2. Получен массив именованных сущностей и два размеченных корпуса, 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. Текущие лучшие модели действительно работают хорошо и на татарском языке, обучаясь на данных настолько, насколько они 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, модели хорошо обучаются и на имеющемся количестве входных данных.

10 Заключение

Проведена большая хорошая работа и по её окончании получены удовлетворительные результаты. Была 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 именованных сущностей, которые можно использовать в дальнейших работах по данной теме. Были обучены модели BiLSTM-CRF и BERT, которые показали результаты не хуже, чем предыдущая работа в данной области.

В дальнейшем будет сотрудничать с Академией наук Республики Татарстан и дальше двигать направление распознавания именованных сущностей, пробовать новые модели не только распознавания, 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 др.;
2. Разработать удобную систему для ручной разметки данных и привлекать к разметке людей, получая более точные результаты;
3. Попробовать другие лучшие текущие модели, конкретно попробовать сделать обёртку CRF над BERT, это скорее всего поможет решить проблему с I-тегами без B-тегов.

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

Список литературы

- [1] Olga Nevzorova Damir Mukhamedshin, Galieva Alfiya. Named Entity Recognition in Tatar: Corpus-Based Algorithm // semanticscholar. 2018.
- [2] DELTA: A DEep learning based Language Technology plAtform / Han, Kun, Chen [и др.] // arXiv e-prints. 2019. URL: <https://arxiv.org/abs/1908.01853>.
- [3] Cloze-driven Pretraining of Self-attention Networks / Alexei Baevski, Sergey Edunov, Yinhan Liu [и др.] // arXiv e-prints. 2019. URL: <https://arxiv.org/abs/1903.07785v1>.
- [4] GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling / Yijin Liu, Fandong Meng, Jinchao Zhang [и др.] // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.
- [5] Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition / Yufan Jiang, Chi Hu, Tong Xiao [и др.] // 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. C. 3585–3590. URL: <https://www.aclweb.org/anthology/D19-1367>.

- [6] Straková Jana, Straka Milan, Hajic Jan. 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. C. 5326–5331. URL: <https://www.aclweb.org/anthology/P19-1527>.
- [7] Cotterell Ryan, Duh Kevin. 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. C. 91–96. URL: <https://www.aclweb.org/anthology/I17-2016>.
- [8] Ju Meizhi, Miwa Makoto, Ananiadou Sophia. 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. C. 1446–1459. URL: <https://www.aclweb.org/anthology/N18-1131>.
- [9] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee [и др.] // CoRR. 2018. T. abs/1810.04805. URL: <http://arxiv.org/abs/1810.04805>.
- [10] HuggingFace’s Transformers: State-of-the-art Natural Language Processing / Thomas Wolf, Lysandre Debut, Victor Sanh [и др.] // ArXiv. 2019. T. abs/1910.03771.
- [11] Ramshaw Lance A., Marcus Mitchell P. Text Chunking using Transformation-Based Learning // CoRR. 1995. T. cmp-lg/9505040. URL: <http://arxiv.org/abs/cmp-lg/9505040>.
- [12] Nevzorova O. Mukhamedshin D., R. Gataullin. 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. C. 108–122.