Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов

Ю.Г. Зеленков, И.В. Сегалович, В.А. Титов
Яндекс, Москва
{yuryz,iseg,uht}@yandex-team.ru

Снятие омонимии полезно во многих приложениях компьютерной лингвистики, в частности, в поисковых системах оно может повысить точность обработки некоторых классов запросов и/или сократить объем хранимой информации.

Существующие подходы к снятию омонимии традиционно разделяются на детерминированные (развиваемые с 60-х годов), то есть основанные на локальном или глобальном синтаксическом разборе и синтаксических словарях, и на вероятностные (начиная примерно с Brill, 1995), использующие статистику совместной встречаемости грамматических признаков слов в больших корпусах, омонимия в которых снята заранее.

В данной работе предложен оригинальный подход, основанный на словарях контекстов, построенных на небольшом, отобранном по оригинальной процедуре и вручную размеченном корпусе. Элементом контекста в данной работе впервые предложено использовать нормализующую подстановку. Предложенный алгоритм позволяет реализовать эффективный и быстродействующий автономный модуль для снятия омонимии с высокой точностью, создает словари необходимых синтаксических правил любого объема и обеспечивает поиск в них за константное время, являясь, по сути, специализированным вариантом локального синтаксического анализа, настроенным на решение конкретной проблемы.

Цель

Алгоритм предназначен для разрешения морфологической омонимии у слов, совпадающих не во всех, а лишь в нескольких грамматических формах (чаще всего в одной, двух или трех, но иногда и больше), т.е. у *омоформ*. Примеры: *три*, *трем* — формы числительного и глагола, *стекло*, *стих*, *стали* — существительного и глагола, *белка*, *пара* — существительных мужского и женского рода, *вина* — существительных женского и среднего рода, *кос* — существительного и краткого прилагательного и т.п.

Актуальность и обзор аналогов

Актуальность проблемы определяется тем, что практически все существующие алгоритмы снятия омонимии включаются в состав синтаксического анализа, что создает трудноразрешимое противоречие, когда для успешного снятия омонимии необходимы точные результаты синтаксического анализа, для получения которых, в свою очередь, нужно предварительно снять омонимию. Кроме того, значительный объем исходного числа связей существенно замедляет обработку, приводя к т.н. «комбинаторному взрыву».

В качестве примера таких алгоритмов можно, в первую очередь, привести соответствующие компоненты лингвистического процессора ЭТАП [1] и синтаксического анализатора Диалинг [2].

В первом из них (ЭТАП) используется «фильтровый метод» синтаксического анализа, при котором сначала строится полный набор допустимых гипотетических синтаксических связей между словами анализируемой фразы, а затем из этого набора удаляются (фильтруются) недопустимые связи (синтагмы). Этот процесс продолжается до тех пор, пока оставшиеся связи не образуют дерево, являющееся искомой синтаксической структурой. Система «Диалинг» основана на комбинировании различных

вариантов анализа фразы. Здесь также морфологическая омонимия крайне негативно отражается на скорости работы.

В качестве примера альтернативного «локального синтаксического» подхода, основанного на использовании всевозможных синтаксических правил типа предложного управления или согласований главного и подчиненного слов, можно привести морфологический анализатор английского языка ENGTWOL [3]. Эта система включает словарь основ объемом 56000 единиц и вручную составленную базу данных, содержащую более тысячи правил, являющихся запретами на появление определенных последовательностей грамматических классов в текстах.

Все вышеописанные алгоритмы относились к классу детерминированных или «основанных на правилах». Другим возможным подходом к снятию неоднозначности является использование вероятностных методов или обучающих примеров, взятых из вручную размеченных корпусов.

Иллюстрацией такого подхода может служить система, разработанная М. Харст для снятия неоднозначности у существительных, являющихся омографами, на основе локальных контекстов с использованием больших корпусов (по материалам обзора [4]). При обучении системы правильные значения многозначного слова вместе с наборами некоторых грамматических и лексических свойств контекста собираются в определенную структуру данных. Этот этап называется "управляемым обучением". Затем алгоритм строит такие же контексты для неразмеченных употреблений слова и выполняет "самостоятельное обучение", выбирая правильные значения. Точность работы алгоритма для различных слов составляла от 73% до 100%. Аналогичная система, построенная Д. Яровски, имеет точность порядка 96%.

