

Влияние токенизации на оценку качества нейросетевого синтаксического анализа

Шамаева Елена Денисовна¹[0009–0002–6233–8958], Лукашевич Наталья Валентиновна¹[0000–0002–1883–4121]

МГУ имени М.В. Ломоносова

Аннотация Синтаксические анализаторы используются в качестве вспомогательного инструмента в разных областях автоматической обработки текста. Поэтому важными направлениями исследований являются разработка критериев выбора синтаксического анализатора для конкретной прикладной задачи и методология оценки качества синтаксического анализатора. На оценку качества синтаксического анализатора влияет этап токенизации. Существует два способа оценки синтаксического анализатора: с использованием встроенного токенизатора и с использованием токенизатора, возвращающего эталонную разметку. Данная статья посвящена сравнению этих способов оценки качества синтаксического анализа. Исследование проведено на русскоязычных корпусах предложений с синтаксической разметкой SynTagRus, GSD, PUD, Taiga, Poetry и для русскоязычных синтаксических анализаторов UDPipe, Stanza, Natasha, DeepPavlov и spacy. Выявлено, что для значимого количества предложений разделение на токены, проводимое встроенным токенизатором, отличается от эталонного. Установлено также, что средние значения метрик UAS и LAS выше при использовании токенизатора, возвращающего эталонную разметку. Разработанная методология описания категорий токенов может использоваться для проверки качества синтаксического анализа при внедрении нового токенизатора. В рамках данного исследования для каждого из рассматриваемых анализаторов реализован токенизатор, возвращающий эталонный набор токенов из датасета. Реализация исследования доступна по адресу: https://github.com/Derinhelm/parser_stat/tree/tokenization_changing.

Keywords: Синтаксический анализ · Нейронные сети · Токенизация.

1 Введение

В эпоху больших языковых моделей классические нейросетевые синтаксические анализаторы (СА) используются для разработки интерпретируемых методов решения задач автоматической обработки текстов. В частности, для оценки сложности текста [1], распознавания именованных сущностей [2,3,4,5], выявления плагиата [6], перефразирования [7]. При проектировании такой системы автоматической обработки текста выбор синтакси-

ческого анализатора осуществляется на основе информации, полученной из исследований по сравнению синтаксических анализаторов.

Существующие исследования по сравнению синтаксических анализаторов различаются с точки зрения проведения предварительного этапа обработки текста, токенизации. На этапе токенизации происходит выделение токенов предложения (слов, знаков препинания и т.п.). Существует два способа оценки качества синтаксического анализатора: с использованием встроенного токенизатора¹ (BT) и с использованием токенизатора, возвращающего эталонный набор токенов из датасета (эталонный токенизатор, ЭТ).

Данная работа посвящена сравнению двух способов оценки качества синтаксического анализа с точки зрения используемого токенизатора. Экспериментальная оценка качества работы синтаксических анализаторов проведена на тестовых выборках русскоязычных датасетов предложений с синтаксической разметкой из проекта Universal Dependencies: GSD², PUD³, SynTagRus⁴, Poetry⁵ и Taiga⁶. Для эксперимента выбраны синтаксические анализаторы UDPipe [8], Stanza [9], Natasha⁷, DeepPavlov⁸, spacy⁹. В рамках данного исследования для каждого из этих анализаторов был реализован токенизатор, возвращающий эталонный набор токенов из датасета. Ранее подобных исследований не проводилось.

2 Аналогичные работы

В большинстве исследований по сравнению синтаксических анализаторов применялся только один из подходов к токенизации. Эталонные токенизаторы использовались в соревнованиях синтаксических анализаторов CoNLL 2007 [10], CoNLL 2008 [11], встроенные — в соревнованиях CoNLL 2017 Shared Task [12], CoNLL 2018 Shared Task [13], GramEval 2020 Shared Task [14] и в проекте Naeval. В данной статье рассматриваются оба подхода.

В данной статье эксперименты проведены для синтаксических анализаторов, выбранных в самом масштабном проекте по сравнению русскоязычных АОТ-систем, проекте Naeval. В нем рассматриваются синтаксические анализаторы UDPipe 1.0, Stanza, Natasha, DeepPavlov, spacy. Среди этих анализаторов только анализатор UDPipe 1.0 принимал участие в соревнованиях для сравнения синтаксических анализаторов на текстах разных язы-

¹ Во многих системах автоматической обработки текста (например, Stanza, Natasha) встроенный токенизатор используется в режиме работы по умолчанию, но может быть заменен на токенизатор, реализованный пользователем.

² https://universaldependencies.org/treebanks/ru_gsd/index.html

³ https://universaldependencies.org/treebanks/ru_pud/index.html

⁴ https://universaldependencies.org/treebanks/ru_syntagrus/index.html

⁵ https://universaldependencies.org/treebanks/ru_poetry/index.html

⁶ https://universaldependencies.org/treebanks/ru_taiga/index.html

⁷ <https://natasha.github.io>

⁸ https://docs.deeppavlov.ai/en/master/features/models/syntax_parser.html

⁹ <https://spacy.io/api/dependencyparser>

ков: CoNLL 2007 [10], CoNLL 2008 [11], CoNLL 2017 Shared Task [12], CoNLL 2018 Shared Task [13], GramEval 2020 Shared Task [14].

