

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ПОСТАНОВКА ЗАДАЧИ	4
1.1 NER	4
1.2 Предмет исследования	4
1.3 Актуальность работы	5
2 МЕТОДЫ РЕШЕНИЯ	6
2.1 LSTM-CRF	6
2.2 CRF + AutoEncoder	8
2.3 ELMo	9
2.4 BERT	9
2.5 CNN Large + fine-tune	11
3 ПРОДЕЛАННАЯ РАБОТА	12
3.1 Реализация существующих подходов	13
3.2 Исследование	15
3.3 Результаты	16
3.4 Выводы	18
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	20

ВВЕДЕНИЕ

Автоматическая обработка естественного языка – популярная область исследований специалистов по машинному обучению. С развитием технологий появляется все больше задач, успешное решение которых требует сложных алгоритмов и больших затрат ресурсов.

Несмотря на то что задача распознавания именованных сущностей была сформулирована в 90-х годах 20 века, до сих пор нельзя утверждать, что она является полностью решенной: нет алгоритма, который решал бы задачу лучше, чем ассессор. В настоящее время задача может рассматриваться как отдельная, а может и как часть большой системы. Таким образом, она является популярной (каждый год появляются новые, более успешные алгоритмы решения) и актуальной.

В данной работе изучаются различные методы решения задачи распознавания именованных сущностей. В частности, рассмотрены основные методы, придуманные в последние 4 года. Были реализованы подходы, описанные в статьях [5] и [9]. Над архитектурой, предложенной в последней статье, было проведено исследование. Была выдвинута, а впоследствии опровергнута, гипотеза о возможном упрощении данной архитектуры.

1 ПОСТАНОВКА ЗАДАЧИ

1.1 NER

Распознавание именованных сущностей является важной подзадачей при извлечении информации из неструктурированных данных. В задаче, представленной на конференции CoNLL 2003, необходимо отнести каждое слово в предложении к одному из четырех классов: PER (человек), LOC (локация), ORG (организация), MISC (другое). Используется IOB-нотация: для сущностей, состоящих более чем из одного слова, необходимо определить начальное (B-) и все внутренние (I-) слова. Если слово не относится ни к одному из четырех классов, то оно помечается отдельно (O).

Стоит отметить, что помимо CoNLL 2003 есть и другие соревнования и датасеты, но в данный момент большинство авторов статей сравнивают свои модели с результатами, показанными на этой задаче. Отметим также, что помимо IOB существуют и другие нотации. В частности, в этой работе используется IOBES-нотация (согласно [7], [2] эта схема позволяет получить более высокие результаты), а для подсчета качества распознавания она переводится в IOB-схему.

1.2 Предмет исследования

В [9] был предложен эффективный способ использования дополнительных признаков для решения задачи распознавания именованных сущностей.

В данной работе проверяется следующая гипотеза: помимо вычисления символьных эмбедингов, сверточная нейронная сеть, используемая в архитектуре финальной модели, сможет выучить и дополнительные признаки, т.е. архитектуру можно упростить без потери качества.

1.3 Актуальность работы

Распознавание именованных сущностей может рассматриваться как отдельная задача (например, для поиска и защиты персональных данных в Интернете) или как часть большой системы (например, как составная часть голосового помощника). В обоих случаях будет полезно минимизировать ресурсы, необходимые для построения и поддержания качественного решения.

2 МЕТОДЫ РЕШЕНИЯ

С 2016 года качество решения задачи распознавания именованных сущностей значительно увеличилось.

Таблица 2.1 – Качество решения задачи NER

Архитектура	Год	F1-score
CNN Large + fine-tune	2019	93.5
BERT	2018	92.8
ELMo	2018	92.22
CRF + AutoEncoder	2018	91.87
LSTM-CRF	2016	91.21

В данном разделе будут описаны самые важные подходы к решению рассматриваемой задачи.

2.1 LSTM-CRF

Стандартным нейросетевым подходом к решению задачи распознавания именованных сущностей можно считать следующую архитектуру: конкатенация символьных и словных эмбеддингов подается в двунаправленный LSTM; непосредственная классификация производится с помощью CRF.

В [4] и [5] были предложены два различных подхода к получению символьных эмбеддингов (такие подходы встречались и раньше, но именно эти работы считаются самыми успешными). Основная идея – использование механизма, который может работать с данными произвольной длины. Для этого в [4] используется рекуррентная нейронная сеть, а в [5] – сверточная.

