УДК 62.506.2

В. И. РУБЛИНЕЦКИЙ, С. Ф. КОРЯК

ОПРЕДЕЛЕНИЕ ГРАНИЦ ФРАЗЫ ПРИ АНАЛИЗЕ ТЕКСТА НА ЭВМ

С развитием кибернетики становится актуальным общение человека с машиной на естественном языке. Эту проблему можно дробить по направлению общения (направление человек→машина, т. е. машинный анализ; направление машина→человек, т. е. машинный синтез) или по форме языка (звучащая речь — письменный текст). Рассмотрим ту часть вопроса, где исследуется машинный анализ текста, а точнее, решается задача, как машине найти границы между фразами в русском тексте обычного формата (со знаками препинания, прописными буквами, абзацами, отступами и т. д.).

Это несложно для человека, и поэтому хотелось бы решить ее на машине без ограничений на вход и с ограничениями на

метод. Первое требование предполагает следующее: работы, посвященные разным задачам машинного анализа текста, в связи с их трудностями ограничиваются текстами узкого семантического и стилистического класса. Возьмем в качестве входного текста любые научно-технические, деловые, публицистические, художественные тексты. Тогда задача осложняется, в чем легко убедиться, если попытаться представить программу, определяющую границы фраз(ы) в следующем, например, тексте:

«Тут молчание настало, долгое,— ну, думаю, наверное ее отчитывает — бог знает, за кого принял! — уж встать хочу, объяснить тому господину, что она — по молодости, и без отца росла, и без всякого там, скажем, какого-нибудь умысла... словом: дура — что...— и вдруг, опять заговорила: «Значит серые? Правда, серые? Нет, вовсе не как у всех людей, а как ни у кого в Москве и на всем свете! Я на лекции была и сама видела, только не знала, серые или зеленые... Вот и выиграла пари... Ура! Ура! Ура!.. Спасибо вам, Андрей Белый, за серые!» (М. Цветаева «Пленный дух»). (1)

Что касается ограничений на метод, то важно понимать следующее: человек, воспринимая предложение, понимает его в ходе восприятия и прогнозирует его продолжение. В машинной системе выделение и понимание фраз удобнее распараллелить: сначала выделить предложение, а затем заниматься его семантико-синтаксическим анализом. Поэтому выделение фразы хотелось бы проводить чисто формальными средствами без всякой опоры на («законченный») смысл. Однако возникают трудности в делении фразы, которые человек, пользуясь более сильными средствами, не замечает. Например, предложение:

«2 августа 1802 г. Наполеон был объявлен пожизненным консулом». (2)

Машинная программа, если не ввести в нее учет аббревиатур, кончающихся точкой, разделяет пример (2) на два предложения до каждой точки.

Прежде чем продолжить алгоритм выделения фраз, необходимо выяснить, как поставленную задачу решает человек. Это необходимо не для моделирования человеческого способа, а для сравнения результатов выделения фраз машиной и человеком с целью отобрать лучший машинный алгоритм.

Традиционная лингвистика много занималась изучением предложения. Вот что говорит по этому вопросу В. А. Звягинцев: «Если Джон Рис в своей книге «Что такое предложение?», вышедшей вторым изданием в 1933 г., привел 139 определений предложения, то к настоящему времени легко можно было бы удвоить, если не утроить количество таких определений» [1]. Понятно, что традиционная лингвистика не владеет общепринятым алгоритмом разделения текста на предложения, иначе не понадобилось бы столько определений. Одна из причин

трудности определения предложения состоит в том, что между предложением и совокупностью предложений нет четкой границы. Ниже делается попытка описать граничные трудности и выработать на смысловом уровне рабочее описание фразы.

Назовем терминальными следующие знаки препинания: . (точка) | ! (восклицательный знак) | ? (вопросительный знак). Примем следующее допущение 1: терминальный знак является необходимым признаком конца предложения. Сформулированное допущение верно в подавляющем большинстве случаев, но не всегда. Например, заголовки, подписи под рисунками, записи в таблице, надписи на чертежах и т. п. не имеют точки в конце. Другой пример — резко индивидуальная авторская пунктуация: так, Эмили Дикинсон кончает многие стихотворения знаком тире. Если принять допущение 1, то остается выяснить, когда терминальный знак является достаточным признаком конца предложения.

Оставляя в стороне примеры типа (2), представляющие трудности для машины, а не для человека, рассмотрим трудности более существенные, где и человек не всегда уверен в делении. В основном они связаны с прямой речью и вводными предложениями. Рассмотрим ряд примеров с прямой речью.

