Компьютерная лингвистика и интеллектуальные технологии:
по материалам международной конференции «Диалог 2017»

Москва, 31 мая — 3 июня 2017

АВТОМАТИЧЕСКОЕ ОПРЕДЕЛЕНИЕ ГРАНИЦ СЛОВА В РУССКОМ ТЕКСТЕ С ПОМОЩЬЮ КОМПЛЕКСА ЛИНГВИСТИЧЕСКИХ ПРАВИЛ

Ермакович М. В. (Maria.Yermakovich@ihsmarkit.com)

ООО «АйЭйчЭс Глобал», г. Минск, Республика Беларусь

AUTOMATIC WORD BOUNDARY DISAMBIGUATION ON RUSSIAN TEXT USING LINGUISTIC RULES

Yermakovich M. V. (Maria.Yermakovich@ihsmarkit.com)

IHS Global LTD, Minsk, Belarus

This paper covers the example of a rule-based approach to word boundary disambiguation in Russian text. WBD module is a first step in complex text tokenization and analysis process implemented in compound linguistic processor. It comprises a collection of regular expressions based rules, is context insensitive, reentrant, considers complex cases and includes abbreviation recognition superstructure. The module measures against manually tokenized SynTagRus corpus with F1 score of 99.93%. However, context insensitive nature of the module sets limitations to its effectiveness in the cases of paired non-word symbols adjacent to different tokens, homographs and numbers in brackets vs. numbered bullets, all of which can only be resolved depending on left and right context. Possible solutions include making the module itself able to see context or considering reuniting some of erroneously splitted words on later stages. The module also has potential to improve performance on domain-specific texts, which requires more domain-specific rules.

Key words: tokenization, word boundary, word boundary disambiguation, ruleset

1. Введение

Разбиение текста на слова, или токенизация, — базовый этап в автоматической обработке текста. Задача определения границ слова в рамках автоматической обработки русского печатного текста не сводится к отграничению фрагмента от пробела до пробела и отделению небуквенных символов от последовательности букв.

Многие системы токенизации либо рассчитаны на специфические задачи, такие как токенизация текста в языках без пробелов, например, в китайском [14]; сегментация устной речи [16]; определение хэштегов в текстах социальных сетей [3] и др., либо входят в состав комплексных систем, например OpenCorpora [6, 7, 8, 15] и NLTK [4, 5]. Вероятностные модели с машинным обучением [6, 7, 8, 15] требуют предварительного создания тренировочного корпуса, изменяются при добавлении новых примеров в корпус и могут иметь ограничения по скорости работы. Такие системы, как NLTK, при заявленной универсальности, натренированы в основном на англоязычных текстах. Некоторые системы токенизации включают словарную проверку (Яндекс.Словари для OpenCorpora [7]). Поскольку современные передовые системы синтаксического анализа языка (такие как SyntaxNet от Google [2]) нужно тренировать на предварительно токенизированном тексте, сложно переоценить актуальность модуля токенизации.

Лингвистический процессор (ЛП) [1] обрабатывает большие объемы текстов различных жанров, в том числе патенты, научную литературу, ленты новостей, сообщения из социальных сетей, отзывы потребителей и другие. Особенностью текстов для обработки в ЛП является большое количество «мусора» — непригодных для обработки фрагментов: это и ошибочный или неаккуратный пользовательский ввод, и ошибки распознавания других форматов (например, PDF) и кодировок. От максимально точной токенизации текста зависит последующее корректное определение частеречной принадлежности (тегирование) каждого из токенов, в том числе и не входящих в словарь [12]. Для адекватной обработки текста необходима особая процедура токенизации, в равной мере учитывающая специфику русского языка, особенности обрабатываемых источников текста и согласование результатов работы модуля с последующими шагами обработки текста в ЛП.

Токенизация русского текста в лингвистическом процессоре [1] проходит в несколько этапов, первым из которых является модуль Word Boundary Disambiguation (WBD), представляющий собой комплекс лингвистических правил. Создание WBD для русского языка — часть локализации ЛП [1], поэтому основана на решении, которое успешно применяется для токенизации языков, содержащих пробелы.