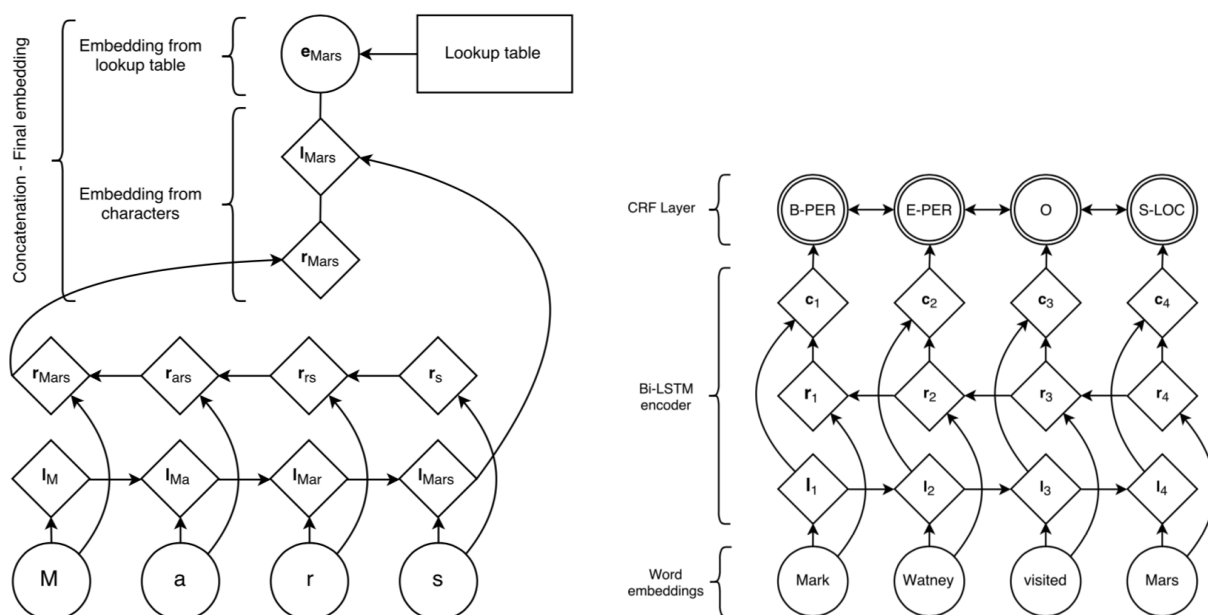


Рис. 2.1 – Символьные эмбединги и архитектура, предложенные в [4]