На всех соревнованиях для оценки качества работы синтаксических анализаторов использовалась F1-мера метрик UAS и LAS с выравниванием, в проекте Naeval — F1-мера метрик UAS и LAS без выравнивания.

3 Методология исследования

3.1 Категории токенов дерева зависимостей

Одним из способов представления синтаксической структуры предложения является дерево зависимостей. Дерево зависимостей — это ориентированный граф вида дерево, в котором вершины соответствуют токенам предложения, а ребра — синтаксическим связям между токенами. Для каждого ребра предложения указан тип связи между соответствующими токенами. Корень дерева соединен с вспомогательным токеном ROOT (соответствующий тип связи — root). Каждому токenu соответствует ровно один главный токен (токен, от которого к данному токenu проведено ребро). Поэтому количество ребер в дереве зависимостей равно количеству токенов в предложении (без учета токена ROOT).

Для оценки на конкретном предложении качества работы синтаксического анализатора происходит сравнение эталонного дерева зависимостей и дерева зависимостей, построенного анализатором.

Набор токенов, полученный с помощью токенизатора, входящего в систему автоматической обработки текста (АОТ-систему), может отличаться от эталонного набора токенов. Например, в датасете SynTagRus фрагмент предложения «ИТ-компаний» рассматривается как 3 токена («ИТ», «-», «компаний»), а анализаторы Natasha и DeepPavlov рассматривают этот фрагмент как один токен.

Поэтому предварительным этапом оценки качества синтаксического анализа является выравнивание. На этом этапе происходит сопоставление токенов из эталонного дерева зависимостей и дерева зависимостей, построенного анализатором. Токены считаются совпадающими, если совпадают индексы их начала и конца в предложении.

При сравнении эталонного дерева зависимостей и дерева зависимостей, построенного СА с ВТ, для каждого токена из эталонного дерева зависимостей может быть получен один из следующих результатов:

1. в дереве зависимостей, построенном СА, нет соответствующего токена,
2. в дереве зависимостей, построенном СА, есть соответствующий токен, для которого неверно определен главный токен и тип связи (или главный токен отсутствует в эталонном дереве зависимостей),
3. в дереве зависимостей, построенном СА, есть соответствующий токен, для которого верно определен главный токен и неверно — тип связи,
4. в дереве зависимостей, построенном СА, есть соответствующий токен, для которого верно определен и главный токен, и тип связи.

В данной работе эти токены будут называться соответственно токенами 1, 2, 3 и 4 категорий. В дереве зависимостей, построенном СА, также присутствуют токены 2-4 категорий (эти токены совпадают в эталонном дереве зависимостей и в дереве зависимостей, построенном СА). Кроме того, в дереве зависимостей, построенном СА, есть токены, которые отсутствуют в эталонном дереве зависимостей. В данной работе эти токены будут называться «несопоставленные» токены. При использовании ЭТ набор токенов эталонного дерева зависимостей полностью совпадает с набором токенов дерева зависимостей, построенного СА. Следовательно, при сравнении эталонного дерева зависимостей и дерева зависимостей, построенного СА с использованием ЭТ, оба дерева зависимостей состоят только из токенов 2-4 категорий.

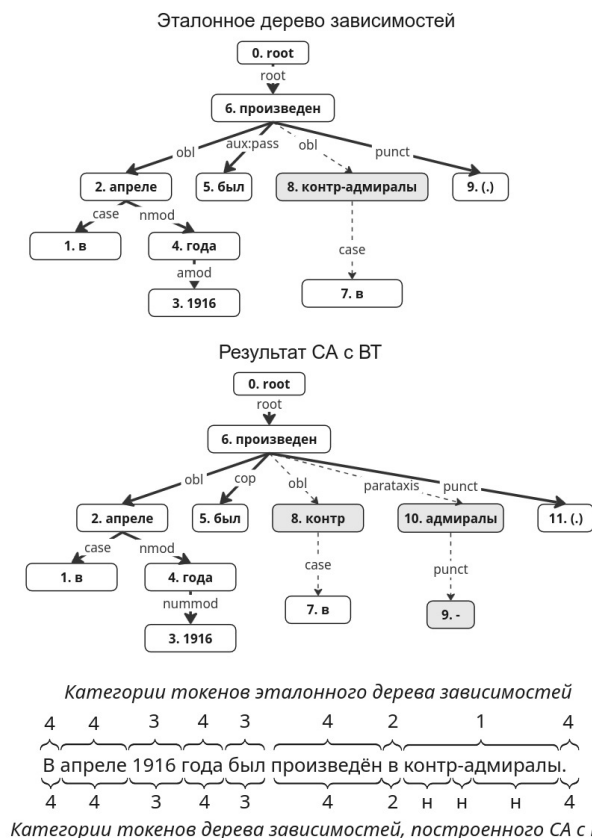


Рис. 1. Пример предложения с разными категориями токенов

На Рис. 1 приведен пример сравнения эталонного дерева зависимостей и дерева зависимостей, построенного СА с ВТ (цифры 1–4 соответствуют