Другим широко известным вероятностным подходом является алгоритм, основанный на использовании скрытой Марковской модели (Hidden Markov Model (HMM) tagging). Основная идея алгоритма

заключается в том, чтобы для каждого слова, входящего в предложение, выбрать грамматический класс (тэг) таким образом, чтобы максимизировать функцию:

*P(word|tag)*P(tag|previous n tags)*, где

P(tag|previous n tags) - условная вероятность (вычисленная по размеченному корпусу), появления данного тэга *tag*, при условии, что предыдущие *n* тэгов уже определены.

P(word|tag) - условная вероятность (также вычисленная по корпусу) появления в данном месте слова *word*, при условии, что это слово имеет данный грамматический класс *tag* [3].

Алгоритм HMM имеет довольно высокую вычислительную сложность (реализуется классическим методом динамического программирования Витерби, подобном используемому при вычислении расстояния Левенштейна) и на практике обычно применяются различные упрощающие предположения, направленные на ее снижение (например, использование не более чем трехсловных последовательностей).

Точность алгоритма HMM для английского языка составляет 96%. Применение данной модели для русского языка может быть затруднено, поскольку потребует размеченных корпусов очень большого объема, учитывая богатство русского словообразования и словоизменения по сравнению с английским.

Еще одним подходом к снятию морфологической неоднозначности является комбинированный подход, сочетающий лучшие возможности как детерминированных так и вероятностных «таггеров». Одним из наиболее известных подходов такого родя является TBL (Transfomation-Based Learning) таггер или морфологический анализатор Брилла (Brill tagger) [5]. Данный метод основан на идеях машинного обучения, при котором правила назначения грамматических классов словам извлекаются автоматически из

предварительно размеченных корпусов и затем применяются к обрабатываемым текстам в духе детерминированных алгоритмов.

При обучении данного алгоритма сначала всем словам размеченного корпуса присваиваются наиболее вероятные значения тэгов. Затем исследуются всевозможные трансформации тэгов (продукционные правила) и оставляются те, которые максимально улучшают качество разбора (с точки зрения уже размеченного корпуса). В результате обучения создается упорядоченный список правил (процедура разбора), который может в дальнейшем применяться при обработке новых корпусов.

Основная идея

В основе данной работы лежит несколько ключевых идей.

1. Использование небольшого, тщательного отобранного и размеченного вручную корпуса как источника построения словаря контекстов омонимов.

2. Естественно предположить, что элементы контекста сильнее или слабее влияют на выбор значения омонима в зависимости от их расположения относительно омонима. В данной работе приоритет влияния соседей выражен в численной форме на основе простой вероятностной модели.

3. В данной постановке целью алгоритма является получение леммы слова, то есть выбор между несколькими продукционными правилами преобразования словоформы в возможные леммы. У флективных языков нормализующие подстановки являются достаточно универсальным средством выражения грамматических свойств, поэтому мы рискнули попробовать использовать именно их в качестве элементов контекста омоформы.

4. При построении корпуса используется идея ранжирования частотных омонимов русского языка по степени «трудности выбора леммы». Трудность выбора леммы аппроксимируется гипотетическим размером

корпуса, необходимым для уверенного разрешения данной неоднозначности. Кроме того, в построении корпуса учитывался принцип максимального пропорционального разнообразия жанров и тем.

В качестве корпуса для обучения алгоритма сначала использовался аннотированный корпус с морфологической разметкой и снятой омонимией из проекта *«Национальный корпус русского языка»* [7]. Затем был создан аннотированный корпус, объемом более 5 Мб. Процедура отбора текстов для этого корпуса была автоматизирована с помощью специально разработанного генетического алгоритма (ГА), который обеспечивает оптимальную репрезентативность выборки документов из Web по ряду наиболее важных показателей (разнообразие явлений омонимии, жанров, тематик и т.п.). Описание ГА приведено далее в этом документе.