«Я сейчас приду»,— сказал мальчик. (3)

«Кот на заборе!» — закричал мальчик. (4)

«Ой! // Смотрите! // Кот на заборе»,— закричал мальчик. (5)

«Ой! // Смотрите! // Кот на заборе!» — закричал мальчик // — «Идите все сюда». (6)

Пример (3), несомненно, представляет одну фразу, пример (4), видимо, тоже лучше считать одной фразой, хотя внутри ее стоит терминальный знак. Но как трактовать примеры (5) и (6)? Будем считать, что эти примеры состоят из трех и четырех фраз соответственно (границы показаны парой косых скобок). Ясно, что пример (6) отличается от произвольного набора из четырех фраз некоей повышенной связанностью. Будем называть блок фраз типа (6) надфразовым единством с прямой речью (НЕПР). Нетрудно придумать алгоритм, который бы не только делил НЕПР на фразы, но и выделил бы его границы.

Второй типичный случай, где трудно уловить границу между фразами и надфразовыми единствами,— это предложения, в которые вклиниваются вводные предложения со своими терминальными знаками. Рассмотрим такой пример:

«Иванов (а ведь он был здесь в прошлом году!) не узнавал улицы». (7)

Здесь вместо скобок можно также поставить тире. Встречаются случаи, когда в скобках (но не в тире) стоит несколько вводных предложений:

«Последние времена пришли! — кипела она... (А то еще какой-то Александр Блок, что за фамилия такая? Из евреев, должно быть!) сочинил «Прекрасную даму», уж (8)

одно название чего стоит, стыда нет!» (М. Цветаева. «Пленный дух»).

Будем считать, что пример (7) состоит из одной фразы, а пример (8) — НЕПР из двух фраз. Приведенные выше объяснения примеров будем считать рабочим описанием фразы (и прекратим употреблять нерегулярный термин «предложение», как синоним фразы). Эмпирическим аргументом в пользу достаточности такого описания будем считать результат следующего эксперимента: десяти испытуемым дали прочитать приведенное выше описание и после этого они одинаково выделили первых сто фраз в «Пленном духе» М. Цветаевой, а ее синтаксис близок к пределу гибкости и сложности.

Для проверки эффективности машинных алгоритмов было выделено по 100 фраз из следующих текстов:

А: М. Цветаева «Пленный дух» (как образец текста с очень сложным синтаксисом);

Б: И. Тургенев «Отцы и дети» (как образец классической художественной прозы);

В: Козьма Прутков «Черепослов, сиречь френолог» (как образец эмоциональной речи в форме пьесы);

Г: Э. Дейкстра «Дисциплина программирования» (как образец прозы в кибернетическом жанре);

Д: Д. Бухтияров и др. «Сборник задач по программированию на языке ПЛ/1» (как образец технического текста, богатого аббревиатурами, математическими обозначениями и нестандартными употреблениями знаков препинания).

Таблица 1

Тип ошибки \ Тип текста	А	Б	В	Г	Д
Пропущена истинная граница	0	0	0	0	0
Найдена ложная граница	20	11	7	9	30

Таким образом, имея материал для контроля и обучения, можно приступить к выборке алгоритма формального выделения фраз. Обсудим устройство и эффективность серии алгоритмов возрастающей сложности.

Пусть $T = (t_1, t_2, \ldots, t_n)$ — введенный в машину текст, а τ — класс терминальных знаков. Рассмотрим простейший алгоритм A_1. 1° $i := 0$; 2° $i := i + 1$; печать t_i; если $i = n$, то останов; 3° Если $t_i \in \tau$ & $t_{i-1} \overline{\in} \tau$, то печать разделителя //; 4° Перейти к 2°. Условие в 3°: $t_i \in \tau \,\&\, t_{i-1} \overline{\in} \tau$ позволяет однократно учитывать комбинации терминальных знаков (. . . ?? ?! и т. д.). Алгоритм выдает на печать введенный текст с разделителями фраз. Точность алгоритма A_1 показывает табл. 1. Ошибки в текстах А, Б, В в основном проходят по трем причинам: неправильно разделяются фразы типа (4) и (7) и по многоточиям внутри фразы (за многоточием

следует строчная буква), в тексте Г ложные границы фраз устанавливаются по точкам и после инициалов, в тексте Д — в аббревиатурах и формулах. Все нули в первой строке таблицы подтверждают реалистичность допущения 1.