токенам категорий 1–4, буква «н» — несопоставленным токенам). При разборе предложения «В апреле 1916 года был произведён в контр-адмиралы.» обнаружено несовпадение токенизации: в эталонном разбиении на токены фрагмент предложения «контр-адмиралы» рассматривается как 1 токен, ВТ выделил 3 токена («контр», «-», «адмиралы»). Эталонный токен «контр-адмиралы» относится к 1 категории, а соответствующие токены из дерева зависимостей, построенного СА, относятся к категории «несопоставленных» токенов. Для седьмого токена («в») в дереве зависимостей, построенном анализатором, главным токеном является токен, для которого нет аналога в эталонном дереве зависимостей. Поэтому он рассматривается как токен, для которого неверно определены главный токен и тип связи, и относится к 2 категории. Для токенов «1916», «был» верно определены главные токены и неверно определен тип связи (nummod вместо amod и cop вместо aux:pass). Эти токены относятся к 3 категории. Для остальных токенов верно определены и главный токен, и тип связи. Эти токены относятся к 4 категории.

3.2 Оценка качества синтаксического анализатора на предложении

На основе количества токенов разных категорий вычисляются метрики UAS (Unlabeled Attachment Score) и LAS (Labeled Attachment Score) [15], которые используются для оценки качества синтаксического анализа. Семантически метрика UAS соответствует доле токенов предложения, для которых верно определен главный токен, метрика LAS — доле токенов предложения, для которых верно определен главный токен и тип связи.

При вычислении этих метрик учитывается как набор токенов в эталонном дереве зависимостей, так и набор токенов в дереве зависимостей, созданном синтаксическим анализатором. Для этого по приведенным ниже формулам вычисляются точность (precision), полнота (recall) и F1-мера. В этих формулах n_1 , n_2 , n_3 , n_4 обозначают количество токенов категорий 1, 2, 3 и 4, n_p — количество «несопоставленных» токенов.

$$UAS_precision = \frac{n_3 + n_4}{n_p + n_2 + n_3 + n_4}, UAS_recall = \frac{n_3 + n_4}{n_1 + n_2 + n_3 + n_4}$$

$$UAS_F1 = \frac{2 * UAS_precision * UAS_recall}{UAS_precision + UAS_recall}$$

$$LAS_precision = \frac{n_4}{n_p + n_2 + n_3 + n_4}, LAS_recall = \frac{n_4}{n_1 + n_2 + n_3 + n_4}$$

$$LAS_F1 = \frac{2 * LAS_precision * LAS_recall}{LAS_precision + LAS_recall}$$

Для предложения на Рис. 1 $UAS_recall = 7/9$, $LAS_recall = 5/9$. В качестве знаменателя используется количество токенов в эталонном дереве зависимостей. В качестве числителя для UAS_recall используется количество

токенов категории 3 и 4, для LAS_recall — количество токенов категории 4 (на рисунке выше текста предложения). В рассматриваемом предложении $UAS_precision = 7/11$, $LAS_precision = 5/11$. В качестве знаменателя используется количество токенов в дереве зависимостей, построенном СА. В качестве числителя для $UAS_precision$ используется количество токенов категории 3 и 4, для $LAS_precision$ — количество токенов категории 4 (на рисунке ниже текста предложения).

3.3 Анализаторы и датасеты

В данном исследовании, как и в проекте Naeval, выбраны анализаторы UDPipe 1.0, Stanza, Natasha, DeepPavlov и spacy. На момент исследования анализатор UDPipe 2.0 не предоставляет доступ к русскоязычным моделям.

Данные анализаторы относятся к классическим нейросетевым синтаксическим анализаторам: на основе переходов [16]–[17] или на основе графов [18]. В анализаторах на основе переходов машинное обучение используется для предсказания преобразований, осуществляемых над предложением для получения его синтаксического разбора. В анализаторах на основе графов с помощью машинного обучения предсказываются веса ребер в полном графе из всех токенов предложения. Из этого графа с помощью алгоритма Чу-Лю/Эдмондса выделяется минимальное остовное дерево, которое и является результатом синтаксического анализа. В Табл. 1 представлена информация о видах анализаторов и датасетах, на которых они были обучены.

СА	Тип анализатора	Датасет, на котором обучен
UDPipe 1.0	на основе переходов	SynTagRus
Stanza	на основе графов	SynTagRus
Natasha	на основе графов	GramEval 2020 Shared Task, новости
DeepPavlov	на основе графов	SynTagRus
spacy	на основе переходов	GramEval 2020 Shared Task, новости

Таблица 1. Анализаторы

В качестве набора тестовых предложений используются тестовые выборки датасетов деревьев зависимостей из проекта Universal Dependencies: GSD, PUD, SynTagRus, Poetry, Taiga. В Табл. 2 приведена информация об источниках текстов и размерах тестовых выборок этих датасетов [19].