Исходные данные

Исходными данными для работы алгоритма являются результаты обработки исходного текста программой морфологического разбора *mystem* [6], где для каждой словоформы определяется одна или несколько (для омонимов) лемм вместе с наборами граммем, представляющими основные грамматические характеристики (часть речи, род, число, падеж, лицо и т.п.).

Пример

```
потом{пот=S,муж,неод=твор,ед|потом=ADV=}
```

Результаты работы

Результаты работы алгоритма в основном представляются в том же виде, что и исходные данные, но для омонимов список лемм дополнительно

упорядочивается по убыванию вероятности выбора этих лемм в качестве значения омонима в текущем контексте с указанием этих вероятностей.

Пример

```
потом{потом:0.91=ADV=|пот:0.09=S,муж,неод=твор,ед}
```

Словарь контекстов

В качестве основной структуры данных, используемой при снятии омонимии, выступает словарь контекстов, который устроен следующим образом. Базовой единицей словаря является тройка *<омоним, элемент контекста, лемма>*, которая является элементарным случайным событием состоящим в том, что при появлении в тексте данного омонима и данного элемента контекста, в качестве значения этого омонима будет выбрана данная лемма. Каждому элементарному событию приписывается определенная вероятность его наступления на основе автоматической обработки аннотированных корпусов со снятой омонимией.

Для уменьшения объема словаря элементы тройки представляются в виде *нормализующей подстановки*, состоящей из трехбуквенного окончания словоформы, за которым в скобках указывается, сколько и на что нужно заменить последние буквы словоформы, чтобы получить лемму. Для омонима указывается несколько вариантов замен (по числу лемм). При том, что у флективных языков нормализующие подстановки являются достаточно универсальным средством выражения грамматических свойств, они, как оказалось в ходе экспериментов обладают высокой чувствительностью к контекстному выбору значения омонима.

Примеры нормализующих подстановок для обычных слов

мал(1ть)	думал{думать=V,несов=прош,ед,изъяв,муж}
ере(1ь)	звере{зверь=S,муж,од=пр,ед}
огу(1а)	дорогу{дорога=S,жен,неод=вин,ед}
ной(4я)	мной{я=S,ед,од=(твор,жен|твор,муж)}

ись(9орачиваться)	повернувшись{поворачиваться=V=прош,деепр,сов}
щем(1)	туловищем{туловище=S,сред,неод=твор,ед}
ерх(0)	вверх{вверх=ADV=}
нно(1ый)	неуклонно{неуклонный=A=ед,кр,сред}

Примеры нормализующих подстановок для омонимов

чал(0о|2инать)	начал{начинать=V=прош,ед,изъяв,муж,сов|начало=S,сред,неод=род,мн}
ерь(1ить|1ять)	поверь{поверить=V,сов=ед,пов,2-л|поверять=V=ед,пов,2-л,сов}
рка(0|1|1ий)	марка{марк=S,муж,од=(род,ед|вин,ед)|марка=S,жен,неод=им,ед|маркий=A=ед,кр,жен}
рую(1ить|1я)	струю{струить=V,несов=непрош,ед,изъяв,1-л|струя=S,жен,неод=вин,ед}
гко(0|1ий)	легко{легкий=A=ед,кр,сред|легко=ADV=}
мой(0|2ыть)	мой{мой=A=(им,ед,муж|вин,ед,муж,неод)|мыть=V,несов=ед,пов,2-л}
дит(0ь|4аживать)	выходит{выхаживать=V=непрош,ед,изъяв,3-л,сов|выходить=V=непрош,ед,изъяв,3-л,несов}

Границы предложений, знаки препинания и слова, не являющиеся русскими, представляются специальными лексемами.

Элемент контекста дополнительно включает координаты его расположения относительно омонима: "–1" — левый сосед, "+1" – правый и т.д.

Увеличение числа букв в окончании словоформы до четырех и более не влияет на точность обработки, но приводит к увеличению размеров словаря на 10–12%.

