# СТАНДАРТНЫЕ ТЕСТЫ ДЛЯ ЗАДАЧ АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТОВ НА РУССКОМ ЯЗЫКЕ И РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ[1]

# STANDARD TESTS FOR NATURAL TEXT PROCESSING TASKS FOR RUSSIAN AND REGRESSION TESTING

***Богуславский И.М.*** *(bogus@iitp.ru),* ***Дружкин К.Ю.****(druzhkin@yandex.ru),* ***Иомдин Л.Л.****(iomdin@iitp.ru),*
***Сизов В.Г.****(sizov@iitp.ru),* ***Цинман Л.Л.****(cinman@iitp.ru)*
*ИППИ РАН, Москва*

Рассматриваются подходы к построению тестов для оценки параметров автоматических систем обработки текстов, в первую очередь, качества и устойчивости синтаксического анализатора. Описывается метод построения тестов для оценки лингвистического процессора ЭТАП-3 при работе с русским языком в качестве входного. Система находится в процессе становления и реализована лишь частично. Авторы надеются, что эти тесты будут применимы для оценки других систем обработки русских текстов.

## *1. Вводные замечания*

В европейской и американской компьютерной лингвистике, начиная с середины 1990-х годов, стало обычной практикой использование специальных тестов, проверяющих работу систем автоматической обработки текстов, в первую очередь, систем машинного перевода (см. об этом, например, [Lehrberger and Bourbeau 1988, Harrison *et al*. 1991, Arnold *et al*. 1993, Nyberg *et al*. 1994, Oepen and Flickinger 1998, Carroll *et al*. 2002]). Прежде всего, такие тесты дают разработчику обратную связь, указывая на слабые места его системы. Кроме того, сравнительный анализ работы систем, основанных на разных принципах, позволяет понять преимущества и недостатки разных подходов. И, наконец, результаты тестирования помогают пользователю выбрать из нескольких систем обработки текстов именно ту, которая лучше соответствует его целям. В этом последнем случае работа по созданию тестов может даже финансироваться потенциальным потребителем системы.

Если тестирование проводится для получения обратной связи, то тестировщик в то же самое время является разработчиком, и работа системы анализируется по принципу «прозрачного ящика»: для каждой неудачи, продемонстрированной тестом, должна быть найдена и устранена конкретная причина (неполнота словаря, недоработка или просто ошибка в тексте правила). Если же тестирование проводит заказчик или внешний эксперт, не имеющий доступа к внутреннему устройству системы, то в этой ситуации тестировщик работает с системой как с «черным ящиком», рассматривая и оценивая только результаты, но не причины неудач.

Критерии оценки распадаются на две группы: это (1) покрытие материала и (2) правильность полученного результата (например, синтаксического анализа) сравнительно с неким эталоном. Примерами критериев первой группы могут служить: количество синтаксических конструкций, которые система знает; доля предложений в корпусе, которым она смогла приписать какую-нибудь синтаксическую структуру; размер словаря; доля опознанных слов в корпусе. Примерами критериев второй группы могут служить: доля правильно проанализированных предложений; доля слов, правильно отнесённых к той или иной части речи.

Оценить степень покрытия материала легко, но эта информация мало полезна. Оценить правильность анализа сравнительно с эталоном – гораздо полезнее, но и гораздо труднее; причём основные трудности вызывает именно разработка эталона. Дело в том, что, составляя эталон, разработчик не всегда может использовать уже готовые результаты своих коллег. К примеру, разные системы синтаксического анализа выдают результаты разной степени подробности и в разных формализмах, часто плохо соотносимых друг с другом. Проблеме отыскания общего основания для сравнения систем посвящена обширная литература (обзор работ на эту тему см. в [Carroll *et al*. 1998]). В частности, предлагается конвертировать структуру составляющих в структуру зависимостей и сравнивать между собой результаты этой конверсии [Lin 1995].

[1] Настоящая работа осуществляется при поддержке Российского фонда фундаментальных исследований (грант № 05-06-80256-а), которому авторы выражают искреннюю признательность.

