Отчет

по проектной работе для итоговой аттестации по курсу Data Science, университет «Иннополис»

Название проекта: проект 4, извлечение именованных сущностей для русского языка.

Задача проекта: обучить и протестировать модель для извлечения именованных сущностей из текста. Провести анализ решения и альтернатив. Выбрать лучшую модель.

Источники данных:

http://bsnlp.cs.helsinki.fi/shared-task.html

https://multiconer.github.io

1. **Исследование входных данных.**

   Первый источник – SemEval 2022, 11 открытое задание по распознаванию именованных сущностей для 11 языков.

   Исходный набор данных содержит 15 300 размеченных предложений и 242 383 токена. Токены размечены с использованием схемы BIO (Beginning, Inside, Outside) для шести типов именованных сущностей: PER : Person, LOC : Location, GRP : Group, CORP : Corporation, PROD : Product, CW: Creative Work.

```
['# id 11b11e4f-73c6-4e3d-babd-0de83e450861\tdomain=train',
 'российская _ _ B-GRP',
 'экологическая _ _ I-GRP',
 'партия _ _ I-GRP',
 '«зелёные» _ _ I-GRP',
 '– _ _ O',
 'до _ _ O',
 'февраля _ _ O',
 '2012 _ _ O',
 'года _ _ O',
 ', _ _ O',
 'входила _ _ O',
 'в _ _ O',
 'состав _ _ O',
 'партии _ _ O',
 'ср _ _ O',
 '',
```

   При этом данные не сбалансированы, ввиду сильного преобладания не именованных токенов Outside (86%), что в принципе свойственно большинству предложений по многим тематикам.

```
c.most_common()
```

```
[('O', 208855),
 ('B-LOC', 4219),
 ('I-PER', 3982),
 ('B-PER', 3683),
 ('I-CW', 3399),
 ('B-CW', 3224),
 ('B-GRP', 2976),
 ('B-PROD', 2921),
 ('B-CORP', 2817),
 ('I-GRP', 2711),
 ('I-CORP', 1914),
 ('I-PROD', 942),
 ('I-LOC', 740),
 ('_ O', 1)]
```

Предварительный анализ показал, что данные уже очищены, разделены на тестовый и тренировочный наборы и не нуждаются в дополнительной очистке.

Второй источник – это 3 открытое задание по распознаванию, нормализации, классификации и межъязыковому связыванию именованных сущностей в славянских языках.

Исходный набор данных для русского языка содержит 3 191 токен в предложениях, находящихся в разных файлах. При этом для каждого предложения – есть 2 файла, в одном содержится разметка только именованных сущностей, в другом – все предложение целиком без разметки.

Схемы BILOU или BIO при разметке не используются. Разметка содержит пять типов именованных сущностей: человек, локация, организация, мероприятие, продукты (persons, locations, organizations, events, products).

Тематика предложений взята из новостных сводок и ограничена двумя основными темами: Brexit и суд над христианской девушкой Асией Биби в Пакистане, обвиняемой в богохульстве.

```
c.most_common()[:40]
```

```
[('Великобритании', 99),
 ('ЕС', 88),
 ('Brexit', 87),
 ('Дэвид Дэвис', 66),
 ('Бориса Джонсона', 63),
 ('Борис Джонсон', 60),
 ('Мэй', 58),
 ('Тереза Мэй', 58),
 ('Евросоюза', 57),
 ('Пакистане', 51),
 ('Терезы Мэй', 50),
 ('Пакистана', 46),
 ('Биби', 43),
 ('Джонсона', 42),
 ('Асии Биби', 40),
 ('Британии', 37),
 ('Джонсон', 36),
 ('Доминик Рааб', 32),
 ('Дэвис', 31),
 ('МИД Великобритании', 29),
 ('МИД', 27),
 ('Асия Биби', 26),
 ('Мухаммеда', 25),
 ('Дэвида Дэвиса', 25),
 ('Дэвиса', 25),
 ('Лондона', 24),
 ('Асию Биби', 22),
 ('Верховный суд Пакистана', 22),
 ('Евросоюзом', 22),
 ('Терезой Мэй', 21),
 ('Верховного суда', 19),
 ('Брюсселем', 18),
 ('Великобритания', 18),
 ('Бориса', 16),
 ('Рааб', 16),
 ('REGNUM', 15),
 ('США', 15),
 ('Доминика Рааба', 15),
 ('Даунинг-стрит', 15),
 ('Англии', 15)]
```

Ввиду ограниченности тем, малой выборки, отсутствия BIO разметки и необходимости дополнительных преобразований, с согласия наставника, было принято решение не использовать данный набор.

