УДК 519.7

В. А. ЧИКИНА

МАШИННОЕ ВЫДЕЛЕНИЕ ПРЕДЛОЖЕНИЯ ИЗ ТЕКСТА
(на материале русского языка)

Большинство программных систем, ориентированных на понимание текста на естественном языке, устроены следующим образом: из текста выделяется предложение, затем проводится его морфологический анализ (иногда и постморфологический, чтобы исключить или уменьшить многовариантность морфологического анализа), синтаксический (иногда определение глубинного синтаксиса), семантический анализ предложения, который может потребовать учета контекста.

Что касается выделения предложения, то обычно применяется простое правило: если стоит точка, а за ней большая буква, то это конец предложения. Поскольку для понимания предъявляются очень простые предложения, то ошибок такое решающее правило дает немного, а те ошибки, что случаются, можно не рассматривать из-за гораздо более многочисленных ошибок на следующих, более сложных, этапах. Это решающее правило, даже в улучшенном и дополненном виде, на типичных, заранее не упрощенных текстах дает около 20 % ошибок [1]. С усовершенствованием компьютеров продолжаются попытки зделать совершеннее прежние программные системы.

Настоящая работа посвящена проблеме усовершенствованию алгоритмов выделения предложений из текста [1]. Считается, что входной текст не прошел никакой предварительной подготовки и относится к произвольному жанру, имеет традиционный формат (со знаками препинания, прописными буквами, пробелами, отступами и т. п.) или современный компьютерный формат, где переносы слов не допускаются (точнее, допускаются только по дефисам: например, можно переносить "нибудь" в слове "какой-нибудь"), где абзацы обозначаются не отступом *перед*, а пустой строкой *за* и применяется выравнивание по правому полю, отчего между словами может стоять несколько пробелов. Для этих двух форматов придется применять несколько иные алгоритмы.

Для решения поставленной задачи возможно использование двух методов: совершенно формального, без опоры на смысл, и с некоторой опорой на смысл (хотя последний полностью должен быть выявлен в конце описанной цепочки программ). Второй метод представляется малоперспективным, поскольку деление "сплошного" смысла на куски соответственно границам предложений очень неоднозначно, как будет показано ниже. Например, М.Таубе [2] пишет: "Бертран Рассел, Уиллард Куайн и большинство современных философов-аналитиков доказали, что именно предложение, а не слово является наименьшим носителем смысла. Конечно, верно, что части предложений имеют свой смысл..., но, с другой стороны, существует множество контекстов, в которых смысловое значение какого-либо предложения выводится из содержания целого параграфа. Для всех случаев, когда в качестве единицы носителя содержания мы могли бы взять сочетание слов, меньшее, чем предложение, найдется столько же случаев, когда в качестве такой единицы придется взять группу предложений или параграфов".

В силу сказанного будем использовать в решающих правилах только формальные признаки. Исследование проведено на материале русского языка. Результаты просто переносятся на другие европейские языки, но они отнюдь не универсальны. Например, семитские языки, где нет ни точек, ни прописных букв, требуют других алгоритмов.

Традиционная лингвистика много занималась изучением предложения. Вот что говорит по этому поводу В.А.Звягинцев [3]: "Если Джон Рис в своей книге: Что такое предложение?, вышедшей вторым изданием в 1933 г., привел 139 определений предложения, то к

настоящему времени легко было бы удвоить, если не утроить, количество таких предложений". Понятно, что традиционная лингвистика не владеет общепринятым алгоритмом выделения предложения из текста, иначе не понадобилось бы столько определений. Разумеется, в решении нашей задачи не поможет определение на уровне обычной схоластическо-бюрократической игры (например, "Автор – то фізична особа, яка творчою працею створила твір"). Во-первых, нужно отметить, что традиционные средства выделения предложения в естественном языке несовершенны, и даже человек, использующий много более сильные средства (он понимает смысл), иногда не может однозначно выделить предложение.

Рассмотрим два примера: обычное повествование

2 августа Наполеон объявлен пожизненным консулом. (1)

и отрывистую дневниковую запись

2 августа. Наполеон объявлен пожизненным консулом. (2)

В первом примере очевидно одно предложение, во втором – два. Толковать ли следующее предложение как (1) или как (2)?

2 августа 1802 г. Наполеон объявлен пожизненным консулом (3)

Трудность состоит в том, что точка имеет несколько функций: ею часто кончаются предложения и менее часто аббревиатуры. Если в каком-то месте точка имеет обе функции, то точка не удваивается, потому неясно, в какой функции она употреблена.

