Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра математического анализа

Компьютерная обработка ограниченных естественных языков

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 219 группы

направления 01.04.02 Прикладная математика и информатика

механико-математического факультета

Лазарева Андрея Вадимовича

Научный руководитель
профессор, д.ф-м.н., профессор _____ С.Ф. Лукомский

Зав. кафедрой
д.ф-м.н., профессор _____ Д.В. Прохоров

Саратов 2018

Введение. Тема моей магистерской работы – «Компьютерная обработка ограниченных естественных языков». Компьютерная обработка естественных языков – междисциплинарная научная дисциплина, использующая методы компьютерных наук, машинного обучения и вычислительной лингвистики для решения задач, касающихся взаимодействия компьютера с естественными языками.

Ограниченный естественный язык – это версия естественного языка, полученный ограничением в использовании грамматики, терминологии и речевых оборотов с тем, чтобы избавиться от его многозначности и сложности. В моей дипломной работе ограниченения на естественный язык накладывала выбранная мной задача, относящихся к области обработки естественного языка – распознавание именованных сущностей в тексте. Таким образом, к обрабатываемому тексту на естественном языке применяется ограничение – он должен содержать именованные сущности.

Что такое именованная сущность? Именованная сущность – это объект реального мира, который может быть определен посредством имени или названия, например, человек, организация, географическое именование. Данный термин впервые прозвучал на конференции посвященной задачам извлечения информации «Message Understanding Conference» в 1995 году. На конференции обсуждались вопросы построения структурированных данных из неструктурированных или слабоструктурированных машиночитаемых текстов. Позднее, уже в 2003 году, на конференции «Conference of Computational Language Learning» была поставлена задача распознавания именованных сущностей и предложена их классификация. Задача распознавания подразумевает обработку текста на естественном языке и присвоению каждому слову в тексте тега, обозначающего определенную именованную сущность.

Ознакомимся с классификацией именованых сущностей и продемонстрируем обозначающие их теги.

- человек: тег «PER» (англ. person);
- организация: тег «ORG» (англ. organization);
- географическое наименование: тег «LOC» (англ. location);
- иные возможные объекты: тег «MISC» (англ. miscellanious – разное);
- слова, не являющихся именованными сущностями: тег «O» (англ. other).

Каждый из представленных выше тегов делится на начальную и внутреннюю часть, ведь сущность может состоять не из одного слова. Первому тегу сущности приставляется префикс «B-» (англ. beginning), а следующим за ним тег «I-» (англ. inside) Так, например, слова, представляющие сущность «человек» «Alexandr Sergeevich Pushkin» должны быть обозначены тегами «B-PER I-PER I-PER».

Задача распознавания именованных сущностей подразумевает создание программного обеспечения, способного обрабатывать текст на ограниченном естественном языке, выделять из него отдельные слова и присваивать каждому слову тег – т.е. выполнять задачу классификации слов в тексте по 9 категориям: «B-PER», «I-PER», «B-ORG», «I-ORG», «B-LOC», «I-LOC», «B-MISC», «I-MISC», «O»).

При выполнении магистерской работы мной был изучен существующий теорический материал, посвященный данной задаче, выделен один из наиболее перспективных подходов к ее решению и реализована программа, выполняющая это решение.

Основное содержание работы. Для компьютерного представления слов используются векторные представления, полученные с помощью алгоритма GloVe. В основе данного алгоритма лежит предпложение из

дистрибутивной семантики, что слова, часто используемые в одном контексте, имеют близкое значение. Алгоритм заключается в извлечении статистической информации о совместной встречаемости слов в тексте и построении на ее основе контекстных векторов – представлений слов. Под совместной встречаемостью понимается статистическая информация о том, насколько часто одно слово встречается в одном контексте с другим словом. Под контекстом понимается 10 слов слева и 10 слов справа от выбранного слова в тексте. Весь текст, преобразованный в векторы – это векторное пространство, а евклидово расстояние между отдельными векторами – метод измерения семантического расстояния (близости значения) между отдельными словами.