Датасет	Источник текстов	Размер
SynTagRus	Худ. проза и публицистика	8800
Taiga	Тексты эл. коммуникации	881
Poetry	Поэтические тексты	728
GSD	Русскоязычные тексты из энциклопедии Wikipedia	601
PUD	Переводы текстов из энциклопедии Wikipedia, новости	1000

Таблица 2. Датасеты

4 Результаты

4.1 Сравнение значений метрик UAS и LAS

Для всех анализаторов и датасетов при использовании ЭТ вместо ВТ средние значения метрик UAS и LAS увеличиваются или остаются на том же уровне. Подробная статистика приведена в Табл. 3 и Табл. 4¹⁰. Наибольшее изменение средних значений метрик происходит для анализатора DeepPavlov и датасета GSD (0.08 по метрике UAS и 0.07 по метрике LAS).

	T		Po		G		PU		S		Сред.	
	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ
natasha	0.70	0.73	0.64	0.65	0.79	0.83	0.88	0.89	0.83	0.84	0.77	0.79
udpipe	0.73	0.78	0.72	0.77	0.79	0.83	0.86	0.89	0.88	0.89	0.80	0.83
spacy	0.77	0.79	0.75	0.77	0.84	0.88	0.91	0.93	0.87	0.88	0.83	0.85
deeppavlov	0.79	0.84	0.84	0.86	0.83	0.91	0.94	0.94	0.92	0.94	0.86	0.90
stanza	0.79	0.84	0.82	0.87	0.85	0.89	0.93	0.93	0.94	0.94	0.86	0.90

Таблица 3. Средние значения метрики UAS

	T		Po		G		PU		S		Сред.	
	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ
natasha	0.64	0.66	0.58	0.59	0.75	0.78	0.84	0.84	0.78	0.79	0.72	0.73
udpipe	0.66	0.69	0.65	0.70	0.71	0.73	0.79	0.81	0.84	0.84	0.73	0.75
spacy	0.70	0.72	0.69	0.71	0.80	0.83	0.87	0.88	0.82	0.83	0.78	0.80
deeppavlov	0.72	0.75	0.78	0.80	0.75	0.82	0.86	0.87	0.89	0.91	0.80	0.83
stanza	0.72	0.76	0.76	0.81	0.79	0.82	0.87	0.87	0.91	0.92	0.81	0.84

Таблица 4. Средние значения метрики LAS

Выбор токенизатора существенным образом влияет на определение «наилучшего» синтаксического анализатора. В частности, на датасете GSD при использовании ВТ самое высокое значение метрики UAS показал анализатор Stanza, а при использовании ЭТ — анализатор DeepPavlov.

При использовании ЭТ уменьшается количество предложений со значениями метрик UAS и LAS, равными 0.0, и увеличивается количество предложений со значениями метрик 1.0. Подробная статистика приведена в Табл. 5, Табл. 6, Табл. 7, Табл. 8. В данных таблицах жирным шрифтом выделены значения с наилучшим результатом для датасета и анализатора¹¹.

Замена ВТ на ЭТ влияет на значения метрик UAS и LAS. Это происходит, поскольку при замене ВТ на ЭТ меняются как категории токенов, так и общее количество токенов в построенном дереве зависимостей. Изменение

¹⁰ Для каждого синтаксического анализатора и способа токенизации жирным шрифтом выделены наибольшие значения, с помощью подчеркивания выделены значения метрик, полученные при использовании ЭТ, которые на 0.05 и более превосходят значения аналогичных метрик, полученных при использовании ВТ.

¹¹ В Табл. 5, Табл. 7 не выделены цветом столбцы датасетов GSD и PUD (из-за небольшого количества рассматриваемых предложений).

	taiga		poetry		gsd		pud		syntagrus	
	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ
natasha	24	12	29	23	1	1	0	0	31	23
udpipe	24	11	17	12	2	1	0	0	21	18
spacy	18	12	31	27	1	1	0	0	29	23
deeppavlov	23	9	6	3	8	1	0	0	31	10
stanza	12	8	6	4	1	1	0	0	16	12

Таблица 5. Количество предложений с UAS=0.0

	taiga		poetry		gsd		pud		syntagrus	
	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ
natasha	34	23	33	27	2	2	0	0	35	28
udpipe	30	17	20	13	3	2	0	0	25	23
spacy	23	17	33	28	1	1	0	0	35	30
deeppavlov	26	13	7	4	9	2	0	0	35	14
stanza	14	10	7	5	3	1	0	0	17	13

Таблица 6. Количество предложений с LAS=0.0

	taiga		poetry		gsd		pud		syntagrus	
	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ
natasha	225	237	131	137	121	127	281	286	2397	2443
udpipe	246	296	192	232	137	144	282	293	3375	3431
spacy	288	312	201	219	177	192	376	399	3159	3298
deeppavlov	314	359	284	299	221	236	453	464	4472	4760
stanza	294	360	239	311	179	198	400	415	4708	4860

Таблица 7. Количество предложений с UAS=1.0

	taiga		poetry		gsd		pud		syntagrus	
	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ	BT	ЭТ
natasha	147	149	103	105	77	79	166	168	1601	1611
udpipe	159	180	137	166	46	46	109	113	2279	2303
spacy	199	205	152	164	119	124	226	237	1920	1994
deeppavlov	194	207	189	197	77	82	179	181	3247	3376
stanza	184	216	173	219	85	89	192	195	3606	3725

Таблица 8. Количество предложений с LAS=1.0

категорий токенов влияет на числители точности и полноты метрик UAS и LAS, изменение количества токенов — на знаменатель точности. Соответственно меняются и значения F-меры этих метрик. В дальнейшем планируется проведение статистического анализа влияния изменения категорий токенов и количества токенов на изменение значений метрик UAS и LAS.