2. Исследование Python NLP библиотек.

Для работы системы в продуктивном режиме, необходимо разбивать на токены входной текст, и модель такого разбиения должна быть максимально похожа на модель разбиения тренировочного и тестового наборов.

Для этой цели было отобрано несколько предложений, содержащие кавычки, инициалы, римские цифры, английские слова и другие не тривиальные токены.

```
ls = u'Российская экологическая партия «Зелёные» — до февраля 2012 года, входила в состав партии СР.'
```

```
ls = 'Иоанн Павел II внёс некоторые изменения в правила проведения конклавов.'
```

```
ls = 'В 1977 году окончил исторический факультет Запорожского государственного педагогического института.'
```

```
ls = 'Куно I фон Ротт — пфальцграф Баварии, граф Фобурга, граф Нижнего Изара.'
```

```
ls = '— Гаев — «Вишнёвый сад», по пьесе А. П. Чехова, реж.'
```

```
ls = 'whites off earth now!! вики'
```

```
ls = 'смерть о'брайан, рональд кларк'
```

```
ls = 'барда (река) офис шерифа округа'
```

Первый же опробованный токенизатор spacy показал хорошие результаты и было принято решение остановиться на нем.

```
ldoc = ltokenize(ls)
```

```
барда NOUN nsubj
( PUNCT punct
река NOUN appos
) PUNCT punct
офис NOUN ROOT
шерифа NOUN nmod
округа NOUN nmod
```

```
ldoc = ltokenize(ls)
```

```
смерть NOUN ROOT
о'брайан PROPN nmod
, PUNCT punct
рональд PROPN conj
кларк PROPN flat:name
```

```
ldoc = ltokenize(ls)
```

```
whites X ROOT
off X flat:foreign
earth X flat:foreign
now X flat:foreign
! PUNCT punct
! PUNCT punct
вики NOUN ROOT
```

```
ldoc = ltokenize(ls)
```

```
— PUNCT punct
Гаев PROPN nsubj
— PUNCT punct
« PUNCT punct
Вишнёвый ADJ amod
сад NOUN ROOT
» PUNCT punct
, PUNCT punct
по ADP case
пьесе NOUN conj
А. PROPN nmod
П. PROPN flat:name
Чехова PROPN flat:name
, PUNCT punct
реж NOUN conj
. PUNCT punct
```

