

## Аннотация

В данной работе исследована возможность использования семантико-синтаксического анализатора Comreno в качестве источника высокоуровневых признаков для решения задачи распознавания именованных сущностей в рамках нейросетевого подхода. Исследование проводилось на англоязычном корпусе CoNLL 2003. Полученные результаты показывают, что высокоуровневые признаки дают ощутимый прирост оценки качества, без какой-либо инженерии над ними.

**Ключевые слова:** нейронные сети, распознавание именованных сущностей.

# Содержание

Введение . . . . .	3
1 Теоретико-аналитическая часть . . . . .	5
1.1 Постановка задачи . . . . .	5
1.2 Обзор литературы . . . . .	5
1.3 Обзор корпусов для задачи NER . . . . .	6
1.3.1 Схемы разметки корпуса . . . . .	6
1.4 Обзор нейросетевого подхода к решению задачи NER . .	6
1.4.1 Векторные представления слов . . . . .	7
1.4.2 Описание нейросетевых моделей . . . . .	8
1.4.2.1 Оконная модель . . . . .	8
1.4.2.2 Сверточная модель . . . . .	11
1.4.3 Выбор метода решения задачи . . . . .	14
1.5 Синтактико-семантические признаки . . . . .	14
2 Проектная часть . . . . .	16
2.1 Алгоритмизация выбранного метода решения задачи . .	16
2.2 Программная часть . . . . .	17
2.2.1 Выбор программного фреймворка . . . . .	17
2.2.2 Реализованная модель . . . . .	17
3 Экспериментальная часть . . . . .	18
3.1 Корпус CoNLL 2003 . . . . .	18
3.2 Синтактико-семантические признаки Compreno . . . . .	18
3.3 Эксперименты без синтактико-семантических признаков	18
3.4 Эксперименты с синтактико-семантическими признака-	
ми сжатыми с использованием SVD . . . . .	21
3.5 Эксперименты с синтактико-семантическими признака-	
ми для совместно-оптимизированной нейросети . . . . .	22
Заключение . . . . .	25
Список иллюстраций . . . . .	28
Список таблиц . . . . .	29

## Введение

Согласно "Инструкции по определению именованных сущностей"<sup>1</sup>, именованной сущностью считается слово или словосочетание, предназначенное для конкретного, вполне определённого предмета или явления, выделяющее этот предмет или явление из ряда однотипных предметов или явлений. Примерами именованных сущностей являются имена людей, названия организаций и локаций. Задача распознавания именованных сущностей (Named Entity Recognition, NER) состоит в выделении и классификация именованных сущностей в тексте. В рамках конференции CoNLL 2003 проводилось соревнование для оценки качества методов распознавания именованных сущностей четырех типов на англоязычном корпусе [Tjong Kim Sang and De Meulder 2003]. Для решения задачи NER предлагалось много разных подходов [Nadeau and Sekine 2007]. В последнее время было показано, что методы на основе нейронных сетей показывают лучшие результаты для различных языков и корпусов, включая CoNLL 2003 [Yang et al. 2016].

Вместо большого количества вручную построенных признаков, нейросетевые методы используют универсальные векторные представления слов [Mikolov et al. 2013]. Согласно гипотезе о дистрибутивности, эти представления кодируют в себе смысл слов [Sahlgren 2008]. Это позволяет строить мультизадачные и языконезависимые архитектуры [Collobert et al. 2011, Yang et al. 2016].

Несмотря на то, что использование универсальных векторных представлений получило в последнее время огромную популярность в силу своей эффективности и огромной экономии человеческих усилий, большой интерес все еще представляет исследование возможностей использования высокоуровневых признаков в качестве входных данных для нейросетей. Так, например, в работах [Xu et al. 2014, Bian et al. 2014] описано использование морфологических, синтаксических и семантических признаков для построения более совершенных векторных представлений слов.

---

<sup>1</sup><http://opencorpora.org/wiki/Nermanual/1>

Compreno - это технология автоматического анализа текстов на естественном языке, в основе которой лежит многоуровневое лингвистическое описание, создававшееся профессиональными лингвистами в течение длительного времени [Anisimovich et al. 2012]. Помимо ручного описания Compreno использует для анализа большое количество информации, извлекаемой различными статистическими методами из текстовых корпусов. В Compreno реализована процедура семантико-синтаксического анализа текста, в результате которой любому предложению на естественном языке (английском или русском) ставится в соответствие семантико-синтаксическое дерево, моделирующее смысл предложения и содержащее грамматическую и семантическую информацию о каждом слове предложения.

В данной работе исследована возможность использования семантико-синтаксического анализатора Compreno в качестве источника высокоуровневых признаков для задачи NER на корпусе CoNLL 2003 в рамках нейросетевого подхода.

Данная работа организована следующим образом:

- в 1 части отражена постановка задачи, проведен обзор методов решения и обоснован выбранный метод решения задачи,
- во 2 части приведен алгоритм решения поставленной задачи и её программная часть,
- в 3 разделе описана экспериментальная часть.

Полученные результаты показывают повышение F1-меры почти на 1% на корпусе CoNLL 2003 при использовании синтактико-семантических признаков Compreno (87.49% против 88.47%). При этом затраты на их внедрение были минимальными - инженерия над признаками не проводилась.