Повторим еще раз постановку задачи: имея в качестве исходных данных текст на английском языке, необходимо, разбить его на отдельные слова, и каждое из них, необходимо классифицировать по 9 признакам «B-PER», «I-PER», «B-ORG», «I-ORG», «B-LOC», «I-LOC», «B-MISC», «I-MISC», «O». Функция классификации слов далее именуется моделью. Архитектуру модели можно разбить на три главные составляющие:

1. Создание векторных представлений слов, кодирующих информацию о строении слов. Для каждого слова необходимо построить его векторное представление $w \in \mathbb{R}^n$. Для создания данных векторных представлений используется алгоритм GloVe, который является свободным программным обеспечением. Так же, строятся собственные векторные представления на основе регистра слов. Для этого используется нейронная сеть с долговременной и краткосрочной памятью. Далее, векторные представления, созданнные с помощью алгоритма GloVe конкатенируются с представлениями, созданными на основе контекста и получаются векторные представления, кодирующие строение слов.
2. Создание векторных представлений, кодирующие информацию о контексте слов. Для каждого слова в контексте, необходимо построить

представление $h \in \mathbb{R}^k$, которое будет кодировать в числовом виде информацию о контексте слов. Для этого, так же, используется нейронная сеть с долговременной и краткосрочной памятью.

3. Финальный шаг работы модели – совершение предположения путем вычисления оценки. Векторное представление каждого слова в предложении используется, чтобы сделать предположение о принадлежности его к одной из 9 категорий «B-PER», «I-PER», «B-ORG», «I-ORG», «B-LOC», «I-LOC», «B-MISC», «I-MISC», «O».

Используемые для решения задачи векторные представления для каждого слова $w \in \mathbb{R}^n$ строятся из векторов, полученных с помощью системы GloVe $w_{gl} \in \mathbb{R}^{d_1}$, а так же, из векторных представлений, построенных на основе регистра слов. Использование такого очевидного свойства как регистр, является очень полезным, т.к. все именованные сущности в тексте начинаются с большой буквы. Конечно, нельзя не отметить, что использование данного свойства является и ограничением, однако, поскольку, модель работат с ограниченным текстом на естественном языке, использование этого принципа я считаю допустимым. Представления слов, построенные с использованием этого свойства обозначаются как $w_{reg} \in \mathbb{R}^{d_2}$. Для построения данных представлений используется многослойный перцептрон, который автоматически извлекает информацию о регистре слов.

Для решения задачи, так же, необходимо построить собственные векторные представления слов, включащие векторные представления, полученные с помощью алгоритма GloVe, которые обозначаются как $w_{gl} \in \mathbb{R}^{d_1}$, а так же, векторные представления, кодирующие информацию о регистре слова. Финальное векторное представление обозначается как $w \in \mathbb{R}^n$, и содержит информацию о значении, строении слова и его регистре. Собственные векторные представления строятся путем конкатенации векторов из системы GloVe $w_{gl} \in \mathbb{R}^{d_1}$ и векторов, содержащих информацию о

регистре слов $w_{reg} \in \mathbb{R}^{d_2}$. Поскольку, ранее уже было сказано о том, что представляют собой векторы слов в модели GloVe, расскажем о том, будут строиться векторы, кодирующие информацию о регистре слов.

Каждый знак c_i слова $w=[c_{1,\dots},c_p]$ (буквы верхнего и нижнего регистра различаются) ассоциируется с вектором $c_i \in \mathbb{R}^{d_3}$. Данные векторы подаются в LSTM сеть. Ее последние состояния конкатенируются и получаются векторы $w_{reg} \in \mathbb{R}^{d_2}$. Полученные векторы кодируют информацию о строении слов. Затем, векторы w_{reg} и w_{gl} так же, конкатенируются для того, чтобы получить векторы представляющие отдельные слова $w=[w_{gl},w_{reg}] \in \mathbb{R}^n$, где $n=d_1+d_2$.

