Итеративное применение алгоритмов снятия частеречной омонимии в русском тексте[1]

Iterative application of part-of-speech disambiguation algorithms for russian text

Епифанов М. Е. (xeme@rambler.ru),
Антонова А. Ю. (a-antonova@list.ru),
Баталина А. М. (batalina_anna@rambler.ru),
Кобзарева Т. Ю. (stamstam@mtu-net.ru),
Лахути Д. Г. (delir1@yandex.ru)

Российский государственный гуманитарный университет, Москва

В статье дается краткое обоснование итерационной версии блока алгоритмов анализа частеречной омонимии в русском предложении и описывается соответствующее расширение функциональности инструментальной среды для экспериментов с алгоритмами синтаксического анализа (ЭСЛА).

Введение

Описываемая работа представляет собой один из этапов исследования и реализации возможности автоматизации поверхностно-синтаксического анализа русского текста средствами, использующими только грамматические (морфосинтаксические) характеристики входящих в него слов. Постановка задачи и предварительные результаты этого исследования описывались в материалах конференции ДИАЛОГ ([1]–[3]) и в указанных там источниках.

Разрабатываемая система анализа состоит из семи основных блоков:

- постморфологии (постморфологического анализа), в котором исправляются некоторые ошибки морфологического анализа, обрабатываются несловарные собственные имена и числовые (количественные) группы;
- снятия частеречной омонимии;
- предсегментации — построения некоторых связей: сложного сказуемого, определительных атрибутивных отношений именных групп, предложных групп и т. д., задающих границы проективных фрагментов, минимально необходимых для сегментации;
- сегментации (сегментационного анализа): построение сегментов — простых главных и придаточных предложений, деепричастных и других обособляемых оборотов;
- внутрисегментного анализа — поиск связей слов внутри построенных сегментов;
- межсегментного анализа — построение связей сегментов;
- построения субграфа кореференции.

В своей работе мы исходили из того, что «сколь бы глубоко и тщательно ни разрабатывался синтаксический анализатор системы МП, в нем неизбежно остаются белые пятна и лакуны. Многие лингвистические факты, в том числе критически существенные для работы анализатора, никогда не попадали в поле зрения ученых просто потому, что они не имели возможности в массовом порядке оперировать материалом неверного или неожиданного синтаксического разбора предложений. Именно такой материал в изобилии поставляет развитая система автоматической обработки текстов, а работа с этим материалом позволяет выявить лакуны научного описания и устранить их» [4]. Это значит, что внесение изменений в правила (алгоритмы) работы системы синтаксического анализа должно быть регулярным режимом ее работы. Для того, чтобы это не приводило каждый раз к перепрограммированию всей системы или значительных ее частей, была разработана специальная программная среда — экспе-

[1] Работа выполнена при частичной поддержке гранта РФФИ № 09-06-00275-а.

Dialog'2010.indb 119 11.05.2010 16:57:04

риментальная система работы с лингвистическими алгоритмами (ЭСЛА) [5].

Предметом описываемой работы является блок снятия частеречной омонимии[2]. Мы хотим подчеркнуть, что речь идет не о конкретном, тем более не о *реализованном* блоке (системе алгоритмов) снятия частеречной омонимии (создание которого является одной из *целей* проводимой нами работы), а об *инструменте*, предназначенном для использования при выборе оптимального варианта такого блока (см. [2]).

Блок снятия частеречной омонимии и его повторное применение в процессе анализа

На предшествующих этапах исследования было выделено всего 63 разных типа частеречной омонимии [6]; из них было выделено 15 наиболее часто встречающихся, для которых в экспериментальном режиме были запрограммированы соответствующие модули снятия (распознания) омонимии.

В актуальной версии модули снятия омонимии разных типов работают последовательно в порядке, задаваемом экспериментатором (т. е. сначала предложение обрабатывается модулем снятия омонимии одного типа, затем модулем снятия омонимии другого типа и т. д.). В ходе предшествующих исследований был определен «условно оптимальный» порядок применения 15 модулей снятия омонимии: сложные предлоги, сложные наречия, словосочетания с частицами *не* и *ни* (типа «ни у кого», «ни для чего»), фразеологизмы, приравниваемые к одному слову), вводные слова и конструкции[3], краткое прилагательное — существительное, деепричастие — существительное, существительное — глагол в спрягаемой форме, существительное — прилагательное, предлог — существительное, краткое прилагательное — существительное — списочный глагол, «его — ее — их», существительное в творит. падеже — наречие, морфологически уникальные, но часто встречающиеся омонимы (типа «том», «отчего», «три» и т. п.).