Создание универсального синтаксического эталона, по-видимому, невозможно; по крайней мере, до тех пор, пока системы автоматической обработки текста пользуются различными подходами к анализу языкового материала.[2] Но составление корпуса искусственных фраз, покрывающего все значимые явления данного языка, является вполне реалистичной задачей. Для английского, французского и немецкого языков такие корпусы составляются, в частности, в рамках крупного проекта TSNLP (Test Suites for Natural Language Processing) [Lehmann *et al*., 1996].

Следует отметить, что, насколько известно авторам, сколько-нибудь представительного набора тестов для оценки параметров систем автоматической обработки текстов на русском языке не существует. Начатая авторами работа, основные элементы которой приводятся ниже, призвана восполнить этот пробел. Система тестов строится в расчете на ее использование в лингвистическом процессоре «ЭТАП-3» [Апресян и др. 1989, 1992, Apresjan *et al*. 2003] и частично уже инкорпорирована в этот процессор. В первую очередь система предназначена для тестирования результатов синтаксического анализа предложений русского языка; кроме того, в ближайшие планы авторов входит также применить ее к результатам автоматического перевода текстов с русского языка на английский.

### *2. Основа тестирования*

Для тестирования компьютерно-лингвистической системы нужны три вещи: во-первых, корпус текстов; во-вторых, эталонный разбор этого корпуса; в-третьих, метрика, то есть способ измерения и оценки расхождений между эталоном и тем, что делает система.

Сделаем сразу же следующее терминологическое уточнение. В англоязычной литературе, посвящённой проблемам тестирования парсеров и систем машинного перевода, собрание связных текстов или связных отрывков текстов называется корпусом (“corpus”), а набор специально подобранных и не связанных между собою фраз называется тестовой последовательностью (“test suite”). Нам представляется более наглядным пользоваться для этой цели терминами “естественный корпус” и “искусственный корпус”, соответственно.

### *3. Естественный корпус vs. искусственный корпус*

**Естественный корпус** состоит из реальных текстов на данном языке; причем подбираются тексты такого жанра и такой тематики, на которые рассчитана система. **Искусственный корпус** – это совокупность специально подобранных фраз, каждая из которых должна иллюстрировать некоторое грамматическое явление.

Естественный корпус отражает реальное употребление языка. Кроме того, он создаётся легко и быстро. С другой стороны, такой корпус неизбежно содержит слишком много однотипных примеров на частотные явления и слишком мало примеров на редкие явления. Маленький естественный корпус будет недостаточно представительным; а с большим естественным корпусом трудно работать.

Напротив, искусственный корпус сравнительно невелик, но отражает все явления, которые, по мнению разработчика, система должна охватить. Использование искусственного корпуса для тестирования представляется наилучшим решением; но, к сожалению, хороший искусственный корпус трудно составить. Проблема заключается в том, что искусственный корпус неизбежно отражает **хотя и все классы явлений, но не все подклассы этих классов**. Представим себе, например, что мы вводим в искусственный корпус конструкции, в которых существительное зависит от предлога – это класс явлений. Те же случаи, когда между предлогом и существительным, скажем, стоит инфинитив, – это частный случай внутри класса. Разумно предположить, что словосочетания типа *для усталого и желающего отдохнуть человека* не попадут в искусственный корпус. Таких частных случаев великое множество, и если составитель искусственного корпуса захочет подобрать по примеру на каждую возможность, он рискует углубиться в бесконечные мелочи и никогда не закончить свою работу. Но именно на мелочах такого рода по-настоящему проверяется качество синтаксического анализатора и автоматического перевода.

### *4. Фиксированный результат vs. идеальный результат*

Эталон может фиксировать уже достигнутое системой качество анализа и/или перевода или, напротив, указывать на желательное состояние системы в будущем.

---

[2] Для английского языка эталонным синтаксически размеченным корпусом считается Penn Treebank, в котором каждому предложению сопоставлена синтаксическая структура в виде дерева составляющих. К сожалению, этот корпус размечен не слишком последовательно.

При работе с **фиксированным эталонным результатом** отслеживаются все изменения, возникшие на небольшом промежутке времени.