Чтобы изложить следующие алгоритмы, введем некоторые обозначения для классов символов, из которых состоит текст. Будем обозначать B — прописная буква; b — строчная буква; C — цифра; D — другой знак (исключая знаки препинания); α — абзацный пробел; β — простой пробел; γ — кавычки; δ — тире; τ — терминальные знаки; ρ — круглые скобки (ρ_1 — открывающая, ρ_2 — закрывающая); σ — остальные знаки препинания.

Таблица 2

Тип ошибки \ Тип текста	А	Б	В	Г	Д
Пропущена истинная граница	0	0	0	0	0
Найдена ложная граница	5	1	0	8	3

Используя введенные обозначения, можно записать алгоритм A_2:

1° $i := 0$;

2° $i := i + 1$; печать t_i; если $i = n$, то останов;

3° Если $t_i \in \tau \,\&\, t_{i-1} \overline{\in} \tau$, то найти ближайший номер j: $j \geqslant i$, такой что $t_j = B \vee b \vee C \vee D \vee \alpha$;

3.1° Если $j = i + 1$, то перейти к 2°;

3.2° Если $t_j = b$, то перейти к 2°;

3.3° Если $t_j = B \vee C \vee D \vee \alpha$, то печать разделителя // и перейти к 2°.

Точность алгоритма A_2 характеризует табл. 2.

Из оставшихся 17 ошибок 10 возникают из-за ложного деления по точке после инициала, 3 — из-за сокращений, требующих точки (например, табл. 1, причем опасна первая точка, вторая устраняется блоком 3.1); еще 3 ошибки возникают в конструкциях типа (8) и 1 (в столбце В) — обязана сомнительной трактовке многоточия внутри фразы:

«Сын... кандидат... Аркаша...» — беспрестанно вертелось у него в голове. (9)

Чтобы устранить эти ошибки, надо, во-первых, взять словарь *Abbr*, где хранятся популярные аббревиатуры, требующие точек (включая инициалы), и, во-вторых, не учитывать терминальные знаки внутри скобок, входящих во фразу. Последняя оговорка важна, так как в скобках могут стоять автономные фразы и группы фраз. Эти поправки введены в алгоритм A_3.

1° $i := S := 0$ (S — число открытых скобок внутри фразы);

2° $i := i + 1$; печать t_i; если $i = n$, то — останов;

2. 1° Если $t_i = \rho_1$, то $S := S + 1$;

Если $t_i = \rho_2 \,\&\, S > 0$, то $S := S - 1$;

3° Если $t_i \in \tau \,\&\, t_{i-1} \overline{\in} \tau$, то

3.1° Если слово W перед t_i входит в $Abbr$ и $t_{i+1} \neq \alpha$, то перейти к 2°;

Если слово $W \in Abbr \,\&\, t_{i+1} \in \alpha$, то печать разделителя // и перейти к 2°;

3. 2° $j := i$;

3.3° $j := j + 1$; печать t_j; если $j = n$, то печать разделителя // и останов;

3.4° Если $t_j = \beta \vee \gamma \vee \delta \vee \sigma \vee \rho_1$, то — перейти к 3.3°;

3.5° Если $t_j = \rho_2 \,\&\, S > 0$, то $S := S - 1$ и перейти к 3.3°;

3.6° Если $t_j = b$, то перейти к 2°;

3.7° Если $t_j = B \vee C \vee D \vee \alpha \,\&\, (S = 0 \vee t_{j+1} = \rho_2)$, то печать разделителя //, $i := j$; перейти к 2°.

Блок 3.1° в алгоритме учитывает, что точка после аббревиатуры считается концом фразы только в конце абзаца. Блоки 2.1°, 3.5° и 3.7° ведут учет скобок.

Точность алгоритма A_3 на материале обучения — одна ошибка (пример 9) на 400 фраз. Для экзамена алгоритм был проверен на 1000 фраз из источника Г, где он не дал ни одной ошибки, так что, несмотря на принципиальное существование фраз, где алгоритм делает ошибку, его точность можно считать хорошей. Данный алгоритм, обеспечивающий выделение фраз в тексте в совокупности с алгоритмом выделения слов в фразе [2] обеспечивает правильную подачу начальных данных для синтаксического анализа.

Список литературы: 1. *Звягинцев В. А.* Предложение и его отношение к языку и речи. — М.: Изд-во Моск. ун-та, 1976. — 160 с. 2. *Бондаренко М. Ф., Рублинецкий В. И.* Слово в вычислительной лингвистике. — Пробл. бионики, 1983, вып. 30, с. 17—25.

Поступила в редколлегию 05.12.83.