Действие данного блока основано на анализе достаточно близкого синтаксического контекста, причем до собственно синтаксического анализа. При этом слова, сами являющиеся частеречными омонимами, не могут использоваться при анализе, так что при наличии таких слов в контексте обрабатываемого омонима некоторые диагностические конфигурации, используемые в алгоритмах, не будут работать. Поэтому при наличии в предложении более чем одного омонима при однократном последовательном применении указанных модулей некоторые омонимы могут остаться нераспознанными, но будут распознаны при следующем применении (следующих применениях) модулей снятия омонимии.

Рассмотрим примеры, для успешного анализа которых необходимо повторное применение модулей снятия омонимии.

(1) Омонимия «краткое прилагательное (Abr) — существительное (N)»:

*Она не помнит ни **добра**, ни **зла**.*

Омонимы данного типа распознаются справа налево. При первом проходе снимается омонимия ***добра***=N; омонимия ***зла*** остается нераспознанной. При втором проходе определяется ***зла***=N.

(2) Омонимия «(местоименное) прилагательное (A) — (местоименное) существительное (N)», «(местоименное) прилагательное (A) — (местоименное) существительное (N) — существительное (N)» («уникальная»), «краткое прилагательное (Abr) — наречие (D) — списочный предикатив (Vcп)»:

***Все это** говорит о **том**, что емкость рынка **достаточно** велика*.

В соответствующих алгоритмах первый из этих типов омонимии распознается справа налево, второй и третий — слева направо. При первом проходе распознаются все омонимы, кроме ***это***, который распознается при втором проходе.

Говоря в общем, встречаются ситуации, когда в предложении содержится несколько омонимов, причем некоторые их пары зависимы в смысле снятия омонимии: чтобы разрешить омонимию типа **A** для слова **a** требуется предварительно снять омонимию типа **B** для слова **b**. Однако в рамках однократного применения блока снятия омонимии слово **b** анализируется после слова **a**. Если удастся разрешить омонимию для **b** при некотором применении блока снятия омонимии к предложению, то для **a** она снимется при следующем проходе.

Повторное применение блока снятия омонимии снижает (вычислительную) эффективность всей процедуры синтаксического анализа. Разработанная система лингвистических алгоритмов Т. Ю. Кобзаревой — открытая система. В процессе ее совершенствования предполагается ее расширение, это относится и к этапу анализа омонимии: в уже имеющиеся алгоритмы могут добавляться новые правила, что увеличит их размер; анализ пока не рассматриваемых типов омонимии потребует добавления новых алгоритмов (как указывалось выше, сейчас рассматривается 15 типов из 63-х).

---

[2] Далее под омонимией будет везде пониматься частеречная омонимия (омонимия частей речи).

[3] Перечисленные типы рассматриваются как случаи омонимии частей речи.

Но, быть может, изменение организации блока и перестройка его алгоритмов позволит избежать повторных проходов?

К сожалению, ни положительного, ни отрицательного «гарантированного» ответа на этот вопрос не существует. Даже если бы мы (путем значительных усилий) добились такого результата, исходя из некоторого состояния подсистемы алгоритмов на фиксированном множестве рассмотренных примеров с подобной зависимостью снятия омонимии, то мы все равно не смогли бы гарантировать успех однократного применения перестроенного таким образом блока на все новых и новых примерах. К тому же открытость системы алгоритмов потребует новых перестроек при ее расширениях. Интуитивно ясно, что разработать «однократно применяемый» блок (вообще говоря, расширяемый) при всех возможных случаях зависимости снятия одной омонимии от успешности разрешения другой, вряд ли удастся. Доказать же невозможность подобного блока, скорее всего, *невозможно*[4].

При исследовании этого вопроса можно ограничиться только рассмотрением возможности изменения порядка, в котором анализируются «зависимые» омонимы **a** и **b**.

В рассматриваемой подсистеме каждый тип снятия омонимии обрабатывается своим алгоритмом.

Если типы **A** и **B** омонимов **a** и **b** различны, как в примере (2) выше, то порядок их анализа обусловлен порядком применения алгоритмов снятия омонимии этих типов. В описанной выше ситуации (снятие омонимии **A** обусловлено снятием омонимии **B**, но **a** анализируется до **b**) изменение порядка применения алгоритмов при их однократном использовании в процессе анализа «спасет ситуацию» для данного предложения и ему (в этом смысле) подобных в случае, если **B** может быть разрешена независимо от **A**. Но тогда перестанут успешно анализироваться предложения, в которых, наоборот, для разрешения омонимии **B** требуется, чтобы уже была разрешена омонимия **A**. Наличие примеров предложений для обоих порядков применения алгоритмов служит (при данном ограничении) аргументом для повторного применения модулей в блоке снятия омонимии.