Работа происходит следующим образом: корпус "прогоняется" через систему, и результаты фиксируются, образуя эталон. При каждом изменении системы (или через определенные промежутки времени) этот же корпус снова подаётся на вход системы, и результат его обработки сравнивается с эталоном. Различия отыскиваются автоматически; затем эксперт просматривает список предложений, получивших отличный от эталона анализ или перевод, находит то изменение системы, которое вызвало расхождение, и, если изменение было ошибочным, оперативно его пересматривает. Таким способом гарантируется неухудшение параметров системы при внесении изменений. Если сравнение результата обработки корпуса с эталоном показывает улучшение работы системы, то этот результат фиксируется в качестве нового эталона.

Такую процедуру тестирования мы называем **регрессионным тестированием** – по аналогии с соответствующей методикой, активно используемой в программировании сложных систем в последние десятилетия. Оперативный режим регрессионного тестирования позволяет разработчику вовремя заметить небольшие шаги системы к лучшему или к худшему. Однако на основании такого теста нельзя дать глобальную оценку работе системы.

Напротив, при работе с **идеальным эталонным результатом** следует исходить из того, что каждое расхождение эталонного корпуса с реальным синтаксическим разбором или переводом является ошибкой системы. Таким образом, измерив количество и качество расхождений, мы получаем информацию о количестве и качестве ошибок системы. Такой эталон заставляет разработчика ставить перед собой далеко идущие цели. Сравнение происходит не между вчерашним и сегодняшним состоянием системы, а между ее сегодняшним состоянием и (практически недостижимым!) идеалом. На фоне огромного количества расхождений плохо заметны текущие изменения в системе.

Об идеальном эталонном результате можно сказать то же самое, что и об искусственном корпусе текстов: это хорошее, но технически сложное решение. Оно наталкивается на ряд проблем, в особенности тогда, когда в качестве эталона принимается результат (человеческой) обработки естественного корпуса.

Сказанное можно пояснить следующим рассуждением. Как хорошо известно, многие предложения естественного текста неоднозначны. Чтобы выбрать единственно верную для данного текста синтаксическую структуру, нужны, помимо всего прочего, энциклопедические знания о мире, которых у компьютера нет. Поэтому компьютер должен: (а) порождать множество альтернативных синтаксических разборов и (б) ранжировать эти разборы по степени их вероятности. Значит, чтобы проверить его работу, следовало бы иметь для каждого предложения эталонное множество альтернативных разборов; такое множество, в котором есть все правильные разборы и нет ни одного неправильного. Но эта задача почти невыполнима: она требует огромного труда.

Более простое решение – приписать каждому предложению один, наиболее "естественный" разбор и сравнивать его с первым из вариантов, предложенных компьютером. Но что значит "естественный"? Когда компьютер ранжирует разборы по степени их вероятности, он опирается на формальные признаки; а человек оценивает вероятность того или иного понимания, опираясь на контекст и на смысл. Поэтому то, что "вероятно" по оценке компьютера, может быть невероятно для человека и наоборот. Например, возьмём фразу

(1) *Как ни жаль, он не придёт.*

Может ли слово *жаль* выступать здесь как императив глагола *жалить*? С точки зрения компьютера (иными словами, с точки зрения системы, описывающей общие языковые закономерности), очень даже может; ср.(1) и устроенную весьма похожим образом фразу *Как ни проси, он не придёт.* С точки же зрения человека – носителя русского языка подобная интерпретация (1) представляется невероятной и допустимой разве что в порядке языковой игры.

Тем не менее, нам приходится считаться с тем, что в естественном корпусе заметную долю составляют предложения, для которых парсер предлагает разбор, хоть и допустимый с точки зрения языковых правил, но неподходящий с точки зрения здравого смысла или описываемой ситуации. Это особый вид ошибок, который также представляет интерес для оценки. Очевидно, что разные парсеры могут по этому параметру значительно отличаться.