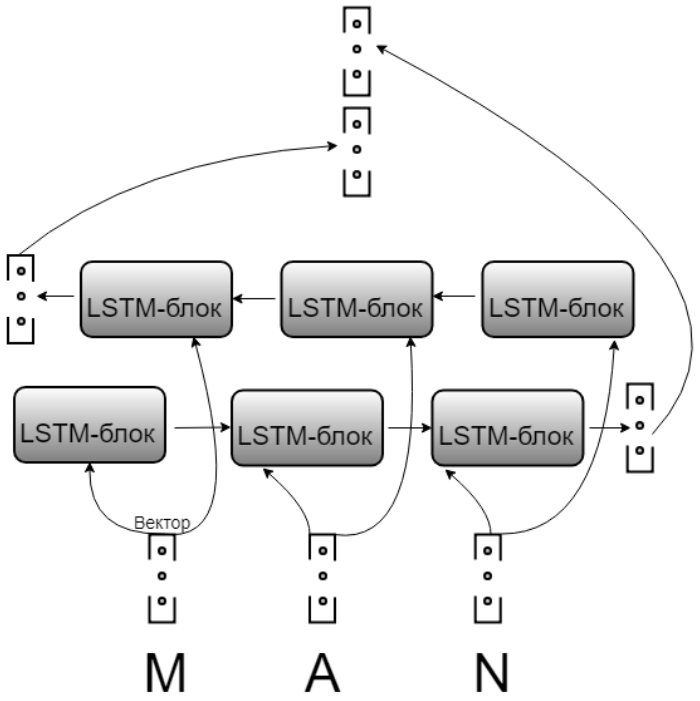

Рисунок 1. Схема построения векторных представлений,
кодирующих строение словс помощью двунаправленной LSTM сети

Итак, были построены векторные представления слов w. На последовательности данных векторов обучается bi-LSTM сеть. Векторы, получившиеся из двух ее скрытых состояний конкатенируются в векторы $h \in \mathbb{R}^k$. Таким образом, имея на входе последовальность векторов слов

$w_{1,\dots},w_m \in \mathbb{R}^n$ на выходе имеется последовательность векторов $h_{1,\dots},h_m \in \mathbb{R}^k$. В то время как w_t содержит информацию о строении слова, векторы h_t, так же, содержат информацию об их контексте.

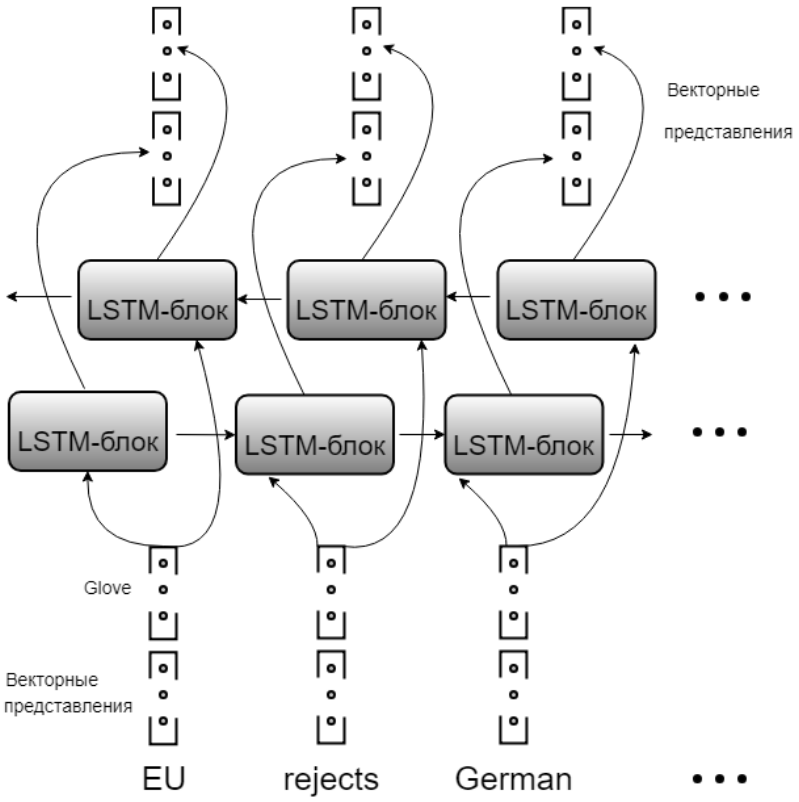

Рисунок 2. Схема построения векторных представлений, кодирующих контекст слов с помощью bi-LSTM сети