Разработка собственной системы, основанной на правилах, велась по ряду следующих причин:

а) модуль WBD для русского языка разрабатывался в составе многоязычного ЛП, в котором рассматриваемый подход, основанный на правилах, успешно применяется для других языков;

б) при разбиении текста, в особенности перенасыщенного нетипичными токенами (смесью букв и пунктуации, например, при анализе

химических текстов), вырабатывается собственная идеология — динамичные соглашения в зависимости от состояния готовности, задач и состава последующих модулей;

в) модуль вынесен в отдельный файл, и, благодаря доступной и прозрачной нотации правил, его поддержка и корректировка возможны без дополнительных знаний алгоритмов;

г) разрабатывая программное обеспечение для предприятий, важно иметь возможность быстро и с минимальным влиянием на другие токены вносить правки в разбиение конкретных случаев в соответствии с запросами бизнеса;

д) правила WBD основаны на регулярных выражениях, представляемых автоматной грамматикой, применяемой с линейной трудоемкостью, в силу чего обеспечивается высокая скорость обработки текстов данным модулем.

2. Нотация правил модуля WBD

Каждое правило модуля WBD, реализованного и применяемого в [1], состоит минимум из двух полей: поля условия, описанного регулярным выражением [11], и поля действия — остановить токенизацию (разбиение) токена (команда _put), или продолжить обработку полученного при разбиении подтокена другими правилами (команда _rec). Более сложные правила, описывающие конкретные контексты, могут состоять из нескольких пар условий и действий.

WBD — контекстно-независимый модуль. Система его правил применяется исключительно к индивидуальным фрагментам входной строки от пробела до пробела (токенам). Правила WBD повторно входимы, т.е. если токен соответствует заданному в правиле регулярному выражению, он будет сохранен в неизменном виде, либо разбит согласно полю действия данного правила; при этом во втором случае система правил продолжит работу с полученными токенами до тех пор, пока разделенные фрагменты токена не будут соответствовать ни одному правилу.

Некоторые из описанных регулярными выражениями повторяющихся токенов для удобства заданы определениями, перечисленными перед правилами с помощью макропроцессора m4 [13], например: rus_word — русское слово в общем случае; eos — варианты пунктуации конца предложения; bullet — маркеры списка, в том числе числовые и буквенные; quote — варианты знака кавычки, и другие. Определение выглядит следующим образом:

```
m4_define( ``quote",`` [>»`'"«„“”‘’»‹›]")
```

Приоритет срабатывания правил соответствует их порядку в исходном файле системы правил модуля WBD, поэтому более общие правила следует предварять описаниями более частных случаев. Напротив, если написать более общее правило раньше, более узкое частное правило может никогда не сработать.

Пример токенизации можно представить следующим образом: для токена **будущем».5** срабатывает одно из частных правил:

(rus_word)(quote)(eos) _rec (bullet)|[0-9]+ _put

Оно разбивает его на две части. Для второй части, состоящей из одной цифры, токенизация на этом останавливается командой _put, а на первой при повторном прохождении через модуль срабатывает более общее правило:

(rus_word) _put (punct_and_more)+ _rec (eos) _put

Оно, в свою очередь, разбивает фрагмент на три токена: русское слово, закрывающую кавычку и точку. И хотя кавычка описана той частью правила, которая позволяет дальнейшее разбиение, для одного символа токенизация не требуется, и поэтому токенизация всего фрагмента останавливается. В результате получаем четыре токена:

будущем » . 5

Из примера видно, что фрагмент текста был затронут двумя правилами, сначала более частным, затем более общим. Для сокращения времени обработки текста правилами также логично помещать правила для токенов, которые нельзя разбивать, в начало модуля. Если не описывать правилами не требующие разбиения токены, то такой фрагмент может быть случайно разбит другим правилом, либо он будет проверен всеми правилами, и ни одно не сработает на нем, что увеличит время обработки текста.

3. Разработка правил

В рамках решения прикладной задачи токенизации корпуса русских текстов создание комплекса правил WBD проходило эмпирически и итеративно. Первая версия правил, предназначенная для базовой токенизации (отделение знаков препинания от полноценных слов), прошла проверку на текстах различных жанров, после чего для выявленных проблемных случаев были добавлены новые правила и скорректированы существующие; дальнейшая разработка проходила итеративно схожим образом.