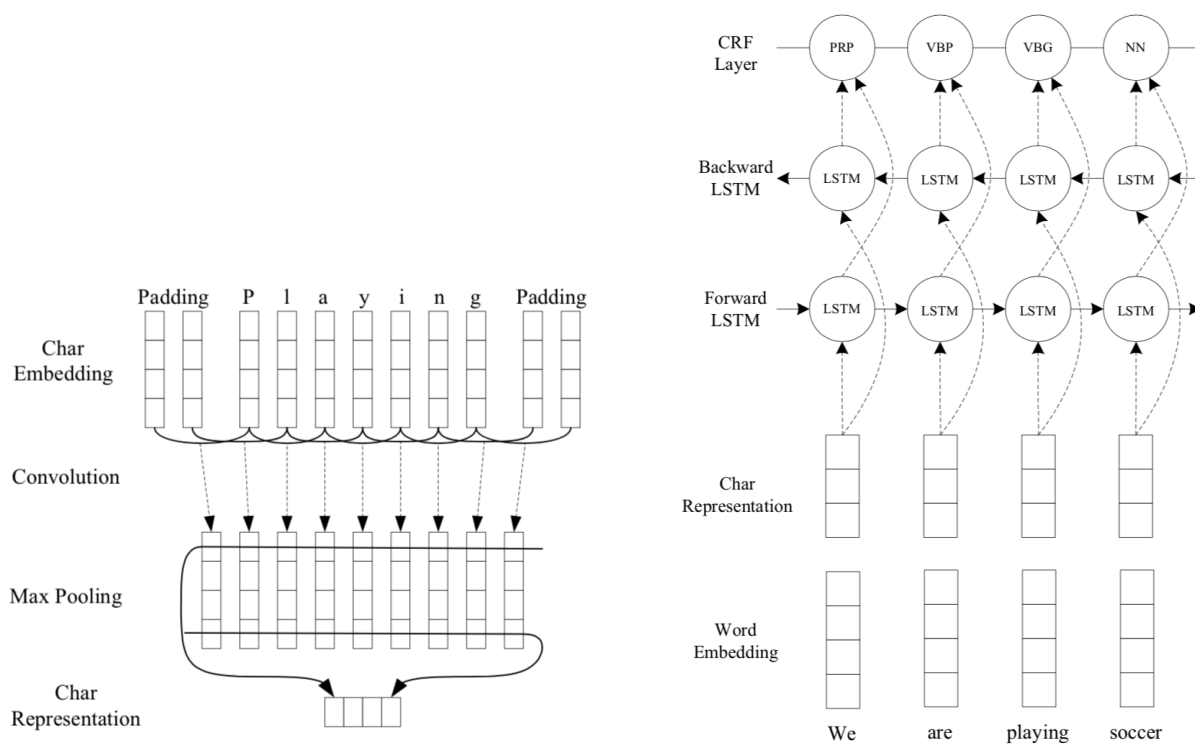


Рис. 2.2 – Символьные эмбединги и архитектура, предложенные в [5]

2.2 CRF + AutoEncoder

Простым способом улучшить качество модели является использование дополнительных признаков: POS-тэгов, данных из географических справочников, шаблонов капитализации. В [9] помимо основной задачи предлагается решать также задачу восстановления этих дополнительных признаков. Для этого вводится автоэнкодер и его функция потерь.