# 1 Теоретико-аналитическая часть

## 1.1 Постановка задачи

Исследовать возможность использования семантико-синтаксического анализатора Comreno в качестве источника высокоуровневых признаков для задачи NER на корпусе CoNLL 2003 в рамках нейросетевого подхода.

## 1.2 Обзор литературы

Победители соревнования по NER CoNLL 2003 [Florian et al. 2003], получившие 88.76% F1, представили систему использующую комбинацию различных алгоритмов машинного обучения. В качестве признаков был использован их собственный, вручную составленный газетир, POS-теги, CHUNK-теги, суффиксы, префиксы и выход других NER-классификаторов, тренированных на внешних данных.

[Collobert et al. 2011] представили комбинацию сверточной нейронной сети с условными случайными полями, получившую 89.59% F1 на корпусе CoNLL 2003. Их нейросетевая архитектура не зависит от задачи и используется как для NER, так и для частеречной разметки (part-of-speech tagging), поиска синтаксически связанных групп соседних слов (chunking), установления семантических ролей (semantic role labelling). Для задачи NER они использовали три типа признаков - векторное представление слова, капитализацию и небольшой газетир, включенный в соревнование CoNLL 2003.

[Chiu and Nichols 2015] представили комбинацию сверточных сетей, рекуррентных сетей и условных случайных полей показывающую 91.62% F1. Они использовали такие же признаки как у [Collobert et al. 2011], дополнительный, вручную сформированный газетир на основе DBpedia и обучались на train+dev<sup>1</sup> выборке CoNLL 2003. Кроме корпуса CoNLL 2003 они тестировали архитектуру на более крупном англоязычном корпусе OntoNotes 5.0. На нем они получили state-of-the-art результат 86.28%.

---

<sup>1</sup>Объединенная обучающая и валидационная выборки

[Yang et al. 2016] представили глубокую иерархическую рекуррентную нейросетевую архитектуру с условными случайными полями для разметки последовательностей. Они использовали такие же признаки как у [Collobert et al. 2011]. Кроме англоязычного корпуса CoNLL 2003, где они получили state-of-the-art 90.94% F1 при обучении только на обучающей выборке (train set), они тестировали работу нейросети на CoNLL 2002 Dutch NER и CoNLL 2003 Spanish NER. На этих корпусах они улучшили предыдущий state-of-the-art результат: 82.82% до 85.19% на CoNLL 2002 Dutch NER и 85.75% до 85.77% на CoNLL 2003 Spanish NER.

Современные работы используют векторное представление слов и условные случайные поля в своих моделях. Из сторонних признаков применяют только газетеры. В работах [Xu et al. 2014, Bian et al. 2014] описано применение дополнительных признаков для слов (морфологических, синтаксических, семантических) для создания более совершенных векторных представлений. Такие векторные представления помогают повысить оценку качества в прикладных задачах [Xu et al. 2014].

### **1.3 Обзор корпусов для задачи NER**

#### **1.3.1 Схемы разметки корпуса**

### **1.4 Обзор нейросетевого подхода к решению задачи NER**

Традиционным подходом к решению задач из области автоматической обработки текстов, включая NER, является использованием алгоритма обучения с учителем, например машины опорных векторов с линейным ядром. Вручную составленные признаки подаются на вход алгоритма обучения с учителем. Выбор признаков - это практически полностью эмпирический процесс, построенный на лингвистической интуиции и решаемой задаче.

Нейросетевой подход предполагает использование минимального количества признаков. Обычно это векторные представления слов полу-

ченные из большого корпуса с использованием алгоритмов обучения без учителя.

В этом разделе будет рассмотрен нейросетевой подход из работы [Collobert et al. 2011]. Была выбрана именно эта работа так как:

- в ней представлена модель, получающая сравнимую со state-of-the-art F1-меру на CoNLL 2003,
- представленная модель имеет различные программные имплементации,
- векторные представления слов под названием «Senna Embeddings» используемые в этой работе находятся в открытом доступе в сети.

Последние два пункта указывают на возможность воспроизведения результатов из статьи относительно небольшими усилиями.

#### 1.4.1 Векторные представления слов

Согласно Википедии<sup>1</sup>, *векторное представление* — это общее название для различных подходов к моделированию языка и обучению представлений в обработке естественного языка, направленных на сопоставление словам (и, возможно, фразам) из некоторого словаря векторов из  $R^n$ , где  $n$  - значительно меньше количества слов в словаре (обычно от 50 до 1000).

[Collobert et al. 2011] использовали нейронные сети для построения векторных представлений слов. Они тренировали нейронную сеть на данных корпусе английской Википедии<sup>2</sup> и Reuters RCV1<sup>3</sup>. Были использованы 130000 самых частотных слов, остальные слова кодировались специальным токеном UNKNOWN. Полученные векторные представления называются Senna Embeddings и доступны в сети по адресу: <http://ronan.collobert.com/senna/>.