Другая проблема при тестировании возникает оттого, что при построении синтаксической структуры иногда приходится принимать произвольные решения вследствие принципиальной неединственности синтаксического описания. Рассмотрим, к примеру, сочетание обстоятельства времени с аналитической формой глагола:

(2) *Завтра я буду работать.*

Вопрос о том, от какой словоформы зависит в (2) *завтра* – от *буду* или от *работать* – представляется мало принципиальным. Мы были бы готовы принять обе структуры как правильные; но как это сделать, если эталонный разбор только один? Другой, чуть более сложный, пример такого рода – предложения типа

(3) *Издалека реки не было видно,*

в котором слово *реки* можно было бы расценивать либо как подлежащее (по аналогии с *Издалека река не была видна*), либо как дополнение (по аналогии с *Издалека реку не было видно*). Возможное решение здесь – переместить проблему в другую область, а именно, в систему оценки интерпретации результатов расхождений.

### *5. Оценка расхождений*

#### *5.1. Аварийная сигнализация vs. оценочный механизм*

Здесь мы должны ещё раз подчеркнуть разницу между регрессионным тестом и оценочным тестом. Регрессионный тест не предназначен для сколько-нибудь глобальной оценки работы системы. Скорее, он похож на аварийную сигнализацию: любое расхождение между эталонной структурой и фактическим результатом должно быть передано на суд человека. В противоположность этому, при работе с оценочным тестом заранее ожидается большое количество расхождений. Человек не может рассматривать каждое из них. Поэтому появляется задача **вычисления некоторой общей оценки.**

#### *5.2. Типизация ошибок*

Процедура оценки расхождений реальной работы системы с эталоном во многом определяется тем, рассматриваем ли мы «ошибки» как единую категорию или разбиваем их на типы и подтипы. Даже такая элементарная задача, как подсчёт количества ошибок, может быть выполнена тремя разными способами:

(а) **Нетипизированная система.** Подсчитывается суммарное количество любых ошибок.

(б) **Плоская типизированная система.** Выделяется несколько типов ошибок, и по каждому типу ведётся отдельная статистика. Минимальное количество типов – два (например, морфологические и синтаксические ошибки); верхнего предела нет.

(в) **Иерархическая типизированная система.** Выделяются большие типы ошибок, внутри них подтипы, которые в свою очередь делятся на ещё более мелкие типы. Минимальное количество ярусов в иерархической системе - опять-таки два (например, верхний ярус - морфология и синтаксис, нижний ярус - различные подклассы внутри морфологии и синтаксиса); количество ярусов принципиально не ограничено.

Вот так, к примеру, может выглядеть результат подсчёта ошибок при развитой системе типизации:

1. Всего ошибок – столько-то.

1.1. Неполнота словаря – столько-то.

1.2. Лексико-морфологических – столько-то.

1.2.1. Неправильно определена часть речи – столько-то

…

1.3. Синтаксических – столько-то.

1.3.1. Неправильное синтаксическое отношение – столько-то.

1.3.1.1. Сирконстантное отношение вместо актантного – столько-то.

…

Такие важные показатели системы, как полнота (recall) и точность (precision) также могут быть отражены с помощью типизации ошибок. Рассмотрим следующие две схемы определения типов:

В эталоне есть X, а в тестируемой структуре на месте X-а есть что-то другое.

В тестируемой структуре есть X, а в эталоне на месте X-а есть что-то другое.

Все типы, определённые по схеме (1), показывают неполноту представления X-а в тестируемой структуре.

Все типы, определённые по схеме (2), показывают неточность представления X-а в тестируемой структуре.

Поясним это на примере синтаксиса. Существует крупный класс ошибок: «выбрано неправильное синтаксическое отношение».

Используя схему (1), здесь можно задать множество подклассов, каждый из которых будет отражать неполноту представления конкретного синтаксического отношения. Например:

Выбрано неправильное синтаксическое отношение, притом что правильное – обстоятельственное (предикативное/ агентивное/ сочинительное…).

Используя схему (2), можно задать множество подклассов, каждый из которых будет отражать неточность представления конкретного синтаксического отношения. Например:

Обстоятельственное (предикативное / агентивное/ сочинительное…) отношение выбрано неправильно.