На последнем шаге вычисляется оценка для тега, которая показывает вероятность того что он подходит для данного слова. Каждое слово w ассоциируется с вектором h, который кодирует информацию о значении слова, его регистре и контексте. Данный вектор используется для того чтобы сделать предположение. На данном шаге используется многослойный перцептрон, чтобы получить вектор, каждый элемент которого соответствует числовому значению вероятности для каждого тега.

Итак, имеется 9 классов (тегов). Создается матрица векторных представлений слов $W \in \mathbb{R}^{9\times k}$ и классов $b \in \mathbb{R}^9$ и вычисляется вектор

$$s = \mathbb{R}^9 = W \cdot h + b$$

В полученном векторе s, каждый его элемент $s[i]$ понимается как значение класса i для слова w. После вычисления оценок, необходимо посчитать вероятность $P(y_{1,\dots},y_m)$ для последовательности тегов y_t и найти последовательность с наибольшим значением вероятности. Здесь y_t обозначает тег для слова с индексом t.

На данном шаге используется функция softmax. Значения $P(y_{1,\dots},y_m)$ нормализуются в вектор $p \in \mathbb{R}^9$, такой что

$$p[i] = \frac{e^{s[i]}}{\sum_{j=1}^{9} e^{s[j]}}$$

Тогда p_i может быть интерпретировано как вероятность того что слово пренадлежит к классу i, а сумма всех вероятностей равна 1. Вероятность $P(y)$ последовательности тегов равна произведению $\prod_{t=1}^{m} p_t[y_t]$.

Обучение модели производилось с помощью метода градиентного спуска. В качества функции стоимости была выбрана функция перекрестной энтропии. Таким образов, функция стоимости описывается формулой

$$H = -\log(P(\tilde{y}))$$

где $\tilde{y}$ – верная последовательность тегов и ее вероятность P.

Найденный в результате обучения класс с наибольшим значением вероятности и есть финальное предположение модели для данного слова.

Заключение. В ходе выполнения магистерской работы была написана программа на языке Python 3.6, выполняющая распознавание именованных сущностей. Ее структура, а так же, код, сопровожденный комментариями приведен в Приложении А к магистерской работе. Для оценки точности работы программы было проведено тестирование.

В качестве метрики эффективности алгоритма, может быть принята точность – доля входных данных по которым алгоритм выполнил верное решение.

$$Точность = \frac{P}{N}$$

где, P – число верных ответов, N – размер обучающей выборки.

В данном случае, метрика присваивает всем классам одинаковый вес, что может привести к некорректным результатам, если распределение данных по классам непропорционально, например, в задаче распознавания сущностей, большая часть слов обучающей выборки, имеет класс «O» (англ. other). В данном случае, у алгоритма больше информации об этом классе, и принимать для него больше правильных решений, а значит, оценки в рамках других классов могут пострадать. Чтобы избежать искажений в оценке, необходимо скорректировать формальную оценку качества.

«Точность» и «полнота» – метрики, используемые для оценки алгоритмов извлечения информации.

- Точность алгоритма в пределах класса – доля элементов выборки действительно относящихся к данному классу, относительно тех элементов выборки, которых алгоритм отнес к данному классу.
- Полнота алгоритма – доля отнесенных к данному классу элементов выборки, относительно всех элементов выборки относящихся к данному классу.

Значения точности и полноты рассчитывается с помощью таблицы контигентности, составляемой для каждого класса отдельно.