Senna Embeddings использовались как входные признаки на нейронные сети для решения задачи NER.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Word\\_embedding](https://en.wikipedia.org/wiki/Word_embedding)

<sup>2</sup>На данных ноября 2007 года из <http://download.wikimedia.org>

<sup>3</sup>Доступно <http://trec.nist.gov/data/reuters/reuters.html>

### **1.4.2 Описание нейросетевых моделей**

В работе [Collobert et al. 2011] описываются две нейросетевые модели:

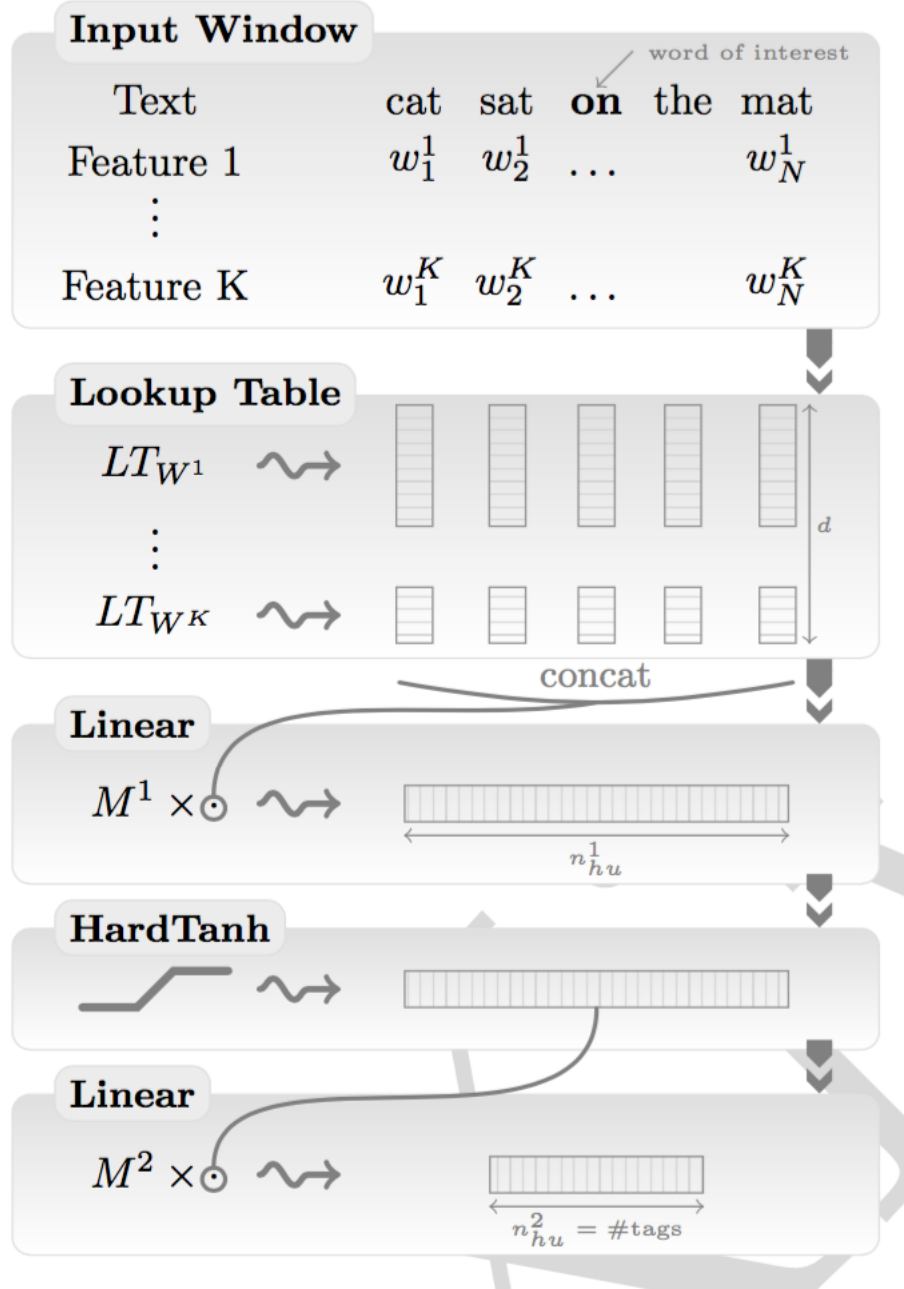
- Оконный (window), который предсказывает тег слова на основе контекста (окна) вокруг слова,
- сверточный (convolution), который предсказывает тег слова используя всё предложение.

#### **1.4.2.1 Оконная модель**

Оконная модель представлена на рис. 1.1.



Рисунок 1.1 — Оконная модель из [Collobert et al. 2011]



На рис. 1.1 предсказывается тег для слова on. При классификации используется контекст для этого слова. Всё окно состоящее из 5 слов пропускается через так называемый Lookup Table.

*Lookup Table* - это специальный слой в нейронной сети, который отображает каждое слово в вектор весов. Причем вектор весов обучается вместе с сетью. Более формально, каждому слову  $w$  из словаря  $D$  ставится в соответствие вектор размерности  $d$ , который задается Lookup

Table слоем  $LT_W(w)$ :

$$LT_W(w) = W_{\cdot, w},$$

где  $W \in R^{d \times |D|}$  - матрица весов для обучения,  $W_{\cdot, w}$  -  $w$ -ый столбец матрицы  $W$ . Размерность  $d$  - гиперпараметр.