Для обеспечения стабильности качества и планомерного улучшения при разработке модуля был создан пополняемый комплекс тесткейсов — случаев, которые либо уже обрабатываются правильно, либо требуют дальнейшего исследования для оптимального разрешения. Примеры таких случаев отражены в таблице:

	Последовательность	**Необходимая токенизация**
(1)	*30,71%.*	*30,71 % .*
(2)	**Самой*	** Самой*
(3)	*стоимость.....1*	*стоимость 1*
(4)	*канд.техн.наук, зав.отделом*	*канд. техн. наук , зав. отделом*
(5)	*Мурманс. обл.,Менатеп,Роспром-соглашение*	*Мурманс. обл. , Менатеп , Роспром - соглашение*
(6)	*колее",-сказал*	*колее " , - сказал*
(7)	*большевизмом),много*	*большевизмом) , много*

	Последовательность	Необходимая токенизация
(8)	*Свинец–5мг/кг*	*Свинец – 5 мг/кг*
(9)	*будущем».5 Политолог*	*будущем » . 5 Политолог*
(10)	*мкг/мл*	*мкг/мл*
(11)	*1. 1) а) 1а) 1а. 1.2 (а) (1) (I) (xvii) 12E*	*1. 1) а) 1а) 1а. 1.2 (а) (1) (I) (xvii) 12E*
(12)	*[2M+Na]+*	*[2M+Na]+*
(13)	*[(Ni+Mn)/(1+Mo)]·P<0,03*	*[(Ni+Mn)/(1+Mo)]·P<0,03*
(14)	*электро-и теплоэнергии*	*электро- и теплоэнергии*
(15)	*к.б.н. и т.д.*	*к.б.н. и т.д.*
(16)	*(см. интервью с И. Д. Новиковым)*	*(см. интервью с И. Д. Новиковым)*
(17)	*±23,5*	*±23,5*
(18)	*-20*	*-20*
(19)	*>20*	*> 20*
(20)	*~20*	*~ 20*

Токенизация текста поднимает вопросы условности границ слов, поэтому в общем случае вопрос о разбиении решался образом, наиболее удобным для дальнейшего тегирования текста в рамках текущих возможностей и извлечения из него знаний. Например, от чисел не отделяются их знаковые атрибуты (примеры 17, 18), при этом отделяются префиксные операторы (примеры 19, 20).

4. Связь модуля WBD с последующими модулями

Следует упомянуть, что на этапе WBD происходит только достаточно базовая токенизация сложных слов, поскольку дальнейшая обработка текста в лингвистическом процессоре производится рядом модулей более высокого уровня.

В частности, решается задача определения границ предложения. Как в случае определения границ слова не всегда пробел либо его отсутствие — надежный показатель границ слова, так и на этапе членения текста на предложения точка перед большой буквой — не всегда показатель начала нового предложения. Самый очевидный пример — инициалы в списке литературы:

(21) *Юдин А. И., Матюхин А. Д. Раннесрубные курганные могильники Золотая Гора и Кочетное. — Саратов: Научная книга, 2006. — 116 с.*

Поскольку модуль членения на предложения учитывает более широкий контекст, случаи, когда требуется отбить точку на границе предложения, проще решить с его помощью.

От работы WBD зависит корректная работа модулей, отвечающих за тегирование (определение частеречной принадлежности) как слов из словаря (наша задача — «освободить» их от небуквенных символов), так и неизвестных слов. Решение последней задачи обеспечивает модуль, распознающий тег слова по формальным признакам — суффиксам и префиксам. При этом особое внимание уделяется проблеме назначения тегов составным словам (написанным через дефис или косую черту), поэтому особенно важно сохранять их контактное расположение до этого этапа:

(22) *год-два*), → *год-два*) ,

По вышеуказанным причинам слова, написанные через косую черту и дефис, не разделяются на этапе WBD, тем не менее, при этом требуется отделять от них несловные символы.