Класс		**Истинное решение**	
Решение алгоритма	Положительное	Положительное	Отрицательное
	Отрицательное	TP	FP
		FN	TN

Таблица 2. Таблица контигентности

Таблица содержит информацию о количестве верных и неверных решений, которые сделал алгоритм для данного класса

- TP — истино-положительное решение;
- TN — истино-отрицательное решение;
- FP — ложно-положительное решение;
- FN — ложно-отрицательное решение.

Точность и полнота вычисляются по формулам

- $Точность = \frac{TP}{TP+FP}$
- $Полнота = \frac{TP}{TP+FN}$

F-мера представляет собой гармоническое среднее между точностью и полнотой – она стремится к нулю, если точность или полнота стремится к нулю $F = 2\frac{Точность \times Полнота}{Точность + Полнота}$

В данном случае, *F*-мера одинаково убывает при уменьшении как точности и полноты. Так же, *F*-мера может быть рассчитана более точно, если установить коэффициент веса точности и полноте. В данном случае, формула принимает следующий вид

$$F = \left(\beta^2 + 1\right)\frac{Точность \times Полнота}{\beta^2 Точность + Полнота}$$

- β принимает значения в интервале $0 < \beta < 1$, если приоритет отдается точности;
- в случае, если прироритет отдается полноте, выбирается $\beta > 1$.

При $\beta = 1$, формула сводится к предыдущей сбалансированной мере, которая, так же, называется F_1 мера. Данная мера использовалась для оценки эффективности программы.

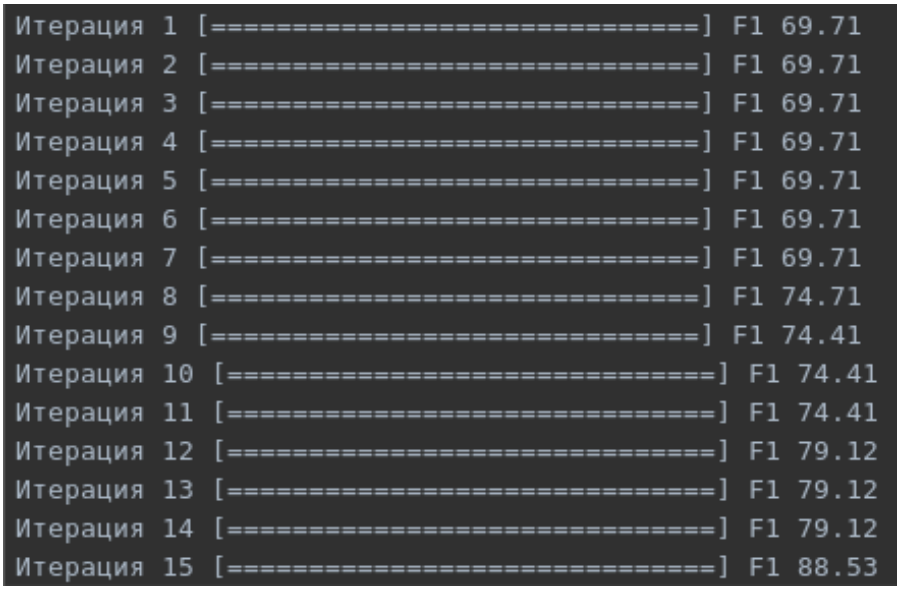

Рисунок 3. Значения F_1 для каждой итерации обучения программы

Значения F_1 на последних итерациях обучения превышают 88%, что можно считать хорошим показателем. На данный момент, наилучшее значение оценки качества алгоритмов в задаче распознавания именованных сущностей находятся в диапазоне 90-91%. Полученный результат можно считать достаточно высоким. Стоит отметить, что его можно улучшить с помощью дополнительной оптимизации алгоритма и увеличения обучающей выборки. Однако, это уже выходит за рамки данной работы.

В ходе выполнения магистерской работы были изучены существующие научные работы, посвященные алгоритмам распознавания именованных сущностей. По результатам изучения теоретического материала, была написана программа, демонстрирующая достаточно высокие показатели оценки качества.