Также этот слой может принимать на вход последовательность слов  $w_1 \dots w_K$ . В этом случае выходом будет матрица:

$$f_1 = LT_W(w_1 \dots w_K) = (W_{\cdot, w_1} \dots W_{\cdot, w_K})$$

После Lookup Table слоя, полученная матрица преобразуется в один вектор с помощью операции *конкатенации*  $Concat$ :

$$f_2 = Concat(LT_W(w_1 \dots w_K)) = \begin{bmatrix} W_{\cdot, w_1} \\ \vdots \\ W_{\cdot, w_K} \end{bmatrix},$$

Затем этот вектор подается на *полносвязный слой* ( $Linear Layer$ ), который выполняет аффинное преобразование:

$$f_3 = Linear(f_2) = W^2 f_2 + b^2, \quad (1.1)$$

где  $W^2 \in R^{d_2 \times |f_2|}$ ,  $b \in R^{d_2}$ . Гиперпараметр  $d_2$  - это количество нейронов в данном слое.

Каждый элемент вектора пропускается через нелинейную функцию. На рис. 1.1 это  $HardTanh$ :

$$f_4 = HardTanh(f_3),$$

$$HardTanh(x) = \begin{cases} -1, & \text{if } x < -1 \\ x, & \text{if } -1 \leq x \leq 1 \\ 1, & \text{if } x > 1 \end{cases} \quad (1.2)$$

Преимуществом этой функции перед гиперболическим тангенсом является более быстрое время вычисления.

Последний слой является полносвязный слой. Количество нейронов в нем равно количеству предсказываемых классов.

$$f_5 = Linear(f_4) = W^5 f_4 + b^5.$$

Каждый элемент полученного вектора пропускается через функцию *Softmax* для получения вероятностей:

$$Softmax(x_i) = \frac{e_i^x}{\sum_j e_j^x}$$

Другими словами это нормализация вектора с помощью экспоненциальной функции.

Для обучения нейронной сети минимизируется функционал ошибки с использованием *пакетного градиентного спуска* (*mini-batch gradient descent*). Функционал ошибки  $C$ :

$$C = - \sum_{(x,y_k) \in T} \log(P(y_k|x, \theta)),$$

$$C \rightarrow \min_{\theta},$$

$$\log(P(y_k|x, \theta)) = f_4(y_k) - \log \sum_i e^{f_4(y_i)},$$

где  $(x, y_k)$  - пара объект, класс из обучающей выборки  $T$ ;  $\theta$  - веса всей нейросети (все матрицы  $W$ ),  $f_4(y_i)$  - значение последнего слоя для класса  $y_i$ .

Функционал ошибки  $C$  минимизируется с помощью пакетного градиентного спуска:

$$\theta = \theta - \alpha \sum_{(x,y_k) \in T_s} \nabla_{\theta}(-\log P(y_k|x, \theta)),$$

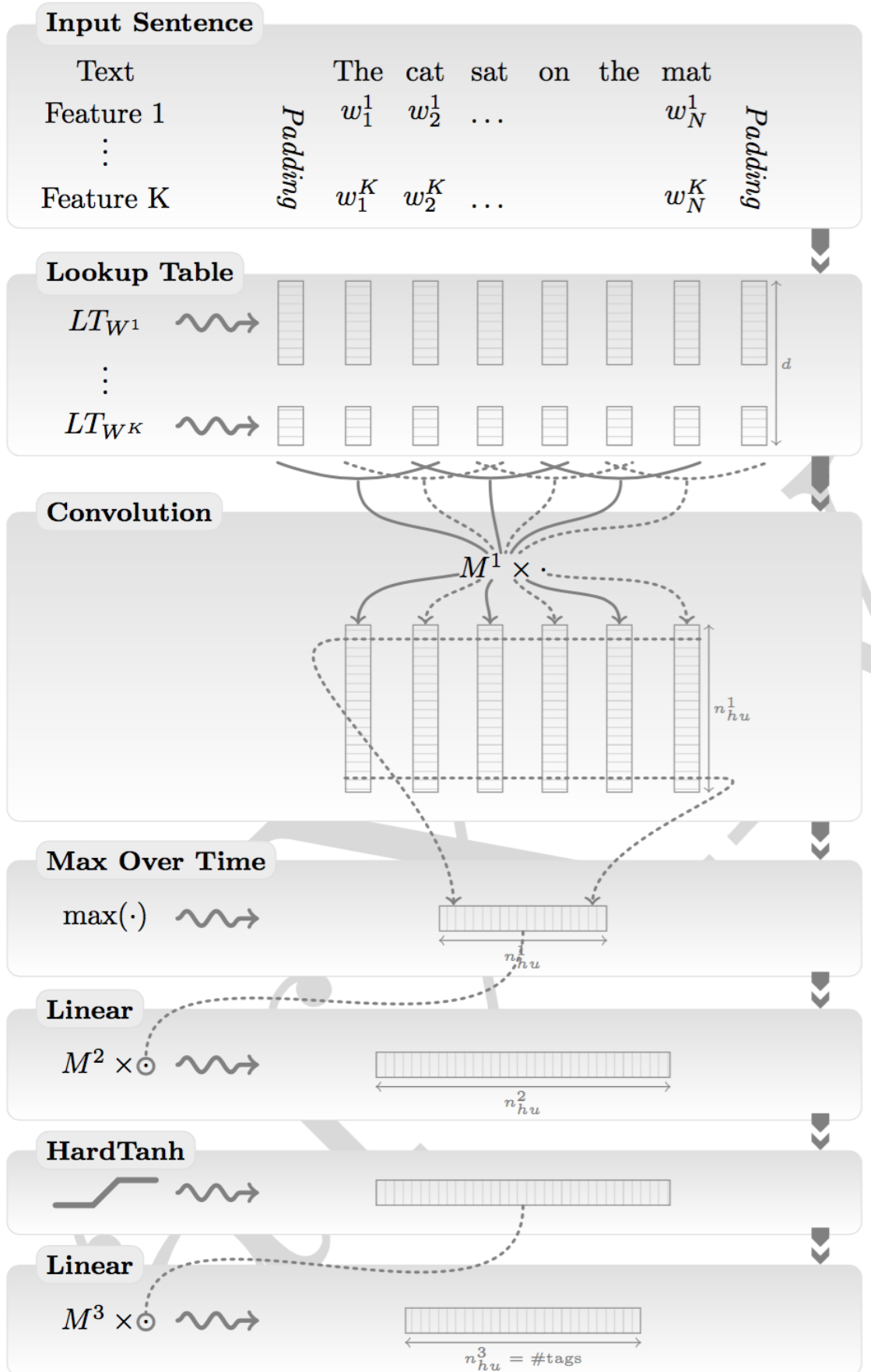
где  $\alpha$  - шаг обучения,  $T_s$  - случайное подмножество объектов из обучающей выборки. Размер подмножества  $T_s$  и  $\alpha$  являются гиперпараметрами.