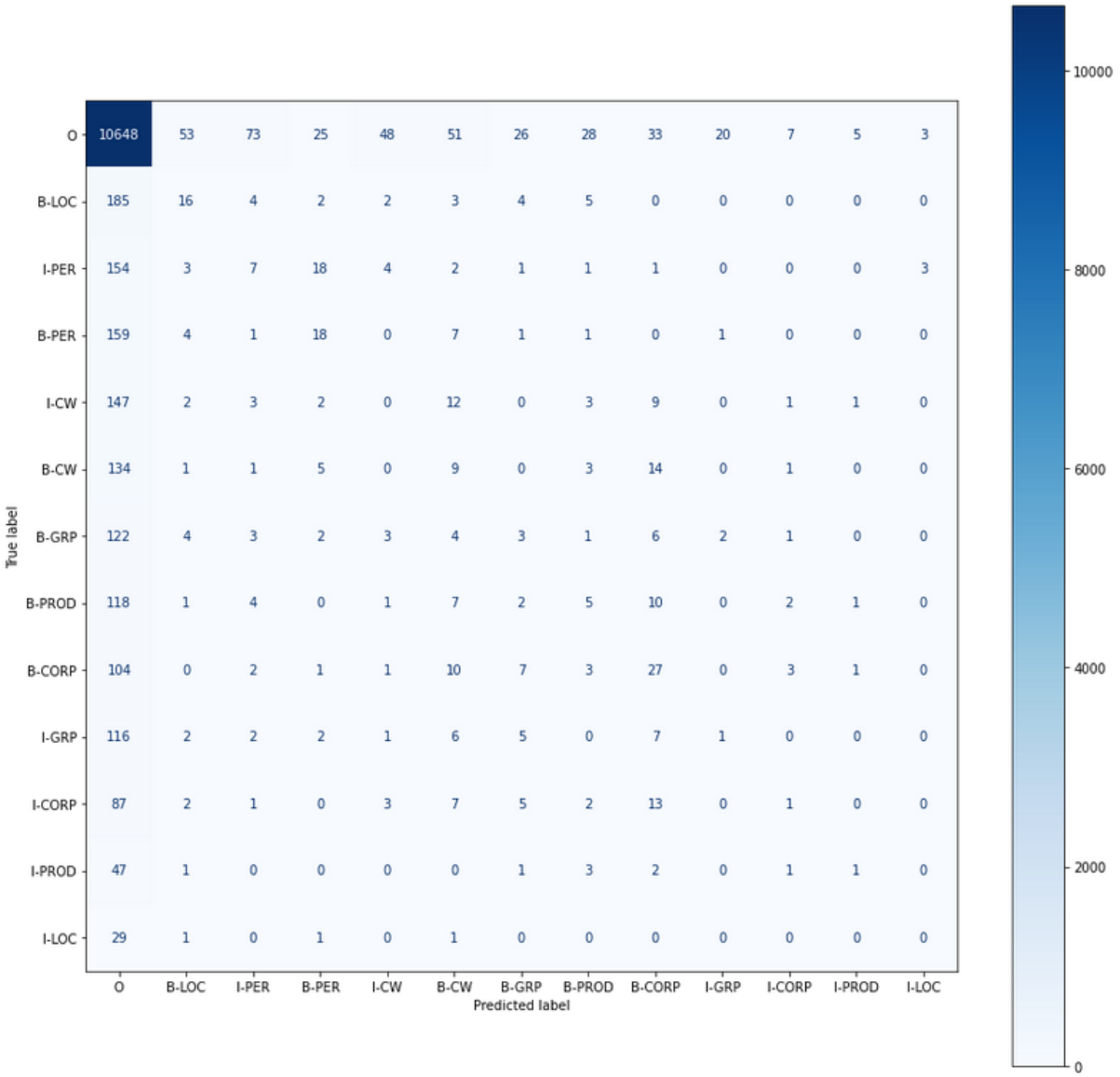
3. Формирование эмбеддингов и обучение базовой модели.

Несмотря на то что сущности часто бывают многословными, обычно задача NER сводится к задаче классификации на уровне токенов, т. е. каждый токен относится к одному из нескольких возможных классов. Есть несколько стандартных способов сделать это, но самый общий из них называется BIOES-схемой. Схема заключается в том,

чтобы к метке сущности (например, PER для персон или ORG для организаций) добавить некоторый префикс, который обозначает позицию токена в спане сущности. И тогда представляется возможным применение одной из базовых моделей классификации. Для классификации текстов можно применять разные алгоритмы. Машины опорных векторов (SVM) — это один из многих алгоритмов, которые мы можем выбирать при классификации текста. Данному алгоритму не нужно много тренировочных данных, чтобы начать давать точные результаты. При этом, SVM алгоритм требует больше вычислительных ресурсов, так-как он может достичь более точных результатов. Кроме этого на консультациях обсуждалось и было подтверждено при выполнении задания аттестации 1 модуля, что SVM очень часто на разных задачах показывает лучшие по точности результаты.

Для базовой модели было принято решение использовать классификатор SVM, т.к. он дает хорошие результаты при маленьком числе примеров и большом числе параметров.

Для формирования матрицы признаков были загружены предобученные модели с векторами из библиотеки spacy ru_core_news_sm. Проведено обучение, которое показало достаточно высокую точность Accuracy = 0,86. Однако при построении confusion matrix стало понятно, что подавляющее большинство корректных предсказаний приходится на не именованные сущности (Outside), а также на данный тип сущностей приходится большинство не корректных предсказаний именованных сущностей.