В процессе разработки изучались и другие варианты представления структуры контекста, в т.ч. как совместное с подстановками, так и отдельное использование наборов граммем (т.е. грамматического класса) слов. Эти варианты в ходе нашего эксперимента хотя и сократили объем словаря, но

показали меньшую точность при снятии омонимии, что явилось гораздо более важным аргументом не в их пользу.

Примеры записей словаря контекстов

ала(1о|3инать)	[р]+2	1о	0.67
ала(1о|3инать)	[р]+2	3инать	0.33
ала(2новиться|2ть)	и(0)-2	2новиться	0.14
ала(2новиться|2ть)	и(0)-2	2ть	0.86
его(3он|3оно)	с(0)-1	3он	0.98
его(3он|3оно)	с(0)-1	3оно	0.02
его(3он|3оно)	с(0)-2	3он	1
его(3он|3оно)	с(0)-3	3он	0.94
его(3он|3оно)	с(0)-3	3оно	0.06
его(3он|3оно)	с(0)+1	3он	1
его(3он|3оно)	с(0)+2	3он	0.95
его(3он|3оно)	с(0)+2	3оно	0.05

Ранжирование элементов контекста

Естественно предположить, что элементы контекста сильнее или слабее влияют на выбор значения омонима в зависимости от их расположения относительно омонима, а при равенстве позиций — от соотношения вероятностей элементарных событий, относящихся к различным леммам.

Например, элемент контекста, выбирающий одну из двух лемм с вероятностью 0.8, а другую с вероятностью 0.2, очевидно «сильнее» элемента с соотношениями вероятностей 0.6 и 0.4, а тот, в свою очередь, «сильнее» элемента с вероятностями 0.5 и 0.5. Последний элемент, по-видимому, является самым «слабым», т.к. он вообще ничего не может выбрать. Точно также, если в некоторой позиции соотношение математических ожиданий выбора одной из двух лемм, вычисленных по всем элементам контекста,

находящимся в данной позиции, составляет 0.7 и 0.3, а в другой — 0.6 и 0.4, то первая позиция «сильнее» второй, т.к. у нее более «резко» выражены предпочтения.

В качестве функции, с помощью которой удобно оценивать степень влияния элемента контекста на выбор леммы, можно использовать слегка измененную формулу энтропии К. Шеннона, которая для вероятностей *p* и *q* (т.е. для случая двух лемм), таких что *p* + *q* = *1*, записывается в виде:

$$F(p,q) = 1 + p \cdot \log_2 p + q \cdot \log_2 q$$

Основным свойством этой функции можно считать то, что чем ближе *p* и *q* друг к другу (чем слабее выражены предпочтения), тем меньше значение функции (сила влияния) и наоборот. Вероятности *p* и *q* можно рассматривать как своего рода меру количества информации о значении омонима, содержащейся в элементе контекста. Рассуждения сохраняют силу для любого количества лемм.

В результате обработки *«Национального корпуса русского языка»* с помощью данной формулы был определен следующий приоритет среди элементов контекста омонима в пределах 10 ближайших соседей (по 5 слева и справа): –1, +1, –2, –3, +2, –4, +3, –5, +4, +5, т.е. соседние элементы слева и справа, затем следующие два элемента слева, следующий элемент справа и т.д. Если принять за 1 «силу» влияния соседнего слева элемента, то для первых пяти самых «сильных» элементов получаются следующие значения:

–1	1.00
+1	0.97
–2	0.93
–3	0.89
+2	0.88

У оставшихся элементов (–4, +3, –5, +4, +5) происходит резкое уменьшение «силы» влияния, что позволяет без ущерба для точности расчетов их отбросить. Это сокращает размер основного словаря на 35–40% и увеличивает скорость обработки на 10%.

Выбор значения омонима

Процесс снятия омонимии происходит следующим образом. Сначала для каждого омонима исходного текста и его ближайших соседей по результатам морфологического разбора строятся их нормализующие подстановки. Затем для каждой пары <*омоним, элемент контекста*> из словаря контекстов выбирается лемма и вероятность ее порождения данным элементов контекста. Далее, для каждой леммы вычисляется сумма вероятностей, умноженных на значение «силы» элемента контекста. Значением омонима в данном контексте считается лемма с наибольшей взвешенной суммой вероятностей.