$$\mathcal{L}_{AE}^t = \sum_{i=0}^T XEntropy(f_i^t, \hat{f}_i^t)$$

$$\mathcal{L} = \mathcal{L}_{CRF} + \sum_t \lambda_t \mathcal{L}_{AE}^t$$

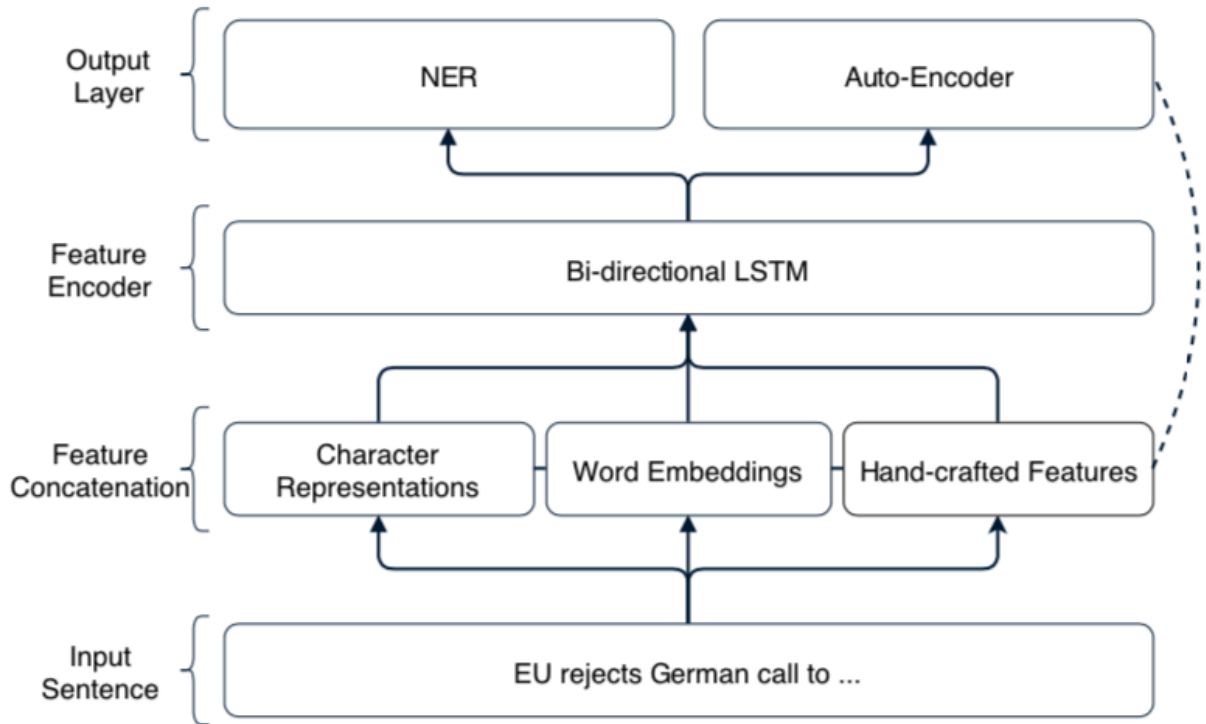


Рис. 2.3 – Архитектура, предложенная в [9]

2.3 ELMo

В статье [6] описывается способ получения эмбеддингов, которые будут учитывать не только синтаксические и семантические признаки, а также контекст рассматриваемого слова. Для этого используется многослойный двунаправленный LSTM, взвешенная сумма состояний которого и определяет вектор токена. Эмбеддинги, полученные таким способом, позволили улучшить качество решения сразу нескольких традиционных задач в области обработки естественного языка.

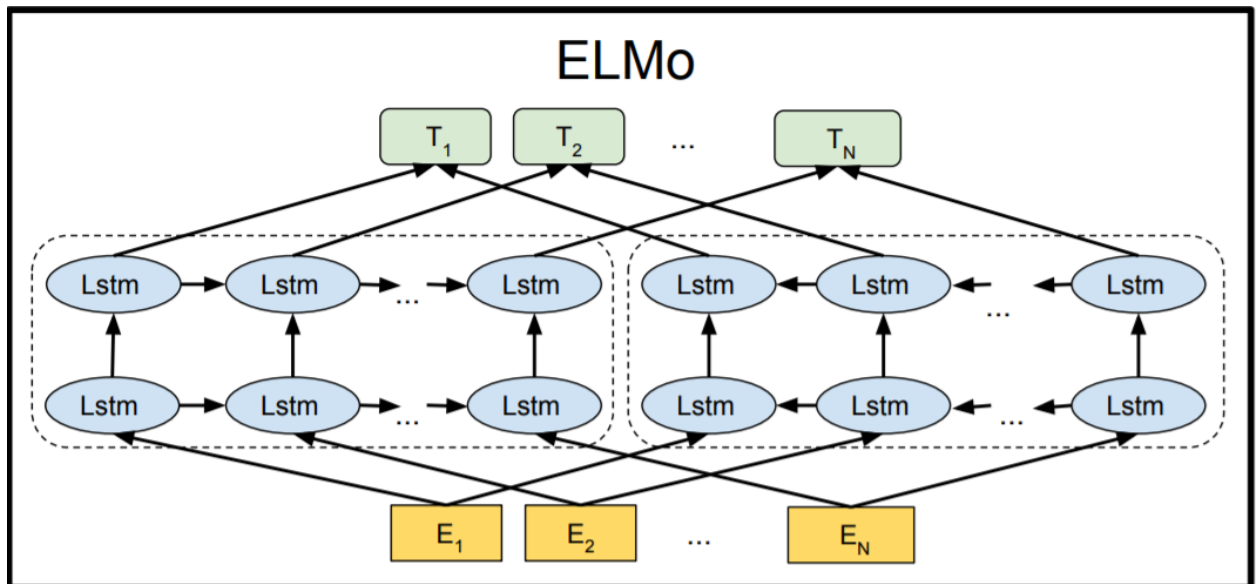


Рис. 2.4 – Архитектура ELMo

2.4 BERT

В 2018 году группа ученых из Google AI Language предложила универсальную архитектуру BERT (см. [3]), которая позволила получить State-of-the-art результаты на большом числе NLP задач.

В то время как большинство языковых моделей работают только с одним из контекстов (левым или правым), в BERT применяется идея Masked

LM – часть слов предложения маскируется специальным символом, а затем эти слова предсказываются (получается, что учитываются все токены).