При этом из-за большой несбалансированности исходных данных получается высокая Accuracy. Это также видно и при формировании classification report.

	precision	recall	f1-score	support
O	0.90	0.93	0.91	11020
B-LOC	0.13	0.12	0.13	221
I-PER	0.05	0.05	0.05	194
B-PER	0.17	0.15	0.16	192
I-CW	0.00	0.00	0.00	180
B-CW	0.07	0.10	0.08	168
B-GRP	0.02	0.01	0.02	151
B-PROD	0.06	0.05	0.06	151
B-CORP	0.20	0.18	0.19	159
I-GRP	0.02	0.01	0.01	142
I-CORP	0.08	0.02	0.03	121
I-PROD	0.04	0.02	0.02	56
I-LOC	0.00	0.00	0.00	32
accuracy			0.81	12787
macro avg	0.13	0.13	0.13	12787
weighted avg	0.78	0.81	0.80	12787

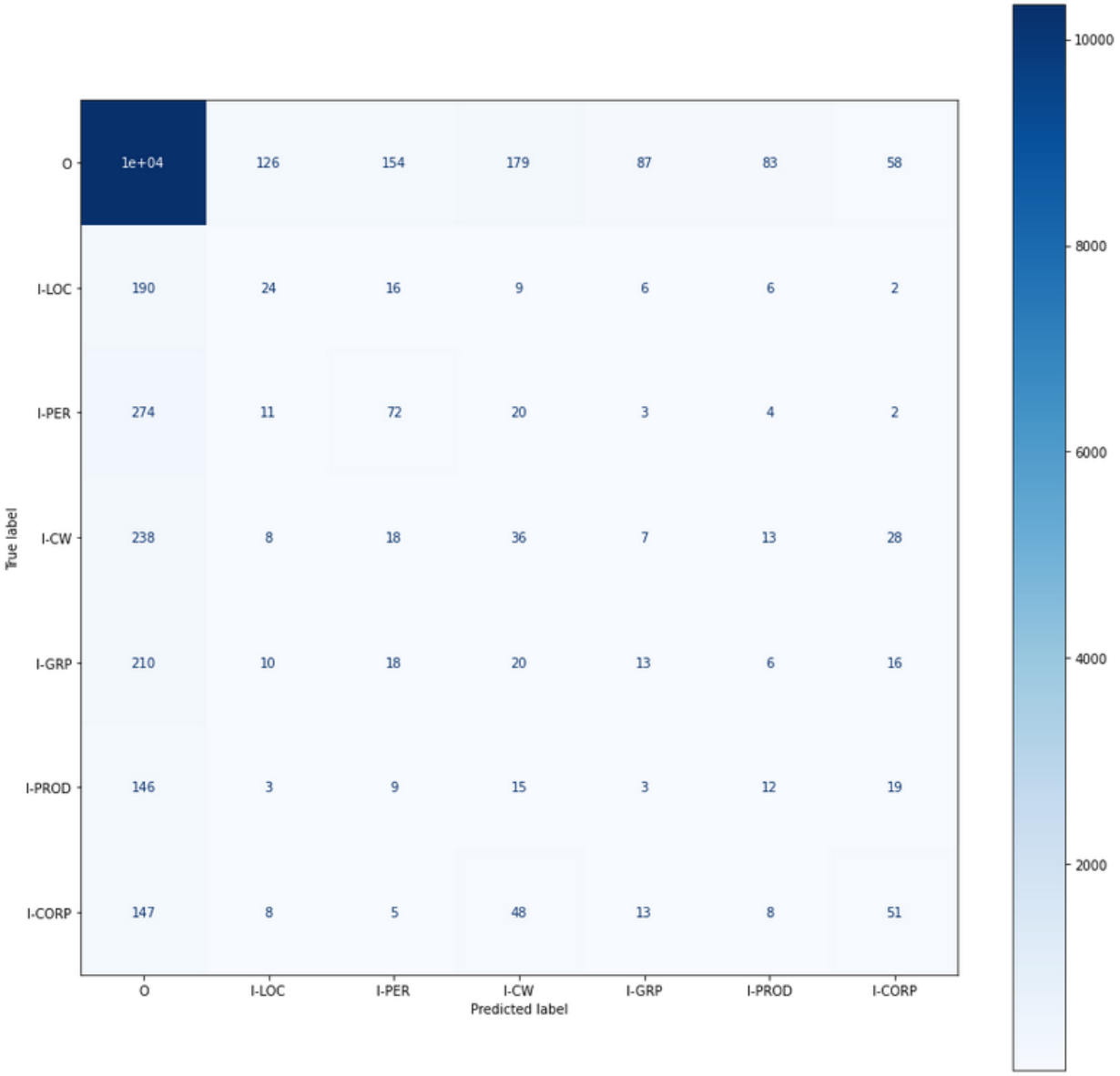
Для улучшения модели были сделаны следующие доработки:

- была убрана схема BIO из изначальных меток;

- реализована функция учета контекста (2 соседних слов), методом конкатенации их векторов;

- добавлены one-hot вектора с частями речи (POS tags);

- включена балансировка весов модели class_weight="balanced".



	precision	recall	f1-score	support
O	0.90	0.94	0.92	11020
I-LOC	0.13	0.09	0.11	253
I-PER	0.25	0.19	0.21	386
I-CW	0.11	0.10	0.11	348
I-GRP	0.10	0.04	0.06	293
I-PROD	0.09	0.06	0.07	207
I-CORP	0.29	0.18	0.22	280
accuracy			0.82	12787
macro avg	0.27	0.23	0.24	12787
weighted avg	0.79	0.82	0.81	12787

Все вышеуказанные действия хотя и позволили несколько улучшить качество классификации NER, однако кардинальных улучшений не дали. В связи с этим было принято решение оставить данную модель как базовую и перейти к реализации с помощью нейронных сетей.

4. **Обучение seq2seq модели.**

Сегодня одним из самых качественных инструментов для решения задачи NER является LSTM – сети с долгой краткосрочной памятью. Это модификация рекуррентных нейронных сетей (RNN), особенность которых в способности обучаться долговременным зависимостям. Эта особенность позволяет им анализировать текст, опираясь не только на конкретное слово, которое сеть «видит» в данный момент, но и на взаимосвязь этого слова с «увиденными» ранее в рамках того же текста. Таким образом, модель учится понимать контекст. Это обеспечивает высокую точность работы модели.

Двунаправленные LSTM являются расширением традиционных LSTM, которые могут улучшить производительность модели в задачах классификации последовательностей. В задачах, где доступны все временные шаги входной последовательности, двунаправленные LSTM обучают два вместо одного LSTM во входной последовательности. Первый на входной последовательности как есть, а второй на обратной копии входной последовательности. Это может обеспечить дополнительный контекст для сети и привести к более быстрому и даже более полному изучению проблемы. Такая обработка последовательности в двух направлениях особенно эффективна в области NLP потому что учитывается контекст всего высказывания, а не

только тот, что сеть увидела с начала предложения, как в случае с однонаправленными LSTM.

Перед применением BiLSTM необходимо подготовить данные. Для этого создаем словари из всех уникальных токенов и меток с присвоением каждому его порядкового номера. Затем каждому токену и каждой метке в исходном наборе присваиваем индекс из словаря.

	words	markers	sentense_num	word_idx	marker_idx
0	российская	B-GRP	1	9681	4
1	экологическая	I-GRP	1	20293	4
2	партия	I-GRP	1	15291	4
3	«зелёные»	I-GRP	1	38551	4
4	—	O	1	43181	6

Затем группируем все токены по предложениям и получаем последовательности из токенов и меток.

	sentense_num	words	markers	word_idx	marker_idx
0	1	[российская, экологическая, партия, «зелёные»,...	[B-GRP, I-GRP, I-GRP, I-GRP, O, O, O, O, O, O,...	[9681, 20293, 15291, 38551, 43181, 1596, 30737...	[4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6]
1	2	[также, посещал, два, семинара, бартольда, кёй...	[O, O, O, O, B-PER, I-PER, O, O, O, B-PER, O, ...	[24008, 47589, 42010, 40044, 31326, 34745, 239...	[6, 6, 6, 6, 1, 1, 6, 6, 6, 1, 6, 1, 6]
2	3	[в, 1999, —, 2006, играла, за, национальную, с...	[O, O, O, O, O, O, B-GRP, I-GRP, I-GRP, O]	[26415, 3416, 43181, 37240, 37335, 27712, 2989...	[6, 6, 6, 6, 6, 6, 4, 4, 4, 6]
3	4	[«, джульетта, », —, кинофильм, 2016, года, ис...	[O, B-CW, O, O, O, O, O, O, O, O, O, O, O]	[49441, 13541, 3797, 43181, 7570, 32408, 34363...	[6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6]
4	5	[мякоть, спелого, плода, съедобна, в, свежем, ...	[O, O, O, O, O, O, O, O, O, O, O, O, O, B-P...	[43136, 34860, 13361, 5001, 26415, 29718, 1827...	[6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 6]

Поскольку для работы нейронной сети все последовательности должны быть одинаковой длины, то вычисляем размер самого длинного предложения, а все остальные доводим служебным токеном <PAD> до нужной длины.