4.2 Изменение метрик на отдельных предложениях

При замене ВТ на ЭТ может происходить как увеличение значения метрик UAS и LAS, так и уменьшение. Соответствующие примеры приведены в приложении А. Количество предложений, для которых значения метрик UAS и LAS отличаются в зависимости от используемого токенизатора, не превышает 40%. На большинстве этих предложений значения метрик выше при использовании ЭТ. Подробная статистика приведена в Табл. 9 и Табл. 10.

	taiga		poetry		gsd		pud		syntagrus	
	B > Э	B < Э	B > Э	B < Э	B > Э	B < Э	B > Э	B < Э	B > Э	B < Э
natasha	2%	17%	2%	5%	1%	19%	1%	5%	1%	7%
udpipe	4%	23%	7%	30%	5%	29%	2%	16%	0%	3%
spacy	8%	19%	8%	16%	4%	25%	3%	13%	8%	15%
deeppavlov	1%	24%	0%	10%	0%	17%	0%	3%	1%	10%
stanza	3%	22%	2%	35%	2%	32%	0%	5%	0%	4%

Таблица 9. Количество предложений с изменениями UAS (B — ВТ, Э — ЭТ)

	taiga		poetry		gsd		pud		syntagrus	
	B > Э	B < Э	B > Э	B < Э	B > Э	B < Э	B > Э	B < Э	B > Э	B < Э
natasha	3%	16%	2%	5%	1%	19%	1%	5%	1%	7%
udpipe	4%	23%	6%	30%	6%	28%	2%	15%	0%	3%
spacy	9%	18%	9%	17%	5%	24%	4%	14%	10%	15%
deeppavlov	2%	23%	1%	10%	0%	17%	0%	3%	1%	10%
stanza	4%	21%	2%	35%	2%	32%	0%	5%	0%	4%

Таблица 10. Количество предложений с изменениями LAS (B — ВТ, Э — ЭТ)

4.3 Изменение категории токенов при замене ВТ на ЭТ

При использовании ЭТ вместо ВТ может произойти изменение категории токенов как для токенов 1 категории, так и для токенов 2–4 категорий. Подробная статистика по изменению токенами категории при замене ВТ на ЭТ приведена в Приложении В. Соотношение токенов, изменивших категорию, зависит от анализатора и датасета. Например, для СА DeepPavlov на датасетах Poetry, GSD и PUD при замене ВТ на ЭТ ни для одного токена не произошло понижение категории. А для СА Natasha на датасете Poetry количество токенов, для которых категория понизилась с 4 на 2, превышает количество токенов, для которых категория повысилась с 2 до 4.

Для датасета Taiga и большинства анализаторов, датасета GSD и анализаторов Natasha и UDPipe большинство токенов 1 категории переходят во 2 категорию (т.е. не влияют на изменения значений метрик UAS и LAS). Для датасетов Poetry, PUD, SynTagRus и большинства анализаторов, для датасета GSD и анализаторов spacy и DeepPavlov большинство токенов 1 категории переходит в 4 категорию (т.е. повышают и UAS, и LAS).

Изменение категории токенов связано с принципами применения нейронной сети для исследуемых анализаторов. Нейронная сеть применяется к векторным представлениям (embedding) токенов предложения. При изменении набора токенов изменяются и векторные представления токенов. Для новых векторных представлений может быть предсказано другое преобразование предложения (для анализаторов на основе переходов) или другое соотношение весов ребер (для анализаторов на основе графов). Следовательно, для скорректированного набора токенов результат синтаксического анализа может стать принципиально другим (в том числе и для токенов, которые не подверглись преобразованию). Более подробный анализ взаимосвязи изменений векторного представления токенов и результатов работы синтаксических анализаторов планируется в дальнейшем.

5 Заключение

В данной работе проведено сравнение двух подходов оценки синтаксических анализаторов: с использованием встроенного токенизатора и с использованием эталонного токенизатора. Для сравнения этих подходов в данной статье разработана система категорий токенов, учитывающая качество и токенизации, и синтаксического анализа.

Исследование проведено для 5 русскоязычных синтаксических анализаторов (UDPipe, Stanza, DeepPavlov, Natasha, spacy) на 5 русскоязычных датасетах деревьев зависимостей (SynTagRus, PUD, GSD, Taiga, Poetry). На всех датасетах и анализаторах для значительной доли тестовых предложений выявлено несовпадение токенизации, полученной с помощью встроенного токенизатора, с эталонной токенизацией.

Установлено, что для датасетов Poetry, GSD и SynTagRus в зависимости от выбора токенизатора изменяется то, какой анализатор достигает наиболее высокого значения метрик. Однако на всех датасетах все синтаксические анализаторы показали более высокие средние значения метрик UAS и LAS при использовании эталонного токенизатора. В зависимости от датасета и анализатора увеличение среднего значения составляет от 0.004 до 0.082 по метрике UAS и от 0.003 до 0.072 по метрике LAS. Наибольшее увеличение значения метрик происходит для датасета GSD и анализатора DeepPavlov.