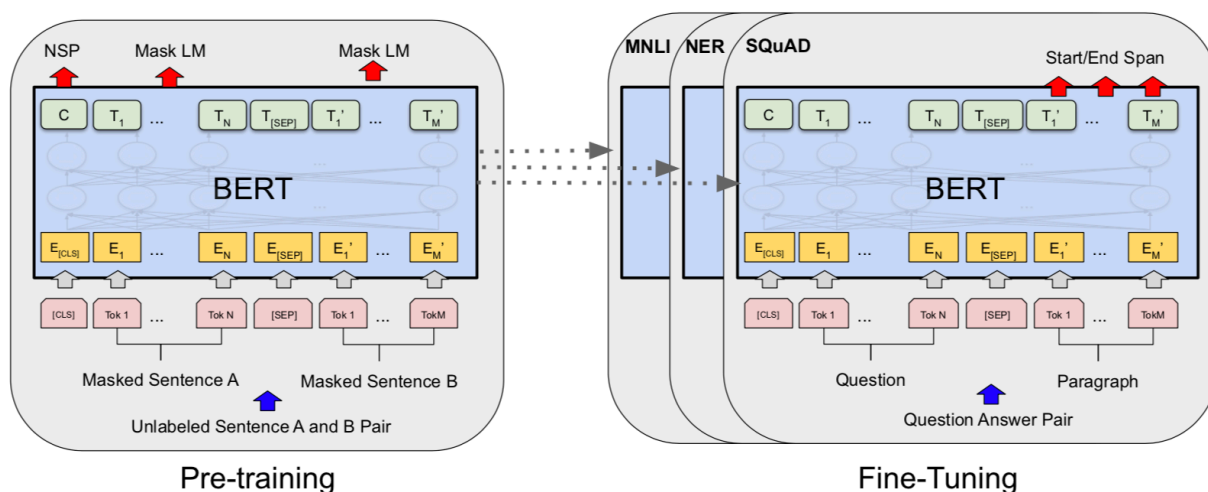


Рис. 2.5 – Pre-training и Fine-Tuning – два процесса, составляющих алгоритм

Универсальность модели также обеспечивается следующими двумя факторами: на вход подается не одно, а два предложения; дополнительно решается задача NSP – определяется, является ли второе предложение продолжением первого.

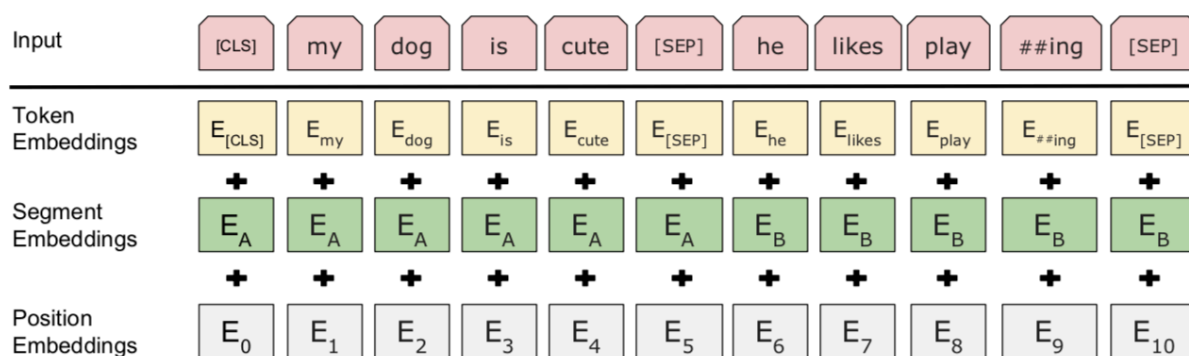


Рис. 2.6 – На вход BERT подаются два предложения, разделенные специальным символом

2.5 CNN Large + fine-tune

Еще более высокие результаты были опубликованы в [1]. Как и в BERT, сначала происходит Pre-training, а затем – Fine-tuning. Основными отличиями являются:

- а) Маскинг каждого токена, а не случайного набора
- б) Отсутствие необходимости решать задачу NSP
- в) Измененные блоки трансформера (см. [8])
- г) Процесс обучения похож на процесс обучения классической языковой модели

3 ПРОДЕЛАННАЯ РАБОТА

Для того чтобы лучше разобраться в исследуемой задаче, сначала были реализованы существующие подходы, а уже после этого проводились эксперименты.

Работа велась с данными, представленными на конференции CoNLL 2003. Основной метрикой качества является F1-score. Для дополнительных задач так же измерялась точность предсказаний.