В основе составляющих данный блок алгоритмов лежит цикл прохода по словам-омонимам предложения в прямом (слева направо) либо обратном порядках. На каждом шаге этого цикла для обрабатываемого омонима анализируются контексты с целью выявления возможности снять омонимию. Если **A=B**, то последовательность рассмотрения слов **a** и **b** определяется направлением такого прохода, как в примере (1) — справа налево. В алгоритме правила анализа контекстов рассматриваемого слова, вообще говоря, сформулированы с учетом направления прохода. В принципе их можно изменить в соответствии с изменением на противоположное направления прохода, однако в случаях, когда они уже достаточно хорошо отлажены, или некоторые из них могут усложниться, делать это нежелательно. Кроме того, изменение направления прохода может привести к тому, что повторное применение блока теперь потребуется для других предложений, содержащих другую зависимость между омонимами типа **A**, рассмотренную в других правилах алгоритма.

Заметим, наконец, что наличие в предложении нескольких подобных зависимостей может привести к необходимости повторять применение модулей снятия омонимии несколько раз.

Итеративная версия блока снятия частеречной омонимии

Универсальным и наиболее простым подходом к реализации повторов анализа омонимии является итеративная версия блока, в которой все составляющие его алгоритмы применяются однократно в определенном порядке при каждой итерации. Схематично основной цикл блока можно описать как

```
while Not IsEqual(S_curr, S_prev) do
  begin
    S_prev:=Copy(S_curr);
    применить последовательно модули
    снятия омонимии к S_curr
  end
```

Цикл выполняется, пока применение алгоритмов «плодотворно», т. е. удается снять хотя бы одну какую-либо омонимию. Как только состояние объектной модели анализируемого предложения после очередной итерации S_{curr} совпадает с таким состоянием после предыдущей итерации S_{prev} («новых изменений нет»), цикл прекращается, переходим к следующей стадии синтаксического анализа.

В рассмотренной в предыдущем пункте ситуации, если на некотором шаге цикла будет снята омонимия **B**, то на следующем шаге разрешится и омонимия **A**.

Итеративная версия блока программно реализована, как и другие лингвистические алгоритмы, на входном языке системы ЭСЛА. На уровне ядра

[4] Приведенные здесь кратко рассуждения аналогичны рассуждениям, мотивирующим построение теории NP-полных задач (см., например, Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи // М.: Мир, 1982). Однако, ввиду открытости самой системы алгоритмов и сложности ограничения (фиксации) случаев зависимости снятия одной омонимии от успешности снятия другой, не удается адекватно сформулировать «практически осмысленную» задачу, для которой можно было бы доказать ее NP-полноту.

интерпретатора ЭСЛА реализованы, как системно-определенные «примитивы», функции сохранения состояния объектной модели анализируемого предложения и сравнения двух состояний объектных моделей.

По мнению авторов, применение итеративного блока снятия омонимии делает принципиально несущественным порядок анализа омонимов в предложении.

Заметим, что в приведенном здесь цикле последняя итерация всегда осуществляется «вхолостую», она служит лишь основанием для того, чтобы, убедившись в отсутствии вновь разрешенных омонимов, завершить выполнение блока. Так, в примерах (1) и (2) цикл будет прекращен лишь при проверке условия в начале четвертого шага. В случаях, когда в предложении успешно снимаются все омонимии анализируемых алгоритмами блока типов, последняя итерация выполняется быстрее предыдущих. Это связано с тем, что на последнем шаге цикла каждый модуль не находит омонимов «своего» типа и сразу прекращает работу.

Выше уже отмечалось, что повторное применение алгоритмов снятия омонимии снижает эффективность всего синтаксического анализа. Поэтому актуально выявление статистически значимых возможностей такого изменения порядка анализа омонимов, после которого для довольно часто встречающейся зависимости снятия омонимии количество итераций уменьшается.

Другим способом осуществить подобную оптимизацию является непосредственный вызов одного алгоритма из другого. Точнее, если снятие омонимии **A** обусловлено снятием омонимии **B** и в анализируемом (в алгоритме снятия **A**) контексте встретился омоним типа **B**, то к нему прямо здесь же и применяется модуль для **B**. Однако такие рекурсивные вызовы одного алгоритма из другого усложняют алгоритмы (увеличивается количество ветвлений) и тем самым затрудняют их последующую модификацию и отладку.