Токенизация в WBD — первый и достаточно универсальный шаг в цепочке преобразований слова. Если для поиска, перевода либо других задач требуется особая токенизация, ее осуществляет отдельный модуль. Например, если на запрос «палатка» требуется находить также тексты, содержащие слово «плащ-палатка», оно будет токенизировано по дефису.

Специфической чертой русского текста является большое количество сокращений, например, слов со строчной буквы с точкой на конце в середине предложения. Многочисленные замеры на сбалансированных корпусах английского и русского языков показывают, что в русском тексте в среднем в шесть раз больше сокращений. Наиболее частотные сокращения с точкой помещены в словарь и учтены в правилах WBD. Для того, чтобы успешно токенизировать другие сокращения — то есть не оделять от них точки — был разработан модуль-надстройка над WBD, сохраняющий точки для последовательности двух и более букв, оканчивающейся на согласный. Это происходит при условии, что в словаре есть большее по длине слово, в которое входит последовательность, но нет слова, совпадающего с ней:

(23) *англ.* → *англ.* (в словаре есть «английский»)

(24) *стол.* → *стол* . (в словаре есть «стол»)

Такой подход, разумеется, решает только общие случаи и нуждается в уточнении:

(25) *атом. %* → *атом* . *%*

«Атом.» сокращением не считается, поскольку словарь содержит полностью совпадающую с ним полную форма «атом».

5. Оценка качества токенизации на корпусе Синтагрус

Для оценки качества определения границ слова модулем WBD мы сравнили его работу на нетегированных предложениях, составляющих Синтагрус — синтаксически размеченный корпус русского языка [8], с его эталонной разбивкой, выполненной вручную экспертами. Объем корпуса составляет 66 273 предложений, 1 173 548 слов.

Большинство конфликтов токенизации являлись идеологическими несовпадениями, не отражающимися на качестве текста. Например, в корпусе Синтагрус [8]:

1) от сокращения в конце, а часто и в середине предложения отделяют точку (млн .);
2) рассматривают буллет как отдельные символы (1)), в нашем корпусе буллеты не разделены;

3) разделяют написанные слитно сочетания слов с «не», «пол» и некоторыми другими префиксами: некогда -> не когда, некому -> не кому, полгода -> пол года.

Следует отметить, что несовпадения возникают и в местах, где автоматические правила разбивают текст лучше, чем это делает аннотатор-человек, в частности в Синтагрусе целыми остались подобные токены:

(26) *99-м,*

После того как вышеуказанные конфликты были разрешены, на данном корпусе точность (количество совпавших слов относительно слов, выделенных модулем WBD) составила 99.9329%, полнота (количество совпавших слов относительно количества слов в Синтагрусе) 99.9226%.

6. Оценка качества токенизации на собственном корпусе

В рамках разработки лингвистического процессора [1] экспертами-лингвистами был создан размеченный корпус текстов различных жанров. Разметка включает токенизацию, деление на предложения и тегирование. Объем корпуса составляет 38 997 слов и 2058 предложений. На этом корпусе точность токенизации (количество совпавших слов относительно слов, выделенных модулем WBD) составляет 99,2916 %, полнота (количество совпавших слов относительно количества слов в собственном корпусе) 99,2051 %. Основные расхождения — это действительные ошибки токенизации, самые типичные из которых:

Точка не отделена от числа на конце предложения, поскольку такой токен ошибочно считается буллетом;

Ошибочно отделена точка от сокращений — омографов полных слов из словаря (***пер .*** *с англ.*)

Точка конца предложения не отделена от полных слов, совпадающих по форме с сокращениями с точкой из словаря (*Я был разочарован* ***им.***)

Также частотным расхождением являются случаи, где необходимо разделять стоящие контактно слова, «потерявшие» пробел: *При программировании* ***микросхемыпрограмма-загрузчик*** *копируется в ОЗУ, таким образом, весь объем FLASH-памяти может быть* ***занятпользовательской*** *программой.*

При этом среди типичных проблем токенизации, не решаемых в рамках текущей функциональности, необходимо отметить следующие:

1) На этапе токенизации невозможно проверить парность скобок, кавычек и других парных символов, если они не находятся в рамках одного токена. Это создает вышеописанные трудности с неверным тегированием буллетов.
2) Неправильное распознавание точек конца сокращения и конца предложения.
3) За неимением контекста невозможно различить омографичные знаки, такие как минус и дефис: *Сравнение энергий стабилизации комплексов молекулярного йода с аргинином -18,17 кДж/моль и с аденозином* ***-10,93*** *ккал/моль и гуанином* ***-11, 50*** *кДж/моль.*

В подобных случаях видится большое поле для улучшений в рамках новой и расширенной функциональности WBD. Такие проблемы преодолимы за счет изменения логики модуля, если добавить в правила контекст, либо, вслед за создателями OpenCorpora [7, с. 4] по умолчанию разбивать все противоречивые случаи на уровне WBD, и затем, в более широком контексте, заново группировать и склеивать без пробелов те из них, которые требуют такого подхода.

7. Сравнение с NLTK модулем TokTok

Сравнение модулей токенизации текста — специфичная задача, осложненная идеологическими различиями в понимании границ слова в каждой из систем. Поскольку токенизатор OpenCorpora [15] основан на машинном обучении, вторым решением, относительно которого мы замеряем качество, стал модуль TokTok [10] — общее решение для токенизации таких языков, как английский, русский, персидский, вьетнамский и другие, в составе большой статистической модели [9]. Как и WBD, TokTok использует правила и определения.

Результаты замеров работы модулей на одинаковых текстах следующие:

корпус		**SynTagRus**	**Собственный корпус**
Объем, предложений		66 273	2 058
Объем, слов		1 173 548	38 997
WBD	**точность**	99,9329 %	99,2916 %
	полнота	99,9226 %	99,2051 %
	F1	99,9277 %	99,2483 %
TokTok	**точность**	99,4926 %	97,6822 %
	полнота	99,7525 %	98,3435 %
	F1	99,6223 %	98,0117 %

В модуле TokTok, в отличие от модуля WBD:

1. отделены точки и скобки от буллетов: **Таблица 2 .**; **(г)**
2. разделяют градусы и обозначения шкалы измерения: **° С**. В WBD принято оставлять их написанными слитно, как одну единицу измерения;
3. не разделяют написанные слитно сокращения и цифры: **фиг.3**, **табл.1**;
4. разделяют полнозначные слова, содержащие цифры и знаки пунктуации: **10 % -ного**; **2 ' -фталимидметилбифенил-2-карбоновой; 60 - х; 3-ацетилсульфанил-2- 6-трет-бутоксикарбониламино-пиридин-3-илметил -масляной;**
5. отделяют точки от инициалов: **П .**

Соответственно, если бы модуль лексико-грамматического анализа в ЛП использовал токенизацию, выполненную с помощью TokTok, слова из пунктов 3 и 4 получили бы ошибочные частеречные теги.

Есть случаи, где ошибаются как TokTok, так и WBD: **http://internal.**psychology**.illinois.edu/ ~rcfraley/attachment . htm**

WBD ошибается в перечисленных выше случаях: путая буллеты с частью предложения, сокращения с полными словами и тому подобное. Можно прийти к выводу, что TokTok имеет хорошее базовое качество токенизации, и сравнение с ним может быть полезно для написания будущих правил WBD. При этом он не подходит для обработки специфических случаев, включенных в корпус тесткейсов [1].

Следует отметить, что даже минимальная в процентном соотношении разница в результатах работы любого модуля вызывает большое неудобство с точки зрения пользователя. Именно поэтому WBD ориентирован в том числе на частные случаи и узкие контексты, и дальнейшая работа ведется с целью сокращения числа случаев ошибочной токенизации.

8. Дальнейшие улучшения

Среди возможных направлений работы над правилами в первую очередь предполагаются усилия, направленные на улучшение токенизации текстов отдельных предметных областей, в частности:

1. Токенизация формул ($xy = p\,[xy/p] + p - 1$) для их последующего анализа.
2. Особые правила для социальных медиа. В рамках текущей функциональности уже сохраняются простые смайлы, а в дальнейшем возможна углубленная работа над данным полем.
3. Расшифровка сокращений. Модуль «приклеивания» точек к сокращениям, фактически, их распознает, поэтому, опираясь на контекст и, подобрав базу сокращений, его можно использовать для задачи восстановления сокращений.
4. «Расклеивание» по ошибке написанных слитно слов с помощью словаря.