Еще один пример такого же рода основан на том, что восклицательный знак тоже имеет несколько функций. Автор математического текста, желая выразить изумление, что число элементов в некоторых двух классах совпало, пишет:

Таким образом, мы снова имеем n! комбинаций! (4)

А если перифразировать текст:

Таким образом, число комбинаций снова n!? (5)

Что ставить в конце предложения (5)? Если "!", то выйдет n, а не n-факториал, если "!!", то это означает, что целые числа до n перемножаются через одно.

Выше упоминались случаи, когда знаки пунктуации, которыми заканчивается предложения, – будем называть их терминальными знаками – имеют несколько функций, отчего трудно выделить предложение. Эта трудность может происходить и по обратной причине: в конце предложения терминальный знак вовсе не ставится. Так, например, не ставятся точки после заголовков и подписей под рисунками, на вывесках, чертежах, географических картах и т. д. В компьютерном формате точки (и запятые) все чаще заменяются пробелами до конца строки.

Рассмотрим следующее условное (но вполне реальное по пунктуации) деловое письмо:

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "МЕХАНО"
Харьков, пл. Восстания, 22

И.П. Сидорову
Президенту фирмы "Харснаб"
Лозовая ул., 33
Харьков 29.02.97

Просьба ускорить отправку в наш адрес следующих изделий:

1. Подшипники 1ШХ.3 – 100 шт.
2. Гайки Г.6 – 900 шт.
3. Болты Б.12 – 900 шт.

С уважением /Подпись/
А.А.Легостаев
Генеральный директор ООО "Механо"

В этом письме 13 точек, и только одна из них обозначает конец предложения. Первая трудность – состоит в том, что традиционные средства обозначения конца предложения несовершенны. Вторая группа трудностей заключается в том, что в естественном языке нет резких разграничительных линий между предложением и группой предложений.

Вообще, чего мы хотим, когда объединяем некоторые синтагмы в предложение? Например, в Пушкинской "Полтаве" сказано:

Выходит Петр. Его глаза
Сияют. Лик его ужасен.
Движенья быстры. Он прекрасен,
Он весь как божия гроза.

Пушкинская пунктуация разрывает сказанное на короткие предложения и таким способом нагнетает напряженный темп. Можно было бы всюду вместо точек поставить запятые – тогда высказанное представлялось бы более слитным, более единым. Еще более сильное средство объединения представлено написанным выше вводящим цитату единым предположением от "Например" до "гроза". Бывают более сложные комплексы, где прямая речь переплетается с косвенной. Вот пример такого образования из Марины Цветаевой: «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!"»

Итак, был рассмотрен случай, когда цитирование (и прямая речь, как частный случай цитирования) образует некие надфразовые единства. Другим источником аналогичных образований в современном языке являются перечисления. Вот сравнительно простой пример из реального документа.

3.1.1.Расширение производства продукции следующего ассортимента:
- этикетки для алкогольных и безалкогольных напитков;
- этикетки для пищевых консервантов;
- этикетки для продуктов химической промышленности (краски, лаки и т. п.);
- коробки для кондитерских изделий;
- коробки для парфюмерии;
- многокрасочный упаковочный материал для пищевой продукции;
- почтовые поздравительные открытки различных видов;
- листовые настенные календари;
- настольные перекидные календари;
- карманные календари;
- рекламные плакаты, буклеты, листовки.

Вот более сложный пример, взятый из того же документа:

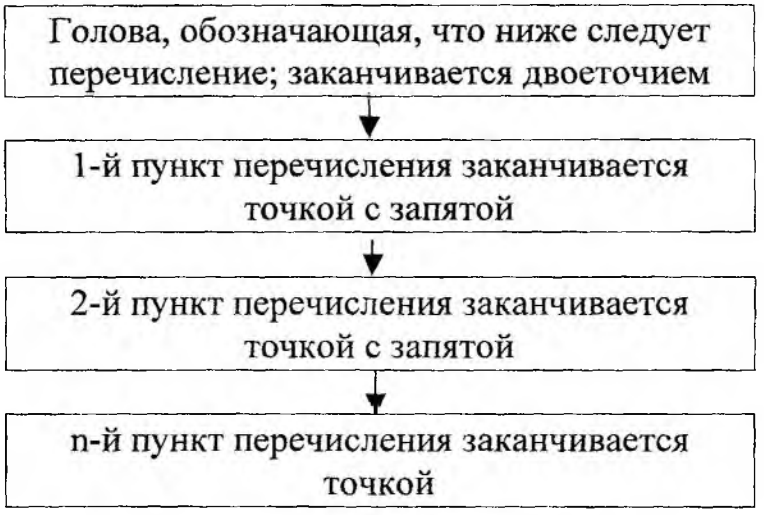
4.3.1. Местные производители:

- полиграфические предприятия, прежде всего государственные. Продукция этих предприятий – невысокого качества, изготовленная на устаревшем оборудовании, не всегда отвечает современным требованиям дизайна и технологии;

- фирмы-посредники, которые ведут активную работу на рынке полиграфпродукции, однако не имеют достаточной маневренности в ценообразовании, так как могут манипулировать только в пределах посреднической надбавки, тогда как производитель способен уста-

навливать систему скидок и применять индивидуальный подход к заказчику на взаимовыгодных условиях.