*Стохастического градиентный спуск* (*stochastic gradient descent*) является модификацией пакетного градиентного спуска. Отличие заключается в том, что размер  $T_s$  равен 1.

#### 1.4.2.2 Сверточная модель

Сверточная модель представлена на рис. 1.2.

Рисунок 1.2 — Сверточная модель из [Collobert et al. 2011]



В отличие от оконной модели, она принимает на вход всё предложение.

Все слова предложения пропускаются через Lookup Table, как описано в разделе 1.4.2.1, после чего попадают на сверточный слой.

*Сверточный слой (temporal convolution)* является обобщением полносвязного слоя (1.1) из оконного подхода:

а) сначала осуществляется проход окном на полученной на предыдущем шаге матрице  $f_1$ ,

б) столбцы попавшие в окно конкатенируются,

в) полученный вектор пропускается через полносвязный слой, причем матрица весов  $W^3$  является одной и той же в каждом проходе.

На выходе, после прохода окном по всей матрице  $f_1$ , получается матрица  $f_3$ . Более формально:

$$f_3 = (W^3 f_1[1, d_{win}] + b^3, \dots, W^3 f_1[|f_1| - d_{win}, d_{win}] + b^3),$$

$$d_{f_1, win} = d_{win}d,$$

где  $f_1[i, d_{win}]$  - означает конкатенацию столбцов  $f_1$  с  $i$  по  $i + d_{win}$ ,  $d_{f_1, win}$  - размерность столбцов после конкатенации,  $W^3 \in R^{d_2 \times d_{f_1, win}}$ ,  $b \in R^{d_2}$ . Гиперпараметр  $d_2$  - это количество нейронов в данном слое.

После сверточного слоя получается матрица  $f_3 \in R^{d_2 \times |f_1| - d_{win}}$ . Количество строк в ней фиксировано, но количество столбцов зависит от длины предложения. Чтобы получить вектор признаков фиксированной длины выполняется *операция получения максимума по строкам (Max over time)* над  $f_3$ . Смысл этой операции заключается в получении наиболее значимых признаков из каждого окна. В результате этой операции получается вектор  $f_5 \in R^{d_2}$ .

Далее над полученным фиксированным вектором признаков проводятся уже знакомые по разделу 1.4.2.1 операции с полносвязными слоями (1.1) и нелинейной функцией HardTanh (1.2).

Для обучения используется минимизация функционала ошибки и пакетный градиентный спуск описанный в разделе 1.4.2.1.

В оригинальной статье [Collobert et al. 2011] в сверточном подходе минимизировался другой функционал, который включает в себя условные случайные поля. Это позволило им достичь более высокой оценки качества, но в то же время замедлило время проведения экспериментов. Данный функционал не поддерживается библиотеками для работы с нейронными сетями, поэтому его необходимо реализовывать с нуля. Из-за этих причин данный функционал ошибки не рассмотрен в этой работе.

### 1.4.3 Выбор метода решения задачи

Лучший результат для задачи NER на CoNLL 2003 показывает комбинация нейросети и условных случайных полей. Они довольно сложны в реализации с нуля, т.к. не поддерживаются существующими фреймворками.