### *5.3. Что подсчитывается?*

Процедура оценки определяется также и тем, что именно подсчитывается: простое количество ошибок в тексте, количество ошибок на единицу текста или соотношение ошибок и не-ошибок.

**(а) Подсчитывается количество ошибок в корпусе.** Если корпус фиксирован, этот метод позволяет сравнивать разные парсеры, а также разные состояния одного парсера. Например: тот парсер сделал 335 ошибок, а этот – только 117. Или: вчера наш парсер сделал 117 ошибок, а сегодня – только 90. Если мы не просто фиксируем ошибку, но и запоминаем, в каком предложении она была сделана, можно определять процент правильно проанализированных предложений, процент предложений, содержащих одну ошибку, две и т.д. Этот метод привлекателен прежде всего своей простотой.

Типизация ошибок в этом случае открывает новые возможности, в частности:

(1) подсчёты могут вестись по каждому типу отдельно,

(2) за ошибки разного типа может назначаться разное количество штрафных баллов.

**(б) Количество ошибок делится на единицу текста.** В зависимости от целей разработчика за единицу можно принять словоформу, предложение или более крупный фрагмент текста. В зависимости от целей и здесь можно использовать типизацию ошибок. В результате можно получить, например, такие показатели:

(1) Среднее по корпусу количество синтаксических ошибок в предложении.

(2) Доля словоформ с морфологическими ошибками в общем количестве словоформ.

Такой метод позволяет оценивать не только работу разных парсеров над данным текстом, но и сложность разных текстов для данного парсера[3]. Становится также возможным делать обобщения о типах текстов (например: газетные тексты обрабатываются хуже научных).

**(в) Подсчитывается полнота и точность отражения некоторых характеристик эталона в оцениваемом массиве.** Этот способ предполагает развитую типизацию ошибок и наиболее сложен алгоритмически, но он же приносит и самые интересные результаты.

Например, пусть $N_1$ – количество узлов, подчинённых своему хозяину по отношению R в эталоне, $N_2$ – количество узлов, подчинённых своему хозяину по синтаксическому отношению R в оцениваемом массиве, $N_3$ – количество узлов, подчинённых своему хозяину по отношению R и в эталоне и в оцениваемом массиве. Тогда частное от деления $N_3$ на $N_1$ – это полнота отражения R, а частное от деления $N_3$ на $N_2$ – это точность отражения R.

### *5.4. Сопоставимость оценок*

Разумно ожидать, что между разработчиками автоматических систем обработки текстов существует гораздо большее согласие по вопросам морфологии, чем по вопросам синтаксиса, поэтому ошибки первого рода можно характеризовать в общепринятых (нейтральных) терминах. Напротив, характеристика синтаксических ошибок сильно зависит от того, каким формализмом пользуется разработчик. В самом деле, неправильная расстановка скобок (при разборе по составляющим) не может быть напрямую сопоставлена с выбором неверного синтаксического отношения (при построении дерева зависимостей).

Поэтому, говоря о неполноте словаря и неправильном морфологическом анализе, авторы описывают проблемы, общие для всех разработчиков, а при обсуждении синтаксиса авторы вынуждены говорить о своей конкретной системе.

## *6. ЭТАП-3*

Тестирование русского синтаксического анализатора системы ЭТАП-3 планируется проводить в двух режимах – регрессионном и глобальном.

При регрессионном тестировании будет использоваться искусственный корпус, содержащий массив предложений, анализируемых системой правильно в начальный момент времени. Этот корпус формируется на основе материала русских синтаксических правил (синтагм), применяемых в синтаксическом анализаторе лингвистического процессора «ЭТАП-3»: для каждого такого правила строится несколько примеров, отражающих основные подклассы данного правила. Отбирая примеры, мы стремимся к разумному компромиссу между полнотой охвата языковых явлений и степенью их представительности. Синтаксических правил в настоящее время насчитывается около пятисот. Соответственно, общий объем этой части регрессионного корпуса составит две-

[3] Два текста имеют одинаковый уровень сложности для данного парсера, если среднее количество ошибок парсера при их обработке одинаково.