Общая схема образований показана ниже.



Видно, что перечисляемые пункты разрастаются: первый из них сам состоит из двух предложений, содержит точку как разделитель предложений, но пункт кончается точкой с запятой. Иногда в конце каждого перечисляемого пункта ставится точка, как в следующем примере (9), взятом из диссертации по лингвистике:

Для дистрибутивных наречий можно предложить следующий проект толкований:

N ВСЕГДАдистр. P='в классе (предметов) N все N (имеют свойство) P'. (6)

N ЧАСТОдистр. P= 'в классе (предметов) N есть (предметы) N, которые не (имеют свойства) P, и есть (предметы) N, которые (имеют свойство) P; (предметов) N, которые (имеют свойство) P, больше нормы.

Это образование еще можно толковать как предложение по той формальной причине, что организующая голова кончается двоеточием.

Следующий предельный случай наблюдается, когда голова кончается точкой, как в следующем примере (7).

Предметная именная группа в принципе допускает шесть денотативных (ДС) статусов.

1.Референтный ДС: именная группа отправляет к индивидуальному объекту или множеству объектов, рассматриваемому как единый объект; ср.: _Скрипач играл (скрипачи играли) прямо под окнами_.

2. Экзистенциальный ДС: именная группа отправляет к объекту, который в принципе не индивидуализируем; ср.: _Кто-нибудь из друзей постоянно меня навещает_. (7)

3. Универсальный ДС: он возникает там, где (а) при имени есть или подразумевается квантор общности; (б) само имя не референтно. Пример: _Все дети любят играть_. # (Ср., однако: _Все мои дети любят играть_, – где не соблюдено условие (б)).

Существенны два обстоятельства. Во-первых, универсальная именная группа называет все объекты, которые удовлетворяют данной дескрипции; этим, собственно, и диктуется наличие или возможное наличие квантора общности. Во-вторых, сам по себе он способен связывать не только универсальные, но и референтные именные группы. Таким образом, рассматриваемый ДС формируется не универсальным квантором, но особым принципом объединения денотатов.

... Предикатный ДС присущ именам в функции предикатива при связочном глаголе. (8)

Здесь перечисление выражается лишь в общей организации идей. Видно, что образование, промежуточное между предложением и группой предложений, окончательно стало группой предложений.

Проведем здесь разграничительную линию и будем считать примеры (6 – 8) еще предложениями, пример (7) уже группой предложений. Для описания алгоритма будем использовать русский язык и алгоритмический язык Паскаль. Для начала понадобятся некоторые множества и обозначения, другие будут вводится по ходу описания.

МНОЖЕСТВА:

– endsigns = ['.', '!', '?'] – терминальные знаки;
– digits = ['0' .. '9'] – цифры;
– letters = ['A' .. 'п'] + ['р' ..'я'] – русские буквы;
– capitals =['A' ..'Я'] – заглавные буквы;
– lowcase =['a' ..'n'] + ['p' ..'я'] – строчные буквы.

Сложный вид множеств letters и lowcase обусловлен тем, что в повсеместно принятой кодировке ASCII русский алфавит расположен не подряд. Пусть T – текст, который надо разбить на предложения, а T[i] – текущая рассматриваемая буква.

Весь алгоритм будет представлен как цикл по рассматриваемым буквам, тело цикла будет заканчиваться на метки yes_fin и no_fin, что означает, соответственно, ставь или не ставь знак конца предложения '@' на место T[i+1].

Если T[i] принадлежит множеству endsigns (T[i] in endsigns), то может оказаться, что T[i+1] in endsigns тоже (случай кратных терминальных знаков, вроде '?!' и т. п.). Для выхода на последний терминальный знак служит следующий простейший алгоритм

A1. Выход на последний терминальный знак

```
If T[i] in endsigns then goto no_fin;
```

Еще одна трудность состоит в том, что терминальные знаки внутри скобок и кавычек не означают конца предложения. Чтобы преодолеть эту трудность, надо до головы цикла quclo:=true (quotes closed – скобки закрыты), par:=0 (количество скобок – parentheses – равно 0). Внутри цикла работает алгоритм.