Выводы

Решение задачи токенизации русского текста с помощью лингвистических правил возможно с точностью более 99 %. Лингвистические правила позволяют точно описывать конкретные случаи и легко изменять модуль. При этом контекстно-независимая природа модуля определяет его ограниченность в случаях необходимости опираться на более чем один токен. Дальнейшие пути развития модуля включают как пересмотр идеологии правил, так и написание новых, специфичных для конкретных доменов.

References

1. US-8,666,730 "Question-answering system and method based on semantic labeling of text documents and user questions". James Todhunter, Igor Sovpel, Dzianis Pastanohau. (2009/2010)
2. *Alberti C., Andor D., Bogatyy I., Collins M., Gillick D., Kong L., Koo T., Ma J., Omernick M., Petrov S., Thanapirom C.* SyntaxNet Models for the CoNLL 2017 Shared Task. arXiv preprint arXiv:1703.04929. Mar 15 2017.

3. *Berardi, G., Esuli, A., Marcheggiani, D., & Sebastiani, F.* (2011). ISTI@ TREC Microblog Track 2011: Exploring the Use of Hashtag Segmentation and Text Quality Ranking. In TREC.
4. *Bird, S.* (2006). NLTK: the natural language toolkit. In Proceedings of the COLING/ACL on Interactive presentation sessions. Association for Computational Linguistics, July 2016, p. 69–72.
5. *Bird, Steven, Ewan Klein, and Edward Loper* (2009), Natural Language Processing with Python, O'Reilly Media, CA.
6. *Bocharov V., Bichineva S., Granovsky D., Ostapuk N., Stepanova M.* (2011) Quality assurance tools in the OpenCorpora project [Instrumenty kontrolja kachestva dannyh v proekte Otkrytyj Korpus]. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog"], Bekasovo, 25–29 May 2011, Vol. 10 (17)
7. *Bocharov V. V., Alekseeva S. V., Granovsky D. V., Ostapuk N. A., Stepanova M. E., Surikov A. V.* (2012) Text segmentation in OpenCorpora project [Segmentatsiya teksta v proekte «Otkryityiy korpus»] Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog"], Bekasovo, 30 May — 3 June 2012, Vol. 11 (18)
8. *Bocharov V. V., Granovsky D. V., Surikov A. V.* (2012) Probabilistic tokenization model in the OpenCorpora project [Verojatnostnaja model' tokenizacii v proekte Otkrytyj korpus] New information technologies in automated systems: materials of the fifteenth scientific and practical seminar [Novye informacionnye tehnologii v avtomatizirovannyh sistemah: materialy pjatnadcatogo nauchno-prakticheskogo seminara]. Moscow State Institute of Electronics and Mathematics
9. *Dehdari J.* (2014), A Neurophysiologically-Inspired Statistical Language Model (Doctoral dissertation). Columbus, OH, USA: The Ohio State University.
10. *Dehdari J.* TokTok Tokenixer Module for NLTK, documentation available at: http://www.nltk.org/_modules/nltk/tokenize/toktok.html#ToktokTokenizer
11. *Friedl J. E. F.* (2006), Mastering Regular Expressions, 3rd Edition, O'Reilly Media, CA
12. *Jurafsky, D.* (2008), Speech and Language Processing, Prentice Hall, New Jersey
13. *Kernighan, Brian W., and Dennis M. Ritchie.* The M4 macro processor. Murray Hill, NJ: Bell Laboratories, 1977.
14. *Palmer D.* (1997), A trainable rule-based algorithm for word segmentation, Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain, July 07–12, 1997, p. 321–328
15. *Surikov A. V.* Lingua::RU::OpenCorpora::Tokenizer Module. Available at URL: http://search. cpan.org/~ksuri/Lingua-RU-OpenCorpora-Tokenizer/
16. *Zioko B., Manandhar S., Wilson R. C.* (2006). Phoneme segmentation of speech. Pattern Recognition. ICPR 2006: 18th International Conferenc, vol. 4, pp. 282–285