три тысячи предложений. Помимо этого регрессионный корпус будет пополняться за счет отладочных предложений, появляющихся в ходе экспериментов и получающих правильную структуру.

Глобальное тестирование будет проводиться на основе естественного корпуса. В качестве такового мы используем фрагмент разработанного в Лаборатории компьютерной лингвистики ИППИ РАН глубоко аннотированного корпуса русских текстов Syntagrus (см. о нем, например, [Apresjan *et al*. 2006]). Этот корпус составлялся с помощью синтаксического анализатора системы ЭТАП-3, но прошел обязательную стадию редактирования результата анализа. Поэтому он может служить в качестве идеального эталона.

В соответствии со сказанным в разделе 4, для работы с идеальным эталоном мы разными способами упрощаем себе задачу: (1) мы создаём только один эталонный синтаксический разбор; (2) мы сравниваем его только с одним (первым) из автоматически полученных разборов; (3) при интерпретации результатов мы закрываем глаза на некоторые виды расхождений.

Как говорилось выше, регрессионное тестирование не нуждается в выработке механизма оценки: все расхождения с эталоном передаются человеку для анализа. Иначе обстоит дело с глобальным тестированием, которое в таком механизме нуждается. Ниже описывается используемая нами система глобального тестирования синтаксического анализатора. Система оценки результатов тестирования автоматического перевода находится в процессе разработки.

Сравнению подлежат: эталонная синтаксическая структура (СинтС) предложения (Э) и СинтС предложения, построенная процессором в ходе тестирования (Т). Прежде чем непосредственно описывать систему тестирования, покажем, как выглядят те структуры, которые мы сравниваем.

### *6.1. Представление дерева зависимостей в формате XML*

Возьмём фразу *Волки выли на луну*. Её структура в формате XML (с некоторыми сокращениями) выглядит так[4]:

*<**S ID**= ”463” **TRANS**= ”The wolves howled at the moon“>*
*<**W ID**= ”1” **KSNAME**= ”ВОЛК“ **FEAT**= ”S МН МУЖ ИМ ОД“ **DOM**= ”2” **LINK**= ”предик“>Волки</**W**>*
*<**W ID**= ”2” **KSNAME** = ”ВЫТЬ“ **FEAT**= ”V НЕСОВ ИЗЪЯВ ПРОШ МН“ **DOM**= ”_root”>выли</**W**>*
*<**W ID**= ”3” **KSNAME** = ”НА“ **FEAT**= ”PR” **DOM**= ”2” **LINK**= ”обст“>на</**W**>*
*<**W ID**= ”4” **KSNAME** = ”ЛУНА“ **FEAT**= ”S ЕД ЖЕН ВИН НЕОД“ **DOM**= ”3” **LINK**= ”предл“>луну</**W**>.*
*</**S**>*

Здесь элементы S и W означают «предложение» и «словоформа», соответственно. Для каждого предложения указываются: его порядковый номер в тексте корпуса (ID) и английский перевод (TRANS). Для каждой словоформы указываются: порядковый номер в предложении (ID), имя лексемы, соответствующей данной словоформе, в комбинаторном словаре (KSNAME), часть речи и морфологические признаки словоформы (FEAT), порядковый номер хозяина этой словоформы (DOM), а также имя синтаксического отношения между ней и хозяином (LINK).

Отношение синтаксической зависимости («стрелка») в XML-представлении непосредственно не отражается, а представлено в виде двух атрибутов зависимой словоформы – DOM и LINK. У вершины предложения атрибут DOM принимает значение «_root»; атрибут LINK, естественно, отсутствует.

Слова, не представленные в словаре, получают следующие атрибуты: KSNAME=”ФИКТ-ЛЕКС” (фиктивная лексема), FEAT=”NID” (nonidentified). Узлы, которые не нашли своего места в структуре, присоединяются к некоторому узлу предложения (обычно ближайшему) с помощью фиктивной синтаксической связи (LINK=”fictit”).

Таким образом, древесная структура представлена в виде строки, а сравнение двух структур сводится к сравнению двух строк. Мы берём пару элементов одного типа и сравниваем, во-первых, атрибуты этих элементов, а во-вторых, содержание этих элементов.