A2. Учет скобок и кавычек.

```
if T[i]='(' then begin dec (par); goto no_fin end;
if T[i]=')' then begin inc (par); goto no_fin end;
if T[i]='"' then begin quclo := not quclo; goto no_fin end;
```

а после метки yes_fin поставить соответствующую проверку.

Самый простой случай конца предложения – это комбинация endsign + пробел+capital. Она разрешается следующим алгоритмом:

A3. Случай endsign + пробел + capital.

```
if (T[i] in endsigns) (and T[i+1] = ' ')
and (T[i+2] in capitals)
then goto yes_fin;
```

Таким образом, последовательность

```
quclo:=true; par:=0;
    for i:=1 to 1 do
        begin
    A1; A2; A3;
yes_fin: if par=0 and cuclo then T[i+1]:='@';
no_fin:
    end;
```

решает простую задачу даже с учетом скобок и кавычек.

Для решения более сложных задач надо между A2 и A3 вставлять все более сложные алгоритмы или усложнять уже описанные.

Следующая сложность – это случай, когда точка (или иной endsign) стоит в конце абзаца и до следующей буквы не один, а много пробелов (а в компьютерном наборе – невидимых знаков, таких, как конец строки, перевод каретки; знак табуляции (0A, 0D, 09 соответственно). Эта трудность преодолевается следующим образом: пишутся функции nextlet(i) и distlet(i), которые находят, пропуская служебные знаки, следующую букву и расстояние (в числе символов) до нее. Достаточно в A3 поставить nextlet вместо T[i+2] и вышеописанная трудность будет преодолена. Еще одна трудность состоит в том, что в русском языке имеются сокращения, содержащие точки (типа "т. е."). К счастью, подавляющее большинство таких сокращений, как можно удостовериться по словарю [4], имеет такую структуру: слово+-точка и подавляющее большинство таких сокращений не может стоять в конце предложения. Поэтому для того, чтобы не находить фальшивых концов предложений после точек в сокращениях, нужно использовать следующий алгоритм:

A4. Обработка сокращений

```
If (T[i] in endsigns)
And (T[i-1] in lowcase)
And (not (N[i-2] in letters))
Then goto no_fin;
```

Алгоритм A4 должен стоять после A2.

Еще одна часто встречающаяся трудность, особенно в компьютерном наборе (см. пример делового письма выше), это – обозначение конца предложения ранним обрывом строки. В этом случае помогает следующий формальный алгоритм. Для его построения понадобится функция size_line(i), которая вычисляет, сколько знаков стоит в данной строке перед смволом T[i]. Понятно, как написать такую процедуру: это цикл от i с уменьшением индекса, пока не встретится перевод строки.

A5. Нахождение конца предложения, не означенного терминальным знаком

```
if (T[i] in letters) or (T[i] in digits)
or (T[i]=')') or T[i]=' " ')
and (size_line(i) < const) and (nextlet(i) in capitals)
and (T[i+distlet(i)+1] in lowcase)
then goto yes_fin;
```

Алгоритм A5 должен быть поставлен после A4.

Изложена лишь часть алгоритмов, поскольку формальный алгоритм выделения предложений – весьма громоздок. Алгоритмы из полного набора могут быть по сложности где-то соразмеримы с представленными. Охактеризуем результат применения программы, напи-

санной по этому алгоритму. Были взяты случайно шесть примеров текстов разных жанров (технический, программистский, общенаучный, деловой, поэтический и прозаический тексты). Программа делала около 3% ошибок на окказиональных сокращениях (типа "субъект" с нижним индексом "пропоз." и тому подобных редкостных образованиях).

Таким образом, формальная программа может служить 501-м определением предложения.

Список литературы: 1. *Рублинецкий В. И.* Определение границ фразы при анализе текста *на ЭВМ* // Проблемы бионики. Вып.34. 1968. С. 34-39. 2. *Таубе М.* Вычислительные машины и здравый смысл. Миф о думающих машинах М.: Прогресс. 1964. 183 с.155. 3. *Звягинцев В. А.* Предложение и его отношение к языку и речи М.: Изд-во Моск. ун-та. 1976. 160 с. 4. *Словарь сокращений русского языка* / Сост.: Д.И. Алексеев, И.Г. Гозман, Г.В. Сахаров / Под ред. Б.Ф. Корицкого. М.: Госиздат иностранных и нац. словарей. 1963. 486 с.

Поступила в редколлегию 4.11.2000