Таблица 3.2 – Датасет CoNLL 2003

Данные	Количество предложений
Обучающая выборка	14041
Валидационная выборка	3250
Тестовая выборка	3453

```
[15] train_sentences[141].  
↳ [['Both', 'O'],  
    ['Iran', 'I-LOC'],  
    ['and', 'O'],  
    ['Turkey', 'I-LOC'],  
    ['mount', 'O'],  
    ['air', 'O'],  
    ['and', 'O'],  
    ['land', 'O'],  
    ['strikes', 'O'],  
    ['at', 'O'],  
    ['targets', 'O'],  
    ['in', 'O'],  
    ['northern', 'O'],  
    ['Iraq', 'I-LOC'],  
    ['in', 'O'],  
    ['pursuit', 'O'],  
    ['of', 'O'],  
    ['their', 'O'],  
    ['own', 'O'],  
    ['Kurdish', 'I-MISC'],  
    ['rebels', 'O'],  
    ['.', 'O']]
```

Рис. 3.1 – Пример обработанного предложения

3.1 Реализация существующих подходов

Сначала была реализована архитектура, предложенная в статье [5]. В отличие от оригинальной статьи была реализована возможность обучения с батчем произвольной длины. Это позволило ускорить процесс обучения, а также сделать его более стабильным.

Таблица 3.3 – Параметры обучения

Размер батча	128
Размерность LSTM	300
Размерность CNN	30
Размер ядра CNN	4
Размер фильтра CNN	30
Dropout	0.5
Recurrent dropout	0.3
Предобученные эмбединги	GloVe.6B.100d
Оптимизатор	Nadam
Ограничение нормы градиента	5

Таблица 3.4 – Сравнение качества реализации модели, представленной в [5]

Модель	F1-score
Ma & Hovy	91.21
Полученная реализация	90.46

Обучение производилось с использованием GPU NVIDIA Tesla K80. Процесс занял около 8 часов на Google Colaboratory. В результате получилась

модель, уступающая оригинальной в качестве. Это может быть связано с тем, что не проводилась явная инициализация весов нейросети. Также возможно влияние используемого фреймворка и рабочей машины.

Следующий этап – реализация архитектуры из статьи [9]. Как и в предыдущем случае, было реализовано обучение с батчем произвольной длины.

Таблица 3.5 – Параметры обучения лучшей модели

Размер батча	32
Размерность LSTM	300
Размерность CNN	25
Размер ядра CNN	3
Размер фильтра CNN	25
Dropout	0.5
Recurrent dropout	0.5
Предобученные эмбединги	GloVe.6B.300d
Оптимизатор	SGD
Learning rate	0.015
Убывание весов	1e-6
Момент	0.9
Ограничение нормы градиента	5

Таблица 3.6 – Сравнение качества реализации модели, представленной в [9]

Модель	F1-score
Wu et al.	91.89 \pm 0.23
Полученная реализация	91.64

Обучение производилось с использованием GPU NVIDIA Tesla K80. Процесс занял около 8 часов на Google Colaboratory. В результате получилась модель, попадающая в заявленный в статье интервал.

3.2 Исследование

Основной целью работы являлась проверка следующей гипотезы: на вход автоэнкодеру (см. [9]) можно не подавать сгенерированные вручную признаки. Остановимся на них подробнее.

Во-первых, это POS-тэги, которые указывают, какой частью речи является данное слово. Используется самая распространенная классификация – каждое слово относится к одному из 45 классов. Так как обучение происходит с батчами, предложения в которых могут иметь разную длину, был добавлен еще один класс, обозначающий паддинг.

Во-вторых, используются шаблоны капитализации. Они показывают, есть ли в слове заглавные буквы, цифры и др. Всего авторам [9] удалось выделить 151 класс. Здесь также был добавлен дополнительный класс для паддингов.

В-третьих, используются ”газитиры” – сведения из картографических справочников. Под них выделено 4 класса.

В [9] можно подробно прочитать, как производился сбор и разметка этих дополнительных данных.

Согласно архитектуре, на вход автоэнкодер получает конкатенацию символьных и словных эмбеддингов, а также конкатенацию ”газитиров”, POS-тегов и шаблонов капитализации, причем последние два подаются в склеенном состоянии.