Если две наибольшие суммы отличаются не более чем на 10–15%, то для повышения надежности выбирается наиболее вероятная лемма с учетом глобального контекста (используется метод «включения парадигм по корпусу текстов» [6]) или вообще без учета контекста (с помощью частотного словаря). В случае невозможности принятия решения выбирается лемма минимальной длины, а при равенстве длин — первая по алфавиту.

Вероятность правильного выбора значения омонима с помощью данного алгоритма составляет, на сегодняшний день, порядка 0.94–0.95 и, при необходимости, может быть увеличена с помощью автоматизированного обучения.

Примеры снятия омонимии в реальных текстах приведены в приложении.

Ранжирование омонимов

Для эффективного обучения системы необходимо использовать текстовые документы, в которых были бы наиболее полно представлены типичные явления омонимии в максимально разнообразных контекстах. Первая задача (*типизация*) решается на основе упорядочения списка омонимов, имеющихся в языке, на основе некоторого рангового критерия, а вторая (*разнообразие*) — с помощью ГА, описанного ниже.

Общее правило, — чем больше необходимо обработать контекстов для уверенного выбора леммы, тем выше ранг омонима.

Для вычисления ранга омонима используется формула:

$$R = \frac{A \cdot B}{(C + 0.5) \cdot (D + 0.5)}, \text{ где}$$

A — частотность,

B — омонимичность,

C — «расстояние» между парадигмами,

D — «расстояние» между частями речи,

0.5 — коэффициенты, равные половине интервала изменения C и D (от 0 до 1) и введенные для учета случаев, когда C или D равны 0.

Частотность A равна $\frac{\log_2 q}{\log_2 q_{\max}}$, где q — частота встречаемости омонима в базе запросов Яндекса, а $q_{\max}$ — максимальная частота.

Омонимичность B учитывает разнообразие лемм и равна $\frac{\log_2 s}{\log_2 s_{\max}}$, где s — количество разных лемм у омонима, т.е. $\log_2 s$ представляет максимальную энтропию (неопределенность) при выборе лемм, а $s_{\max}$ — максимально возможное количество лемм у омонимов (=6–8).

Синтаксическая характеристика *C* показывает степень «омоформности» лемм омонима и определяется с помощью следующего правила. Сначала для каждой пары лемм вычисляется отношение пересечения множества словоформ, входящих в парадигмы этих лемм, к их объединению. Затем определяется *среднее арифметическое* полученных дробей, и результат вычитается из единицы. Полученное значение и называется «расстоянием» между парадигмами омонима.

Семантический показатель *D* отражает смысловую дистанцию между частями речи и равен *среднему арифметическому* расстояний между этими грамматическими категориями для каждой пары лемм омонима, вычисленных с помощью *контекстной метрики*. Определение контекстной метрики и примеры различных лингвистических расстояний, полученных на ее основе, приведены далее.

Примеры омоформ с высоким рангом

света	1.58
родам	1.58
сони	1.46
духи	1.46
свет	1.45
трусы	1.42
роды	1.42
пары	1.34
тома	1.32
родов	1.23
белка	1.23
поля	1.22
графики	1.20
курсы	1.19
полка	1.14

Введенный ранговый критерий является *абстрактным* показателем омонимичности. Для учета неодинаковой частоты встречаемости разных лемм омонима в реальных текстах необходимо дополнить ранг омонима множителем, равным *энтропии омонима*, вычисленной на основе аннотированных корпусов. В результате мы получим показатель *реальной* омонимии.

Некоторые статистические показатели:

- исходный список омонимов был получен на основе словаря Зализняка и включал 178 295 омонимов;
- после применения эвристических фильтров осталось 52 536 омонимов;
- после применения частотного фильтра осталось 22 678 омонимов, которые и были упорядочены на основе вышеприведенного критерия.

Контекстная метрика для вычисления лингвистических расстояний

Контекстная метрика является удобным инструментом для количественной оценки степени сходства или различия между разнообразными элементами языка. В ее основе лежит понятие *дистрибуции* или сочетания какого-либо лингвистического объекта (словоформы, леммы, граммемы и т.п.) с окружающими объектами в тексте.