Условные случайные поля сами по себе показывают неплохой результат, но долго обучаются и требуют сложную инженерию признаков.

Нейронные сети показывают достаточно хороший результат, быстро обучаются, расширяемы и имеют различные Open Source библиотеки, поддерживаемые сообществом и крупными компаниями. Исходя из цели работы был выбран нейросетевой подход. В перспективе в него можно внедрить условные случайные поля для получения state-of-the-art результата.

## 1.5 Синтактико-семантические признаки

Существует много инструментов для получения дополнительных признаков для слова. Для извлечения синтаксических признаков часто используют MaltParser [Nivre et al. 2006]. Для получения семантических признаков применяют BabelNet [Navigli and Ponzetto 2010].

В данной работе для получения синтактико-семантических признаков используется Compreno. Вершины синтактико-семантического дерева Compreno кодировались бинарным представлением и соотносились с токенами исходного текста<sup>1</sup>, тем самым наделяя их синтактико-семан-

---

<sup>1</sup>Почти для всех токенов в соответствующем дереве нашлась соответствующая вершина. Токены для которых не была найдена вершина, кодировались специальным признаком 83951

тическими признаками. Размерность пространства синтактико-семантических признаков получилась равной 83950.

Плотные вектора большой размерности сильно замедляют процесс оптимизации и для хорошего обучения требуется много данных и вычислительных ресурсов. В таких случаях часто применяют методы для уменьшения размерности, например сингулярное разложение или автоэнкодеры. Минусом таких методов является потеря информации после сжатия.

Если же вектора большой размерности разреженные, то используют специальные методы для работы с такими данными [Davis et al. 2016].

В данной работе предлагается 2 способа внедрения синтактико-семантических признаков:

- сжать синтактико-семантические вектора с помощью сингулярного разложения (SVD) и добавить как еще один Lookup Table в сверточную нейронную сеть;
- добавить еще одну нейронную сеть для синтактико-семантических признаков и оптимизировать её вместе со сверточной нейронной сетью.

## 2 Проектная часть

### 2.1 Алгоритмизация выбранного метода решения задачи

Для реализации была выбрана сверточная нейронная сеть из статьи [Collobert et al. 2011]. Из модели были удалены условные случайные поля для более быстрого обучения и проведения экспериментов.

В качестве признаков выступают вектора слов, позиция относительно слова в предложении для которого предсказывается тег, капитализация и присутствие слова в газетире, который включен в соревнование CoNLL 2003.

Общий алгоритм работы следующий:

а) В начало и конец предложения добавляют по одному специальному токenu, чтобы его длина была не меньше трех. Каждый токен отображается в набор идентификаторов признаков. На вход нейросети поступает набор идентификаторов признаков для всего предложения.

б) Набор идентификаторов пропускается через специальный слой (lookup table), который отображает каждый идентификатор в вектор весов. На выходе каждому токenu предложения соответствует вектор.

в) Полученные вектора объединяются в матрицу признаков предложения.

г) Полученная матрица признаков подается на следующий слой, который проходится окном размера 3 и выполняет операцию свертки (temporal convolution). Т.е. три столбца матрицы признаков конкатенируются в один вектор и перемножаются на матрицу весов справа. На выходе получается матрица с количеством строк фиксированной длины (для любого предложения).

д) Затем извлекается максимум по каждой строке (max over time). Таким образом все предложения любой длины получают вектор признаков фиксированной длины.

е) Полученный вектор признаков подается на полносвязный слой.

ж) Затем выход полносвязного слоя подается на выходной слой, который возвращает вероятность для каждого тега (softmax).



В качестве функции потерь используется кросс-энтропия (cross entropy).

Подробная математическая модель описана в статье [Collobert et al. 2011].

## **2.2 Программная часть**

### **2.2.1 Выбор программного фреймворка**

### **2.2.2 Реализованная модель**

Нейронная сеть написана с использованием открытого фреймворка torch<sup>1</sup>.

Код для воспроизведения экспериментов выложен по адресу: [github.com/sld/torch-conv-ner](https://github.com/sld/torch-conv-ner).

Скорость обучения на машине с GPU Amazon AWS g2.2xlarge<sup>2</sup>:

- 1 эпоха при одиночной обработке (stochastic gradient descent): ~450 сек.
- 1 эпоха при пакетной обработке (mini-batch gradient descent): ~171 сек.
- Модель получающая 87.49% обучалась 91 эпоху (~4.2 часа).
- 1 эпоха при пакетной обработке с использованием признаков Compreno: ~615 сек.

Скорость классификации составляет 2500 токенов в секунду при пакетной обработке.

---

<sup>1</sup><http://torch.ch>

<sup>2</sup><https://aws.amazon.com/ru/ec2/instance-types/>

### 3 Экспериментальная часть

#### 3.1 Корпус CoNLL 2003

CoNLL 2003 [Tjong Kim Sang and De Meulder 2003] - англоязычный корпус для оценки качества методов распознавания именованных сущностей. Корпус содержит обучающую, тестовую и валидационную выборку. Размечено 4 типа сущностей - персоны (PER), организации (ORG), локации (LOC) и другие (MISC). Корпус размечен по схеме Inside, Outside, Begin (IOB). Оценка качества считается с помощью F1-micro-average.