### *6.2. Типизация расхождений*

Два узла (в Э и в Т) не имеют расхождений, если все их атрибуты совпадают.

Выделяются следующие типы расхождений:

**I. Расхождения, вызванные неполнотой словаря**

Для узла в Т отсутствует словарная статья.

**II. Лексико-морфологические расхождения**

Для узла в Т присутствует словарная статья, но выбрана другая лексема.

[4] Подчеркивание и жирный шрифт добавлены для удобства чтения.

Для узла в Т выбрана та же лексема, но другой набор морфологических характеристик.

**III. Синтаксические расхождения**

Узел является вершиной в Э, но не является вершиной в Т.

Узел является вершиной в Т, но не является вершиной в Э.

Узел в Т сменил своего хозяина.

Узел в Т зависит от своего хозяина по другому отношению.

Узел в Т зависит от своего хозяина по фиктивной связи.

### *6.3. Система штрафных баллов*

В настоящее время каждое расхождение либо штрафуется одним баллом, либо игнорируется. В частности, игнорируются следующие расхождения:

1. Несовпадения в присоединении обстоятельств (как выше в примере (2)).
2. Интерпретация предложной группы как атрибута существительного или обстоятельства при глаголе (ср. *видел город в тумане*).
3. Использование некоторых синтаксических отношений вместо других, но близких к ним.

По мере накопления материала тестирования планируется развитие системы штрафных баллов. Также в дальнейшем планируется вычислять полноту и точность установления отдельных типов связей.

### *6.4. Некоторые технические проблемы*

В процессе оценочного тестирования возникают определенные технические трудности, связанные с особенностями системы ЭТАП-3, которые необходимо учитывать при интерпретации результатов. Они связаны с тем, что в эталоне и оцениваемом массиве текстов соответствующие предложения могут иметь разное количество словоформ. Это происходит в двух случаях.

Случай первый: **сложные слова.**

Сложные слова типа *нефтепереработка* проходят этап морфологического анализа по-разному. Если такое слово есть в словаре, то оно получает свой набор морфологических характеристик и передаётся синтаксическому анализатору. Если в словаре его нет, но зато есть *нефть* и *переработка*, то слово разрывается на две части, и в процессе синтаксического анализа обе части связываются синтаксическим отношением «композ». Предположим, что за время между созданием эталона и тестированием слово *нефтепереработка* было добавлено в словарь. Тогда в эталонной структуре оно будет представлено двумя узлами, а в оцениваемой структуре – уже только одним.

Случай второй: **эллипсис.**

ЭТАП-3, вообще говоря, не рассчитан на полноценную обработку предложений с эллипсисом. При разметке текстов для Национального корпуса русского языка (НКРЯ)[5] такие предложения правятся вручную: на место эллипсиса вставлялся «фантомный» узел. Например, предложение *Вася уехал в Москву, а Петя – в Казань*, получает синтаксическую структуру, в которой появляется еще один (фиктивный) глагол, который принимает на себя все синтаксические связи опущенного слова: *Вася уехал в Москву, а Петя* [фиктивный_глагол] *в Казань*. Если использовать синтаксически размеченный фрагмент НКРЯ в качестве эталонного корпуса, то в эталонной структуре может оказаться больше узлов, чем в оцениваемой.

С расхождениями такого рода приходится поступать следующим образом. Если удаётся точно указать условия, когда они появляются, они игнорируются при оценке. В противном случае сравнение двух предложений оказывается невозможным.

### *6.5. Примеры*

Приведем в заключение два примера тестовой оценки предложений из естественного корпуса.

(4) *Его разработали специалисты компании "АММ-2000", занимающейся созданием ультразвуковых приборов для медицинской диагностики.*

В эталонной СинтС: причастие *занимающейся* является формой глагола *заниматься,* от него в существительное *создание* ведет 1-ая комплетивная связь.

Тестовая СинтС интерпретирует это причастие как форму страдательного залога глагола *занимать*; от него в существительное *создание* ведет агентивная связь.