Дополнительно выявлено, что при замене встроенного токенизатора на эталонный токенизатор качество синтаксического анализа может измениться и для токенов, которых не затронуло изменение токенизации. В частности, может происходить ухудшение качества синтаксического анализа.

В дальнейшем планируется исследование взаимосвязи между изменениями набора токенов, векторных представлений токенов, результатов синтаксического анализа и числовых значений метрик UAS и LAS.

При внедрении нового токенизатора качество работы синтаксического анализатора может как улучшиться, так и ухудшиться. Для проверки работоспособности синтаксического анализатора с новым токенизатором может быть применена предложенная методология сравнения категорий токенов.

Список литературы

1. Morozov, D., Lagutina, K., Drozhashchikh, G. et al.: Exploring the Feature Space for Cross-Domain Assessing the Complexity of Russian-Language Texts. In: 2024 Ivannikov Ispras Open Conference (ISPRAS), pp. 1–8. IEEE, Moscow, Russian Federation (2024) <https://doi.org/10.1109/ISPRAS64596.2024.10899137>
2. Li, L., Chen, Z., Liao, S. et al.: Event Extraction in Complex Sentences Based on Dependency Parsing and Longformer. In: Proceedings of 2024 International Conference on Machine Learning and Intelligent Computing, pp. 1–7. PMLR, Wuhan, China (2024)
3. Vasiliev, S., Korobkin, D., Fomenkov, S.: Extracting the Component Composition Data of Inventions from Russian Patents using Dependency Tree Analysis. In: 2023 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), pp. 1030–1034. IEEE, Sochi, Russian Federation (2023)
4. Alonso, M., Gómez-Rodríguez, C., Vilares, J.: On the Use of Parsing for Named Entity Recognition. *Applied Sciences* **11**(3)(2021) <https://doi.org/10.3390/app11031090>
5. Nikolaev, I.: Knowledge and skills extraction from the job requirements texts. *Ontology of Designing* **13**(2), 282–293 (2023) <https://doi.org/10.18287/2223-9537-2023-13-2-282-293>
6. Taufiq, U., Pulungan, R., Suyanto, Y.: Named entity recognition and dependency parsing for better concept extraction in summary obfuscation detection. *Expert Systems with Applications* **217**, 119579 (2023) <https://doi.org/10.1016/j.eswa.2023.119579>
7. Liu, T., Sun, Y., Wu, J. et al.: Unsupervised Paraphrasing under Syntax Knowledge. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 13273–13281 (2023) <https://doi.org/10.1609/aaai.v37i11.26558>
8. Straka, M., Strakova, J.: Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies 2017, pp. 88–99. Vancouver, Canada (2017). <https://doi.org/https://doi.org/10.18653/v1/K17-3001>
9. Qi, P., Zhang, Y. et.al: Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations 2020, pp. 101–108. Online (2020). <https://doi.org/10.18653/v1/2020.acl-demos.14>
10. Nivre, J., Hall, J., Kubler, S. et.al: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll), pp. 915–932. Association for Computational Linguistics, Prague, Czech Republic (2007)
11. Surdeanu, M., Johansson, R., Meyers, A. et al.: The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In: CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning, pp. 159–177. Coling 2008 Organizing Committee, Manchester, England (2008)
12. Zeman, D., Hajic, J., Popel, M. et al.: CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 1–19. Vancouver, Canada (2017)
13. Zeman, D., Hajic, J., Popel, M. et al.: CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 1–21. Brussels, Belgium (2018)