Для тестового набора данных все слова, отсутствующие в словаре заменяем служебным токеном <UNKNOWN>. Предложения больше максимальной длины обрезаем. Метки преобразуем в one-hot формат.

Формируем архитектуру сети. Добавляем Embedding Layer, который преобразовывает положительные целые числа (индексы) в плотные (dense) векторы фиксированного размера, например [[4], [20]] -> [[0.25, 0.1], [0.6, -0.2]]. Второй слой – непосредственно BiLSTM. И последний слой - TimeDistributed layer. Оптимизатор – Adam и функция потерь для мультиклассовой классификации - categorical_crossentropy.

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_4 (Embedding)     (None, 59, 300)           15770400

 bidirectional_2 (Bidirectio  (None, 59, 600)          1442400
 nal)

 time_distributed_2 (TimeDis  (None, 59, 9)            5409
 tributed)

=================================================================
Total params: 17,218,209
Trainable params: 17,218,209
Non-trainable params: 0
```

Тренируем модель на 6 эпохах и видим, что она практически обучается уже на первой эпохе и далее validation accuracy не растет. При этом classification report и confusion matrix показывают картину очень схожую с той, что мы наблюдали при использовании SVM модели. При этом введенная новая метрика – процент правильно предсказанных полных предложений показала по 1 предложению на тестовой и валидационной выборках.