[5] См. http://www.ruscorpora.ru/search-syntax.html

**Оценка:**
за ошибку в выборе лексемы (*заниматься*): 1 штрафной балл
за ошибку в установлении отношения 1-компл: 1 штрафной балл
Итого: штрафных баллов 2.
(5) *Я 40 лет в политике.*
Эталонная СинтС: предик(В, Я); длит(В, ЛЕТ).
Построенная СинтС: предик(ЛЕТ, Я), обст(ЛЕТ, В).
**Оценка:**
за ошибку в определении вершины (*в*): 1 балл;
за ошибку в установлении хозяина отношения предик: 1 балл;
за ошибку в установлении отношения длит: 2 балла.
Итого: штрафных баллов 4.

***Список литературы***

1. Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л., Лазурский А.В., Перцов Н.В., Санников В.З., Цинман Л.Л. Лингвистическое обеспечение системы ЭТАП2. М: Наука, 1989. 295 стр.
2. Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л., Лазурский А.В., Митюшин Л.Г., Санников В.З., Цинман Л.Л. Лингвистический процессор для сложных информационных систем. М: Наука, 1992. 256 стр.
3. Arnold D., Moffat D., Sadler L., Way A., 1993. Automatic Test Suite Generation. // *Machine Translation*, vol. 8, Nos. 1-2, p. 29-38.
4. Apresian Ju.D., Boguslavsky I.M., Iomdin L.L., Lazursky A.V., Sannikov V.Z, Sizov V.G., Tsinman L.L. ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT. // *MTT 2003, First International Conference on Meaning – Text Theory.* Paris: École Normale Supérieure, 2003, p. 279-288.
5. Apresjan Ju.D, Boguslavsky I.M, Iomdin B.L, Iomdin L.L, Sannikov A.V., Sizov V.G. A Syntactically and Semantically Tagged Corpus of Russian: State of the Art and Prospects. // *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006).* Genoa: 2006, p. 1378-1381.
6. Carroll J., Briscoe E., and Sanfilippo A. 1998. Parser evaluation: a survey and a new proposal. // *Proceedings of the 1st International Conference on Language Resources and Evaluation*, Granada, Spain, p. 447–454.
7. Carroll J., Frank A., Lin D., Prescher D., and Uszkoreit H. (Eds.). 2002. Beyond PARSEVAL - Towards Improved Evaluation Measures for Parsing Systems. Workshop at the 3rd International Conference on Language Resources and Evaluation (LREC-02). Las Palmas, Gran Canaria, Spain.
8. Harrison P., Abney S., Fleckenger D., Gdaniec C., Grishman R., Hindle D., Ingria B., Marcus M., Santorini B., and Strzalkowski T. Evaluating syntax performance of parser/grammars of English. // *Proceedings of the Workshop on Evaluating Natural Language Processing Systems, ACL.*, 1991.
9. Lehmann S., Oepen S., Regnier-Prost S., Netter K., Lux V., Klein J., Falkedal K., Fouvry F., Estival D., Dauphin E., Compagnion H., Baur J., Balkan L. and Arnold D. 1996. TSNLP — test suites for natural language processing. // *Proceedings of the International Conference on Computational Linguistics, COLING-96*. Copenhagen, Denmark: p. 711–716.
10. Lehrberger J. and Bourbeau L., 1988. Machine Translation: Linguistic characteristics of MT systems and general methodology of evaluation. John Benjamins.
11. Lin D. 1995. A dependency-based method for evaluating broad coverage parsers. // *Proceedings of International Joint Conference on Artificial Intelligence* (IJCAI-95), Montréal, Canada: p. 1420–1427.
12. Nyberg E., Mitamura T. and Carbonell J. (1994). "Evaluation Metrics for Knowledge-Based Machine Translation. // *Proceedings of the International Conference on Computational Linguistics, COLING-94,* Kyoto, Japan: p. 95-99.
13. Oepen S. and Flickinger P. 1998. Towards systematic grammar profiling. Test suite technology ten years after. // *Journal of Computer Speech and Language 12(4)*, p. 411–435.