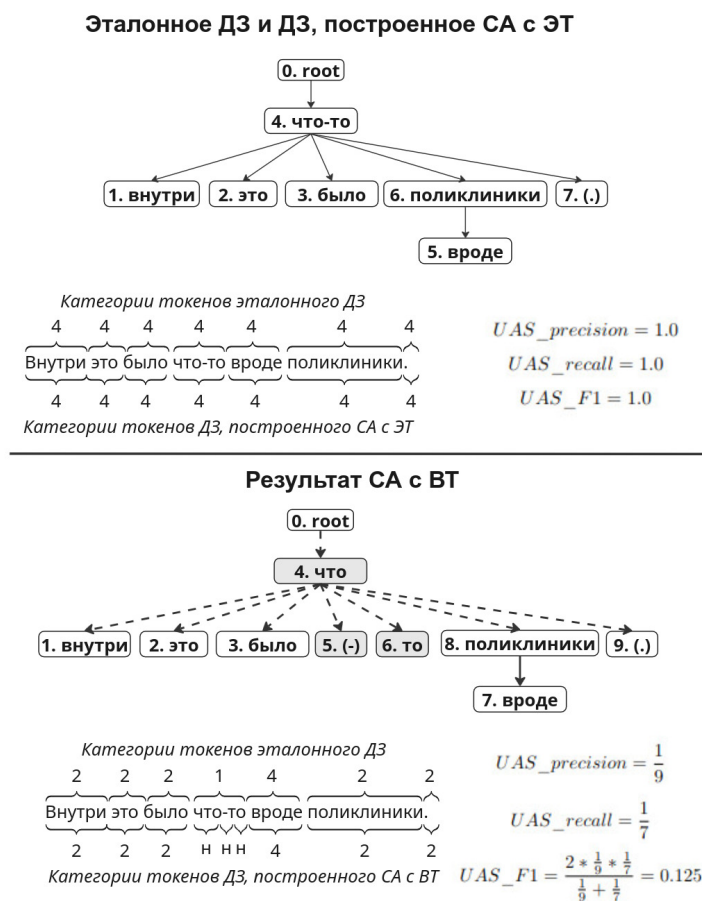
14. Ляшевская, О. Н., Шаврина, Т. О., Трофимов, И. В. и др.: GramEval 2020: дорожка по автоматическому морфологическому и синтаксическому анализу русских текстов. In: Annual International Conference Dialogue, pp. 553–569. Publisher, Location (2020)
15. Zhang, M.: A survey of syntactic-semantic parsing based on constituent and dependency structures. Science China Technological Sciences **63**(10), 1898–1920 (2020)
16. Chen, D., Manning C.: A fast and accurate dependency parser using neural networks. In: In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 740–750. Association for Computational Linguistics, Doha, Qatar (2014)
17. Andor, D., Alberti, C., Weiss, D. et.al: Globally normalized transition-based neural networks. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 2442–2452. Association for Computational Linguistics, Berlin, Germany (2016)
18. Dozat, T., Manning, C.D.: Deep Biaffine Attention for Neural Dependency Parsing. In: International Conference on Learning Representations (ICLR). Publisher, Location (2017)
19. Droganova, K., Lyashevskaya, O. et al.: Data Conversion and Consistency of Monolingual Corpora: Russian UD Treebanks. In: Proceedings of the 17th international workshop on treebanks and linguistic theories (tlt 2018), pp. 53–66. Linköping University Electronic Press, Linköping, Sweden (2018)

Приложение А. Влияние токенизации на качество синтаксического анализатора

На Рис. 2 показаны эталонное и построенное деревья зависимостей (ДЗ) для предложения «Внутри это было что-то вроде поликлиники.». На рисунке пунктиром выделены ребра, которые считаются «неверными» и не учитываются в качестве числителя метрик UAS и LAS.

Для этого примера при использовании ВТ значение метрики UAS равняется 0.125 (см. формулы ниже), при использовании ЭТ — 1.0. При использовании ВТ эталонный токен «что-то» отсутствует в построенном дереве зависимостей: для всех его дочерних токенов главный токен считается определенным неверно.

На Рис. 3 приведен пример предложения «Бритва была где-то на дне.», для которого значение метрики UAS при использовании ВТ выше, чем при использовании ЭТ.

**Рис. 2.** Пример увеличения UAS при замене ВТ на ЭТ

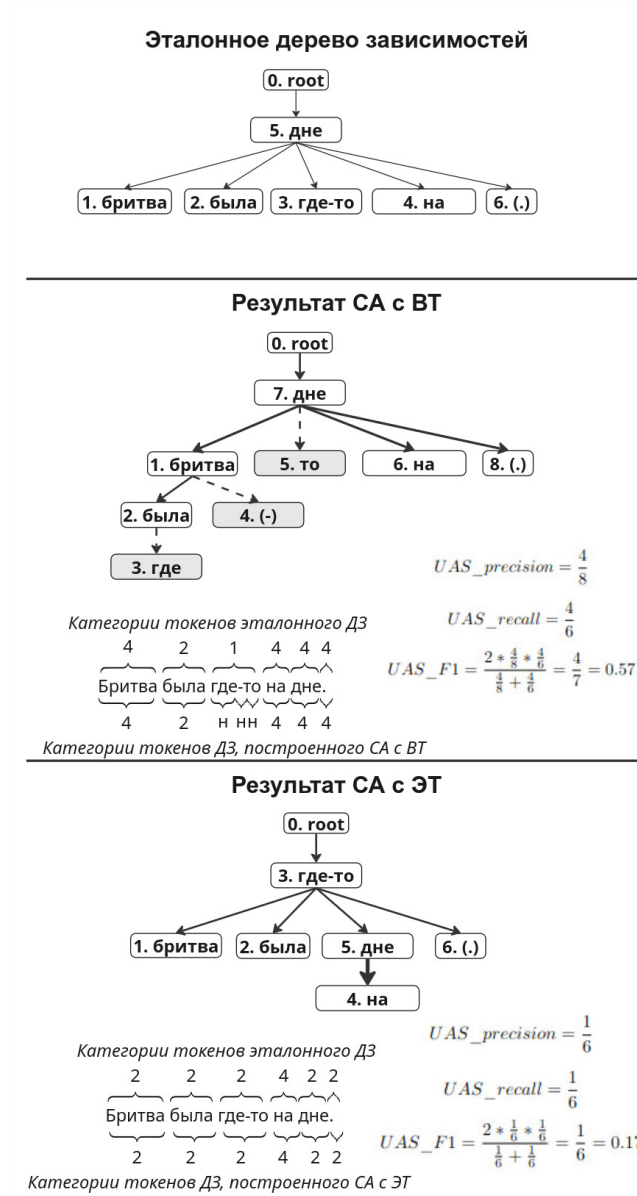


Рис. 3. Пример уменьшения UAS при замене ВТ на ЭТ

Приложение В. Изменение категорий токенов

На Рис. 4 приведена статистика по изменению категорий токенов при замене ВТ на ЭТ.