Помимо основной задачи NER, решается задача восстановления полученных на вход ”газитиров”, POS-тегов и шаблонов капитализации. Таким образом, у модели 4 выхода.

В данной работе проверяется, можно ли оставив выход нейросети прежним, отказаться от подачи на вход либо всех дополнительных признаков, либо только от POS-тегов и шаблонов капитализации.

3.3 Результаты

Для проверки гипотезы были обучены еще две нейросети. Одна получала на вход дополнительно "газители", другая не получала никаких дополнительных признаков.

Таблица 3.7 – Параметры обучения моделей

Параметры	С "газителями"	Без дополнительных признаков
Размер батча	64	48
Размерность LSTM	400	200
Размерность CNN	25	27
Размер ядра CNN	4	3
Размер фильтра CNN	25	27
Dropout	0.4	0.47
Recurrent dropout	0.5	0.47
Предобученные эмбединги	GloVe.6B.300d	GloVe.6B.300d
Оптимизатор	SGD	SGD
Learning rate	0.03	0.04
Убывание весов	1e-5	5e-5
Момент	0.9	0.9
Ограничение нормы градиента	5	10
Метод Нестерова	Да	Да

Обучение производилось с использованием GPU NVIDIA Tesla K80. Во всех случаях процесс занял около 8 часов на Google Colaboratory. Получились следующие результаты.

Таблица 3.8 – Качество реализации оригинальной модели

	F1-score	Accuracy
NER	91.5 ± 0.1	
POS-теги	98.90	98.90
Шаблоны капитализации	99.02	99.02

Таблица 3.9 – Качество модели, принимающей на вход ”газитиры”

	F1-score	Accuracy
NER	91.05	
POS-теги	88.83	94.98
Шаблоны капитализации	65.09	98.12

Таблица 3.10 – Качество модели, не принимающей на вход дополнительных признаков

	F1-score	Accuracy
NER	90.78	
POS-теги	94.31	99.09
Шаблоны капитализации	72.77	99.14

В результате, достичь того же качества, что и в оригинальной статье, не удалось.

3.4 Выводы

Как видно из результатов, отказаться от подачи дополнительных признаков на вход автоэнкодеру без потери качества нельзя. Объяснить это можно следующим образом: когда на вход подаются дополнительные признаки, нейросеть выучивает необходимое представление, о чем говорит высокий F1-score и точность. Если же признаки не подавать на вход, то нейросеть учится не так успешно: точность восстановления высокая, но необходимые для правильной классификации признаки не обучаются (об этом говорит невысокий F1-score).

ЗАКЛЮЧЕНИЕ

В данной работе были подробно изучены способы решения задачи распознавания именованных сущностей. Алгоритмы, предложенные в статьях [5] и [9] были успешно реализованы. Также проведено исследование архитектуры, рассматриваемой в последней статье. По результатам этого исследования была отвергнута гипотеза о возможном упрощении модели.

Исходный код моделей, их обучение и исследование, а также все необходимые данные и скрипты выложены в репозитории ^[*] в открытом доступе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Alexei Baevski и др. *Cloze-driven Pretraining of Self-attention Networks*. 2019. arXiv: 1903.07785.
- [2] Jason P. C. Chiu и Eric Nichols. *Named Entity Recognition with Bidirectional LSTM-CNNs*. 2015. arXiv: 1511.08308.
- [3] Jacob Devlin и др. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. arXiv: 1810.04805.
- [4] Guillaume Lample и др. *Neural Architectures for Named Entity Recognition*. 2016. arXiv: 1603.01360.
- [5] Xuezhe Ma и Eduard Hovy. *End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF*. 2016. arXiv: 1603.01354.
- [6] Matthew E. Peters и др. *Deep contextualized word representations*. 2018. arXiv: 1802.05365.
- [7] Lev Ratinov и Dan Roth. “Design Challenges and Misconceptions in Named Entity Recognition”. В: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. Boulder, Colorado: Association for Computational Linguistics, июнь 2009, с. 147—155. URL: <https://www.aclweb.org/anthology/W09-1119>.
- [8] Ashish Vaswani и др. *Attention Is All You Need*. 2017. arXiv: 1706.03762.
- [9] Minghao Wu, Fei Liu и Trevor Cohn. *Evaluating the Utility of Hand-crafted Features in Sequence Labelling*. 2018. arXiv: 1808.09075.