Таблица 3.1 — Количество статей, предложений, токенов и именованных сущностей

Выборка	Статьи	Предложения	Токены	LOC	MISC	ORG	PER
Обучающая	946	14987	203621	7140	3438	6321	6600
Валидационная	216	3466	51362	1837	922	1341	1842
Тестовая	231	3684	46435	1668	702	1661	1617

Как и у [Collobert et al. 2011], данные были сконвертированы из схемы IOB в схему IOBES (Inside, Outside, Begin, End, Single). Во время тестирования, данные конвертируются обратно в формат IOB и подаются на вход скрипта, включенного в CoNLL 2003, оценивающего качество классификации.

#### 3.2 Синтактико-семантические признаки Comreno

Синтактико-семантически признаки были получены с помощью Comreno. Они представляют собой разреженные вектора размерности 83950. Они покрывают около 60% корпуса CoNLL 2003. Все токены, не покрытые Comreno, кодировались как дополнительный признак 83951.

#### 3.3 Эксперименты без синтактико-семантических признаков

Нейросетевая модель имеет такие же параметры как и у [Collobert et al. 2011]. Небольшой модификацией является добавление Dropout слоя

в качестве регуляризатора, после каждого полносвязного слоя. Размерность выходного слоя - 17. Четыре для каждого из четырех типов тегов и один для Outside.

В качестве векторного представления слов (embeddings в таблице 3.2), использовались Senna embeddings<sup>1</sup>, которые находятся в открытом доступе.

---

<sup>1</sup><http://ronan.collobert.com/senna/>

Таблица 3.2 — Результаты экспериментов без использования синтактико-семантических признаков

Модель	Признаки	Выборка	Метод оптимизации	Полученная F1, %	F1 в статье Collobert et al. [2011]
Window	Embeddings, Capitalization	train	Mini-batch gradient descent	86.27	-
Window	Embeddings, Capitalization	train	Stochastic gradient descent	-	86.97
ConvNet + CRF	Embeddings, Capitalization, Position	train	Stochastic gradient descent	-	88.67
ConvNet + CRF	Embeddings, Capitalization, Position, Gazetteer	train	Stochastic gradient descent	-	89.59
ConvNet	Embeddings, Capitalization, Position	train	Stochastic gradient descent	86.77	-
ConvNet	Embeddings, Capitalization, Position, Gazetteer	train	Stochastic gradient descent	87.89	-
ConvNet	Embeddings, Capitalization, Position, Gazetteer	train + dev	Stochastic gradient descent	88.37	-
ConvNet	Embeddings, Capitalization, Position, Gazetteer	train	Mini-batch gradient descent	<b>87.49</b>	-

По таблице 3.2 видно, что результаты немного ниже чем у [Collobert et al. 2011]. Это связано с тем, что для Window подхода ис-

пользован другой метод оптимизации, а для Convolution подхода не были применены условные случайные поля.

В качестве референсной, будет использована модель из последнего эксперимента показывающая 87.49% F1. Это сделано для чистоты эксперимента, т.к. далее обучение происходило только на обучающей выборке по правилам соревнования CoNLL 2003 и применялся mini-batch gradient descent для ускорения экспериментов.

### **3.4 Эксперименты с синтактико-семантическими признаками сжатыми с использованием SVD**

Синтактико-семантически признаки Compreno размерности 83950 были сжаты с использованием TruncatedSVD<sup>1</sup> до размерности 1024. После сжатия описываемая дисперсия была равна 72%. Т.е. потерялось 28% информации. Сжатые вектора были добавлены в нейронную сеть с помощью дополнительного Lookup Table слоя.

---

<sup>1</sup><http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>

Таблица 3.3 — Результаты с синтактико-семантическими признаками сжатыми SVD

Модель	Признаки	Выборка	Метод оптимизации	Полученная F1, %
ConvNet	Position, Compreno SVD 1024	train	Mini-batch gradient descent	75.89
ConvNet	Capitalization, Position, Gazetteer, Compreno SVD 1024	train	Mini-batch gradient descent	81.83
ConvNet	Embeddings, Capitalization, Position, Gazetteer, Compreno SVD 1024	train	Mini-batch gradient descent	86.85
ConvNet	Embeddings, Capitalization, Position, Gazetteer	train	Mini-batch gradient descent	87.49

По таблице 3.3 видно, что такой способ ведет к небольшому ухудшению F1-меры.

### 3.5 Эксперименты с синтактико-семантическими признаками для совместно-оптимизированной нейросети

Была добавлена еще одна нейронная сеть для синтактико-семантических признаков и оптимизирована вместе со сверточной нейронной сетью.