При практическом построении дистрибуций, как правило, используются только ближайшие элементы контекста (*m* элементов слева и *n* справа в пределах предложения), хотя, в общем случае, такое требование не является обязательным.

Сформированная дистрибуция представляет собой упорядоченную по уменьшению вероятностей последовательность всех элементов контекста

объекта в пределах аннотированного корпуса. В случае больших корпусов можно, дополнительно, ввести минимальные пороговые значения вероятностей для сокращения размеров дистрибуций.

Сходством *s(a,b)* между двумя объектами *a* и *b* (например, значениями слов при нахождении синонимов, падежами существительных, сочетаниями вида и залога у глаголов или частями речи при их кластеризации и т.п.) будем называть сумму минимальных (из двух) значений вероятностей элементов контекста, входящих в обе дистрибуции этих объектов. Другими словами, при вычислении «сходства» находим общие элементы двух дистрибуций, выбираем у каждого элемента минимальное значение вероятности и суммируем их.

Тогда *«расстояние»* между объектами будет равно $d(a,b) = 1 - s(a,b)$.

Далее приведены некоторые примеры использования введенной контекстной метрики для вычисления «расстояний» между различными лингвистическими объектами. При формировании дистрибуций использовалась часть *«Национального корпуса русского языка»* со снятой омонимией.

Пример 1. «Расстояния» между частями речи

	S	A	NUM	V	ADV	PR	CONJ	PART	INTJ
S	0.00	0.26	0.43	0.30	0.30	0.38	0.36	0.36	0.65
A		0.00	0.39	0.32	0.34	0.32	0.39	0.37	0.69
NUM			0.00	0.46	0.45	0.45	0.52	0.48	0.71
V				0.00	0.28	0.37	0.40	0.32	0.66
ADV					0.00	0.42	0.34	0.27	0.45
PR						0.00	0.44	0.44	0.74
CONJ							0.00	0.40	0.66
PART								0.00	0.64
INTJ									0.00

Пример 2. «Расстояния» между падежами существительных

	им	род	дат	вин	твор	пр
им	0.000	0.450	0.453	0.422	0.494	0.525
род		0.000	0.468	0.390	0.424	0.433
дат			0.000	0.422	0.483	0.519
вин				0.000	0.417	0.439
твор					0.000	0.469
пр						0.000

Пример 3. «Расстояния» между сочетаниями вида и залога у глаголов (в скобках указаны образцы словоформ)

	несов+действ (называющий)	несов+страд (называемый)	сов+действ (назвавший)	сов+страд (названный)
несов+действ (называющий)	0.000	0.501	0.202	0.396
несов+страд (называемый)		0.000	0.525	0.508
сов+действ (назвавший)			0.000	0.415
сов+страд (названный)				0.000

Генетический алгоритм отбора текстов для обучения

Алгоритм реализует эффективный отбор документов из Web на основе критерия максимального разнообразия явлений омонимии, контекстов омонимов, тематических рубрик, жанров, хостов и пр., характеризующих эти документы, с целью создания аннотированных корпусов для формирования словаря контекстов и обучения модуля снятия омонимии.

Особенностью алгоритма является запоминание наиболее приспособленных особей и популяций для инициализации последующих

циклов обучения. Это позволяет ускорить сходимость алгоритма и уменьшить количество порождаемых поколений во время очередного цикла.

Другая особенность состоит в том, что в качестве функции, оценивающей разнообразие омонимов выбрана не традиционная энтропия К. Шеннона, которая отражает преимущественно синтагматические отношения между объектами, а взвешенная:

$$H(p,q) = -A \cdot p \cdot \log_2 p - B \cdot q \cdot \log_2 q ,$$

где p и q — вероятности омонимов в документе, а A и B — их ранги, отражающие иерархию между ними. Такой подход дополнительно позволяет учитывать и парадигматические связи между омонимами. В реальных документах, естественно, омонимов будет не два, а значительно больше, но все рассуждения сохраняют свою силу.

При дополнительной фильтрации документов на основе максимального разнообразия тематических рубрик, жанров, хостов и пр. в качестве критерия фильтрации используется обычная энтропия, вычисленная для этих показателей.