Для борьбы с переобучением были изменены параметры модели dropout=0.9, output_dim = 300.

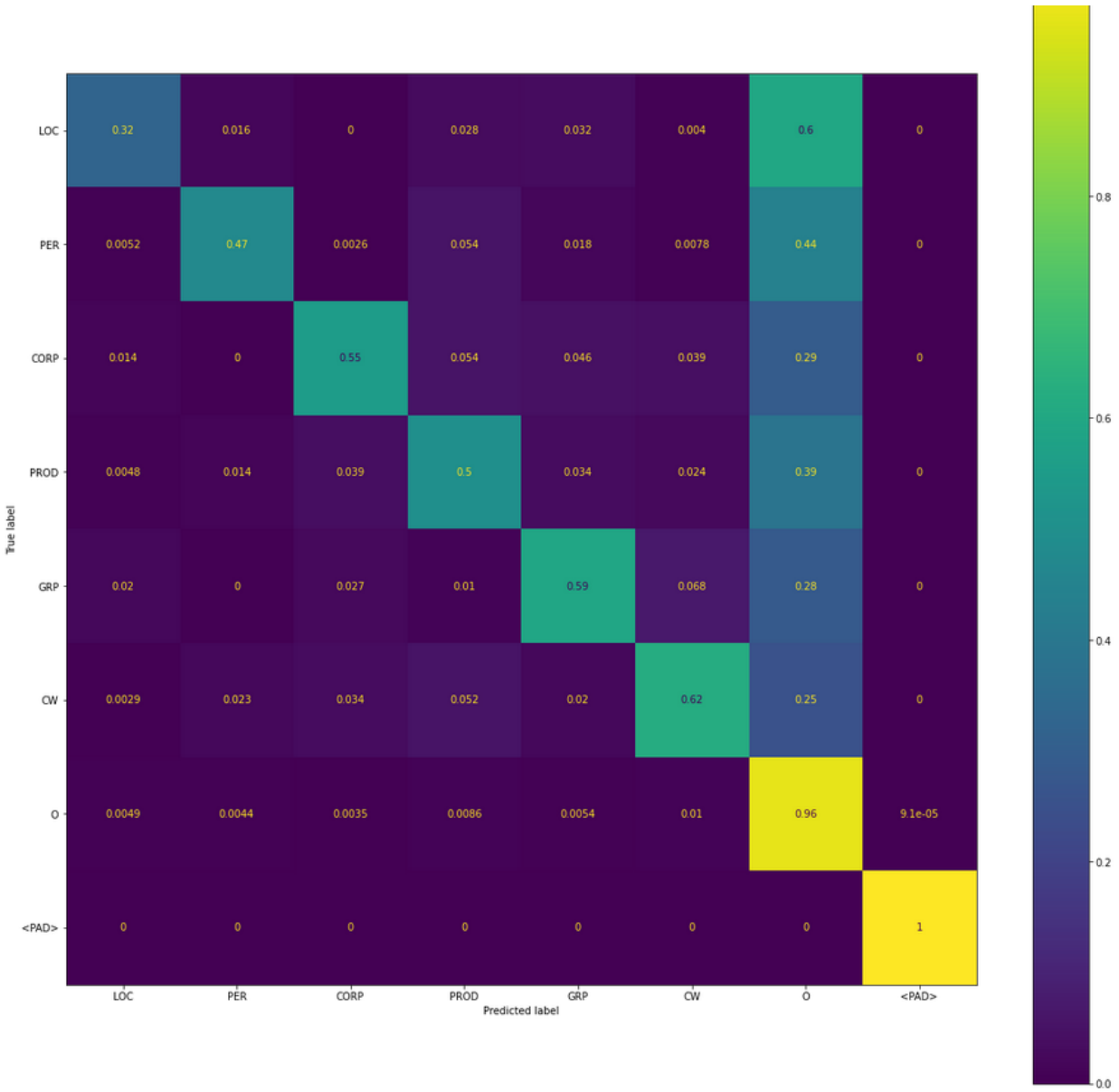
В стандартной нейронной сети производная, полученная каждым параметром, сообщает ему, как он должен измениться, чтобы, учитывая деятельность остальных блоков, минимизировать функцию конечных потерь. Поэтому блоки могут меняться, исправляя при этом ошибки других блоков. Это может привести к чрезмерной совместной адаптации (co-adaptation), что, в свою очередь, приводит к переобучению, поскольку эти совместные адаптации невозможно обобщить на данные, не участвовавшие в обучении. Dropout предотвращает совместную адаптацию для каждого скрытого блока, делая присутствие других скрытых блоков ненадежным. Поэтому скрытый блок не может полагаться на другие блоки в исправлении собственных ошибок.

Результат получился уже лучше. Модель стала полностью правильно классифицировать 144 предложения из 800 - 18%

	precision	recall	f1-score	support
I-CW	0.40	0.45	0.42	180
B-LOC	0.52	0.29	0.37	221
I-PER	0.55	0.38	0.45	194
I-CORP	0.58	0.21	0.30	121
_ O	0.00	0.00	0.00	0
I-PROD	0.31	0.27	0.29	56
O	0.92	0.98	0.95	11020
B-PER	0.47	0.32	0.38	192
I-GRP	0.72	0.34	0.46	142
B-PROD	0.43	0.37	0.40	151
B-CORP	0.67	0.19	0.29	159
B-CW	0.39	0.36	0.38	168
B-GRP	0.71	0.24	0.36	151
I-LOC	0.14	0.03	0.05	32
<UNKNOWN>	0.00	0.00	0.00	0
<PAD>	1.00	1.00	1.00	34413
micro avg	0.97	0.97	0.97	47200
macro avg	0.49	0.34	0.38	47200
weighted avg	0.96	0.97	0.96	47200