Нейронная сеть для синтактико-семантических признаков работает следующим образом:

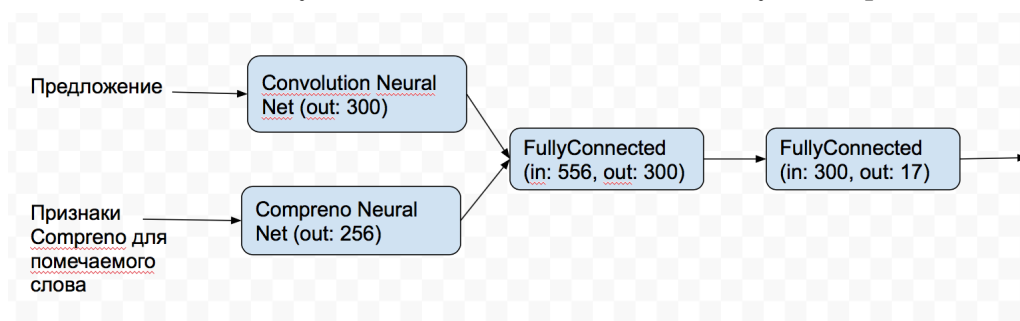
- а) На вход подается разреженный вектор признаков слова (размерность 83951) для которого предсказывается тег.
- б) Далее этот вектор пропускается через 2 полносвязных слоя.

в) На выходе еще один полносвязный слой, который выдает вероятность определенного тега. Выходов также 17.

Сверточная сеть, учитывающая всё предложение, и полносвязная сеть, обрабатывающая синтактико-семантические признаки слова для которого предсказывается тег, соединяются следующим образом (рис. 3.1):

- а) Из обеих нейросетей удаляются выходные слои.
- б) Предыдущие слои из обеих сетей соединяются в новый полносвязный слой.
- в) Новый полносвязный слой соединяется с выходным слоем. Выходов как и тегов 17.

Рисунок 3.1 — Объединение двух нейросетей



Веса у объединенной сети были инициализированы обученными моделями - моделью показывающую 87.49% для сверточной сети и моделью показывающую 72.85% (см. таблицу 3.4) для второй нейронной сети.

По таблице 3.4 видно, что признаки Comreno улучшают F1-меру почти на один процент.

Таблица 3.4 — Результаты с синтактико-семантическими признаками для объединенной нейросети

Модель	Признаки	Выборка	Метод оптимизации	Полученная F1, %
Compreno Net	Compreno sparse features	train	Mini-batch gradient descent	72.85
ConvNet	Embeddings, Capitalization, Position, Gazetteer	train	Mini-batch gradient descent	87.49
ConvNet + Compreno Net	Embeddings, Capitalization, Position, Gazetteer, Compreno sparse features	train	Mini-batch gradient descent	<b>88.47</b>
ConvNet + Compreno Net	Embeddings, Capitalization, Position, Gazetteer, Compreno sparse features	train + dev	Mini-batch gradient descent	88.81



## Заключение

В данной работе исследована возможность использования семантико-синтаксического анализатора Comrgeno в качестве источника высокоуровневых признаков для задачи NER на корпусе CoNLL 2003 в рамках нейросетевого подхода. Удалось найти простой вариант подключения признаков Comrgeno к сверточной нейронной сети за счет которого F1-мера повысилась с 87.49% до 88.47%.

В будущем планируется внедрить условные случайные поля в существующую модель для повышения F1-меры и исследовать работу предложенного решения на других корпусах. Также интересным направлением для исследований является создание векторных представлений слов с учетом синтактико-семантических признаков.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- Anisimovich, K. V., Druzhkin, K. J., Minlos, F. R., Petrova, M. A., Selegey, V. P., and Zuev, K. A. (2012). Syntactic and semantic parser based on ABBYY Compreno linguistic technologies. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference Dialog*, page 18.
- Bian, J., Gao, B., and Liu, T.-Y. (2014). Knowledge-powered deep learning for word embedding. In *Machine Learning and Knowledge Discovery in Databases*, pages 132–148. Springer.
- Chiu, J. P. and Nichols, E. (2015). Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.
- Davis, T. A., Rajamanickam, S., and Sid-Lakhdar, W. M. (2016). A survey of direct methods for sparse linear systems.
- Florian, R., Ittycheriah, A., Jing, H., and Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 168–171. Association for Computational Linguistics.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.
- Navigli, R. and Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting*

*of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics.

Nivre, J., Hall, J., and Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219.

Sahlgren, M. (2008). The distributional hypothesis. from context to meaning: Distributional models of the lexicon in linguistics and cognitive science (special issue of the italian journal of linguistics). *Rivista di Linguistica*, 20(1).

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Xu, C., Bai, Y., Bian, J., Gao, B., Wang, G., Liu, X., and Liu, T.-Y. (2014). Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1219–1228. ACM.

Yang, Z., Salakhutdinov, R., and Cohen, W. (2016). Multi-task cross-lingual sequence tagging from scratch. *CoRR*, abs/1603.06270.

## Список иллюстраций

1.1	Оконная модель из [Collobert et al. 2011] . . . . .	9
1.2	Сверточная модель из [Collobert et al. 2011] . . . . .	12
3.1	Объединение двух нейросетей . . . . .	23

## Список таблиц

3.1	Количество статей, предложений, токенов и именованных сущностей . . . . .	18
3.2	Результаты экспериментов без использования синтактико-семантических признаков . . . . .	20
3.3	Результаты с синтактико-семантическими признаками сжатыми SVD . . . . .	22
3.4	Результаты с синтактико-семантическими признаками для объединенной нейросети . . . . .	24