taiga_natasha				taiga_udpipe			taiga_spacy			taiga_deeppavlov			taiga_stanza			
встроен.	1	195	80	76	194	65	146	158	57	83	178	113	160	111	52	133
	2	2863	27	110	2262	37	131	2015	37	181	1602	22	111	1687	27	119
	3	10	564	8	27	710	9	26	571	13	5	718	9	17	659	5
	4	57	4	6280	94	10	6589	139	12	6982	14	1	7341	50	5	7409
		2	3	4	2	3	4	2	3	4	2	3	4	2	3	4
poetry_natasha				poetry_udpipe			poetry_spacy			poetry_deeppavlov			poetry_stanza			
встроен.	1	45	10	35	192	7	215	47	12	75	57	14	94	84	3	236
	2	3934	14	58	2459	37	183	2403	24	152	1595	8	57	1489	15	245
	3	13	595	3	41	693	33	32	553	24	0	700	0	10	624	15
	4	67	3	5261	161	26	5991	112	16	6588	0	0	7513	92	13	7212
		2	3	4	2	3	4	2	3	4	2	3	4	2	3	4
gsd_natasha				gsd_udpipe			gsd_spacy			gsd_deeppavlov			gsd_stanza			
встроен.	1	200	26	183	448	40	219	203	72	226	379	140	1044	308	102	307
	2	2166	31	137	1646	57	212	1332	20	190	845	13	82	1055	21	153
	3	20	486	7	31	913	10	9	488	5	0	867	0	13	674	9
	4	56	8	8065	147	19	7643	63	13	8764	0	0	8015	52	12	8679
		2	3	4	2	3	4	2	3	4	2	3	4	2	3	4
pud_natasha				pud_udpipe			pud_spacy			pud_deeppavlov			pud_stanza			
встроен.	1	57	32	41	241	63	339	67	17	95	17	24	55	7	15	50
	2	2389	13	60	2104	24	224	1370	16	125	1230	3	28	1364	4	57
	3	1	870	2	8	1384	10	9	834	12	0	1411	0	1	1196	4
	4	23	1	15866	64	12	14882	51	12	16747	0	0	16587	4	0	16653
		2	3	4	2	3	4	2	3	4	2	3	4	2	3	4
syntagrus_natasha				syntagrus_udpipe			syntagrus_spacy			syntagrus_deeppavlov			syntagrus_stanza			
встроен.	1	1019	322	915	141	39	303	790	246	932	270	427	1573	67	23	347
	2	29248	208	718	20038	91	453	17486	207	1510	9879	79	943	9727	24	533
	3	34	6782	20	9	7150	8	143	7148	184	16	5007	80	1	4153	1
	4	378	25	118049	90	8	129388	1452	221	127399	156	45	139243	33	9	142800
		2	3	4	2	3	4	2	3	4	2	3	4	2	3	4
		эталон.			эталон.			эталон.			эталон.			эталон.		

Рис. 4. Изменение категорий токенов

Рецензирование

Рецензия 1:

1. **Исправлено** — «"В рамках данного исследования для каждого из этих анализаторов реализован токенизатор, возвращающий эталонный набор токенов из датасета. возможно имеет смысл в README.md на GitHub указать в каком файле находится эта реализация»;
2. **Исправлено** — в файле README.md «Указать, где в коде выполняется запуск с использованием встроенного токенизатора, а где с использованием эталонного»;
3. **Исправлено** — в файле README.md добавить «Описание основных классов»;
4. **Исправлено** — в файле README.md «Более подробно описать как в коде вычисляются значения метрик».

Рецензия 2:

1. **Добавлено теоретическое обоснование полученных результатов, статистический анализ планируется провести в рамках отдельного исследования** — «рекомендуется добавить более подробное обсуждение результатов, представленных в таблицах 3-10, и подробнее обсудить, с чем связаны расхождения между ВТ и ЭТ и вариативность результатов для разных датасетов»;
2. **Исправлено** — «добавить выводы, касающиеся практического использования токенизаторов в NLP-пайплайнах».

Рецензия 3:

1. **Добавлено теоретическое обоснование полученных результатов, статистический анализ планируется провести в рамках отдельного исследования** — «Цитата: "Например, на датасете Poetry при использовании для анализатора Natasha эталонного токенизатора вместо встроенного токенизатора количество токенов, на которых произошло ухудшение качества синтаксического анализа, превысило количество токенов, для которых произошло улучшение.." — не раскрыты причины ухудшения. Следует провести анализ и объяснить.»
2. **Исправлено** для таблиц 5–10 — «В таблицах результатов следует выделить шрифтом лучшие строки с лучшими значениями. В текущем виде таблицы трудно читаемые».
3. **Исправлено** — «В заключение стоит указать минимальные и максимальные величины улучшения UAS/LAS».
4. **Исправлено** — «Цитата: "Подробная статистика по изменению токенами категории при замене ЭТ на ВТ приведена в Приложении А." — вероятно, перепутан порядок "ВТ на ЭТ"» (Приложение А перемещено в Приложение В).