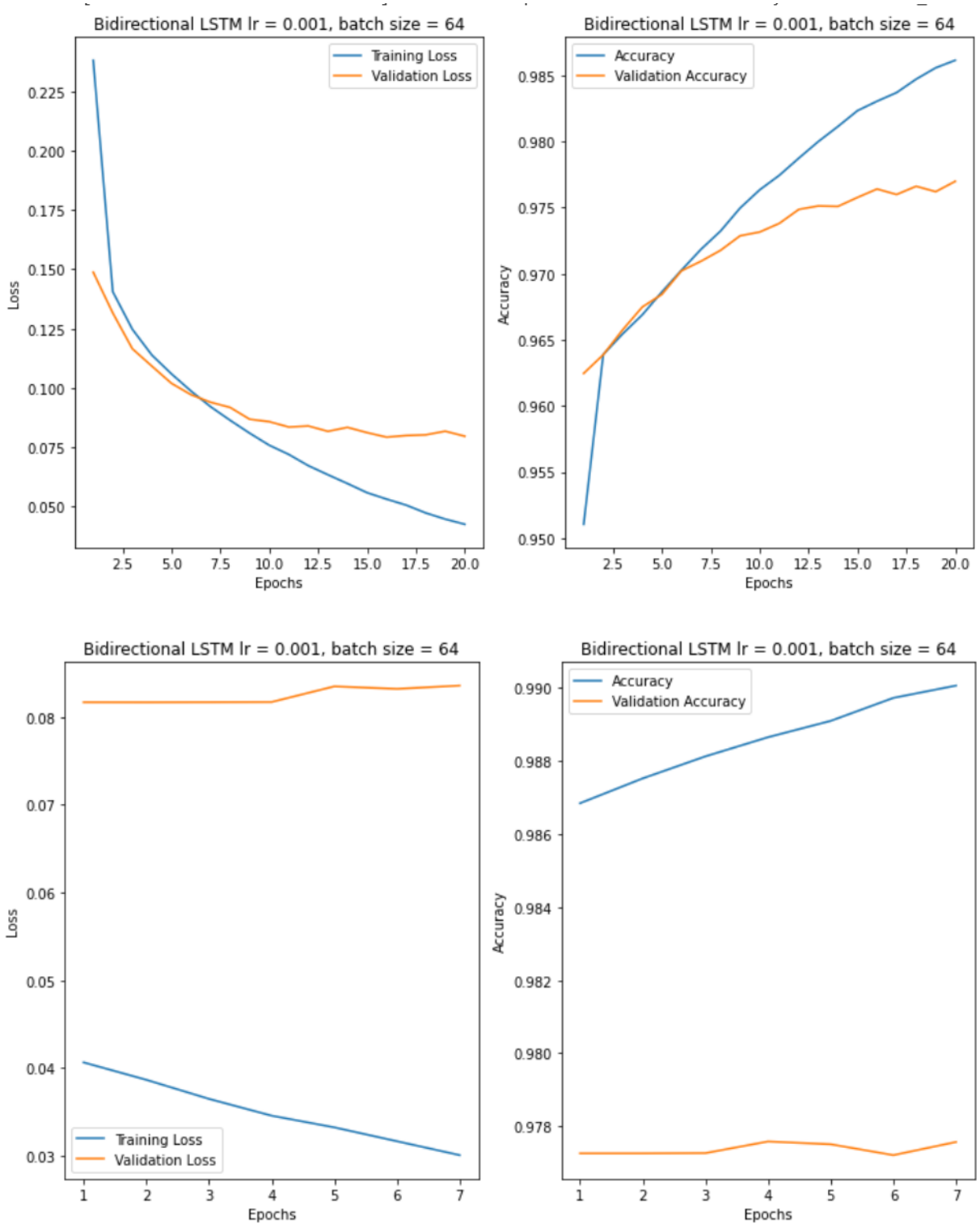
Для еще большего улучшения модели были предприняты следующие шаги:
- была убрана схема BIO из изначальных меток;
- были добавлены предобученные эмбединги из библиотеки spacy, но уже расширенные ru_core_news_lg с размерностью 300.

Для этого сформирована embedding matrix на основе всего входного словаря и передана в параметр weights первого слоя. Модель стала работать медленнее, но результат стал намного лучше. На 20 эпохах обучения модель достигла следующих результатов.



	precision	recall	f1-score	support
LOC	0.55	0.32	0.41	253
PER	0.74	0.47	0.58	386
CORP	0.70	0.55	0.62	280
PROD	0.39	0.50	0.44	207
GRP	0.63	0.59	0.61	293
CW	0.59	0.62	0.60	348
O	0.94	0.96	0.95	11020
<PAD>	1.00	1.00	1.00	34413
accuracy			0.97	47200
macro avg	0.69	0.63	0.65	47200
weighted avg	0.97	0.97	0.97	47200

Модель стала безошибочно распознавать 242 предложения из 800 - 30.25%. При этом исходя из кривой обучения, модель приблизилась к своему максимуму на 20 эпохах. И дальнейшее увеличения числа эпох не дало ощутимого прироста точности.



Дальнейшее увеличение точности модели возможно с увеличением количества исходных данных.

5. Создание ВЕБ-приложения

После разработки и тестирования, обученная модель, а также необходимые словари были сохранены. С помощью библиотеки Plotly Dash был разработан ВЕБ-интерфейс, который позволяет вводить любой текст и на выходе получать набор меток для него.