Расширение функциональности инструментальной среды

С целью более удобного проведения экспериментов с алгоритмами анализа частеречной омонимии и их отладки разработана форма, специализированная на совершенствование рассматриваемого здесь блока. В ней пользователь может

- выбрать подмножество модулей, которые он хочет применить в процессе анализа тестового предложения. Эти актуальные модули выбираются не только из тех, что составляют блок снятия омонимии, но и из входящих в блок постморфологии, т. к. последний применяется до анализа омонимии. Применение только части модулей, и, в частности, единственного алгоритма, может существенно упростить исследование поведения реализованных в узлах алгоритмов правил, обеспечивающих распознавание актуальных для анализа данного предложения ситуаций, в том числе возможности снятия омонимии в конкретных контекстах. Эта же функциональная возможность при отладке облегчает локализацию ошибок программной реализации алгоритмов и неточностей в них самих.
- задать порядок применения выбранных модулей, что позволяет оптимизировать структуру блока. Список выбранных пользователем модулей в заданном порядке сохраняется в виде отдельного файла.
- указать, применять ли выбранные модули итеративно или однократно.
- просмотреть результат анализа предложения непосредственно в одном из полей формы и сохранить его в виде отдельного файла.
- просмотреть *трассу прохождения* — последовательность действительно посещенных узлов применяемых алгоритмов в процессе анализа предложения непосредственно в одном из полей формы. Эта трасса также сохраняется в отдельном файле-журнале.

На основании сформированных пользователем данных (выбор модулей и порядок их применения, указание, применять ли их итеративно) генерируется так называемый *основной*[5] (головной) алгоритм на входном языке системы ЭСЛА, который и применяется к тестовому предложению. В основном алгоритме осуществляются вызовы модулей из заданной их последовательности, причем такая последовательность вычисляется либо итеративно, либо однократно. Код основного алгоритма сохраняется в отдельном файле и, при необходимости, может быть использован для пошаговой отладки в интерпретаторе ЭСЛА [3].

Все перечисленные выше *выходные* файлы помещаются в папку так называемого *проекта выполнения* (прогона, runtime project) сгенерированного основного алгоритма, созданную в процессе его применения к тестовому примеру. Туда же будут положены и другие, связанные с данным прогоном алгоритма *исходные*, *промежуточные* и выходные файлы:

- с текстом тестового примера,
- с результатом морфологического анализа предложения,
- с исходным состоянием объектной модели предложения, представленном во входном языке ЭСЛА и полученном соответствующей конверсией результата морфоанализа,

[5] Аналог основной функции или процедуры в процедурных языках программирования, например функции main() в C или C++.

- xml-файлы с данными для показа результата синтаксического анализа в одной из двух предназначенных для этого программ (разработанная в последнее время интерактивная подсистема визуализации результата синтаксического анализа обладает существенно более широкими функциональными возможностями, в ней удобно просматривать графы синтаксического разбора больших предложений, в то время как прежде использовавшийся модуль и сейчас часто применяется для «быстрой» оценки правильности результата).

Заключение

Реализация и последующее применение итеративной версии блока снятия частеречной омонимии расширило класс успешно анализируемых предложений.

В настоящее время проводится отладка блока, проводятся эксперименты с целью оптимизации его структуры. Эту работу существенно облегчает специально разработанный инструмент, позволяющий применять алгоритмы блока к тестовому предложению избирательно и в нужном порядке.

Литература

1. *Кобзарева Т. Ю., Лахути Д. Г., Ножов И. М.* Модель сегментации русского предложения. // Диалог'2001. Аксаково 2001. Т. 2, с. 185–194.
2. *Баталина А. М., Айриян Г. Ю., Епифанов М. Е., Кобзарева Т. Ю., Лахути Д. Г.* Автоматизация отладки алгоритмов поверхностно-синтаксического анализа // Труды международной конференции «Диалог 2005». М.: Наука, 2005. С. 45–50.
3. *Баталина А. М., Епифанов М. Е., Кобзарева Т. Ю., Кушнарёва Е. В., Лахути Д. Г.* Экспериментальная реализация сегментационного анализа русского предложения // Тр. Междунар. конф. «Диалог 2007». М. РГГУ, 2007. С. 29–34.
4. *Иомдин Л. Л.* Уроки русско-английского (из опыта работы системы машинного перевода) // Компьютерная лингвистика и интеллектуальные технологии: Труды Международной конференции Диалог'2002. Т. 2. С. 234–244.
5. *Баталина А. М., Епифанов М. Е., Ивличева О. О., Кобзарева Т. Ю., Лахути Д. Г.* Инструментальная среда для экспериментов с алгоритмами поверхностно-синтаксического анализа // Компьютерная лингвистика и интеллектуальные технологии: Труды Международной конференции «Диалог 2004», М.: Наука, 2004. С. 32–38.
6. *Кобзарева Т. Ю., Афанасьев Р. Н.* Универсальный модуль предсинтаксического анализа омонимии частей речи в РЯ на основе словаря диагностических ситуаций // Компьютерная лингвистика и интеллектуальные технологии. Тр. Междунар. сем. Диалог 2002. Т. 2. М., 2002. С. 258–268.