Результатом работы ГА является выбор определенного количества документов, удовлетворяющих критерию максимального разнообразия некоторых объектов (омонимов, контекстов) и свойств (тематик, жанров, хостов), представленных в них.

Другими словами, задача состоит в поиске глобального экстремума функции взвешенной энтропии, заданной на некотором множестве наборов документов размера k (=100–200 документов), выбираемых из универсального множества размера n (=3000–5000 документов).

Универсальное множество формируется на основе документов из базы Яндекса, имеющих максимальное сходство с ранжированным списком омонимов. В качестве меры сходства используется взвешенная по вероятностям сумма рангов омонимов базового списка, входящих в

документ. Вероятность омонима равна отношению числа появлений омонима в тексте документа к общему числу слов документа. При этом размеры документов должны быть в диапазоне от *min* (=1000) до *max* (=10000) слов.

Принципы итеративного обучения

В основу обучения положены два принципа: типизация омонимов и разнообразие явлений омонимии, представленных в документах.

В начале обучения необходимо переранжировать базовый список омонимов, т.е. умножить ранги омонимов базового списка на *коэффициент трудности обучения*. В качестве такого коэффициента используется отношение количества ошибок выбора лемм у омонима к общему числу встречаемости омонима в предыдущих обучающих выборках. На первом этапе обучения значение коэффициента равно 1 (максимальная трудность обучения).

Затем происходит предварительный отбор нужного количества *n* (=3000–5000) документов из Web на основе максимального сходства между этими документами и новым переранжированным списком омонимов. Расчет такой меры сходства описан в предыдущем разделе.

И, наконец, из полученного списка документов на основе ГА формируется новая обучающая выборка из *k* (=100) документов с максимальным разнообразием представленных явлений омонимии, выполняется интерактивное обучение системы (создание аннотированного корпуса), и весь цикл выполняется заново до получения удовлетворительных результатов на тестовом наборе.

Экспериментальная оценка эффективности алгоритма

Для проверки эффективности работы предлагаемого алгоритма было проведено экспериментальное сравнение точности снятия омонимии данным алгоритмом и лингвистическим процессором системы ЭТАП [1]. В качестве исходных данных для обработки были выбраны 20 случайных текстов из предварительно размеченного обучающего корпуса, состоящего примерно из 500 текстовых фрагментов. Принципы формирования такого корпуса рассматривались выше. Оставшиеся фрагменты использовались для обучения алгоритма.

Таким образом, тексты, используемые в эксперименте, не были знакомы ни одной из систем. Исходные данные содержали 22548 словоформ текста, из них 3549 являлись омонимами.

Пример начальной части одного из фрагментов (в фигурных скобках указаны правильные леммы для омонимов):

"Музей? Ну вроде того{то}. Но чей - не помню (читай: не знаю). Здесь, естественно{естественно}, в море{море} ходят только в купальниках. Достаю плавки{плавки}. А Коле{коля}... а он... а ему{он}... Загорает он, короче{коротко}... в штанах... Ну забыл человек плавки{плавки}, ну и что? Прыгая по камешкам пляжа, захожу в воду{вода}. Поплавал - лепота! Вылажу{вылазить}. Отстукивая дрыжака{дрыжак}, бегу{бежать} к вещам, вытягиваю фотык, вручаю Коле{коля} - все{все} равно{равно} ничего не делает, пусть щелкнет меня пару{пара} раз."

В результате обработки были получены следующие результаты. Предлагаемый алгоритм правильно снял омонимию в 3457 случае (точность 97.42%), а лингвистический процессор системы ЭТАП - только в 3261 случаях (91.88%).

При этом ошибкой считался как случай явно неправильного выбора леммы, так и невозможность выбора леммы из нескольких вариантов.

Повторяющиеся в разных местах текста одни и те же ошибки участвовали в подсчете. Различные варианты выбора лемм, связанные с различием морфологических моделей, ошибками не считались и не учитывались.

Заключение и будущие работы

Предлагаемый алгоритм достаточно надежно решает задачу снятия морфологической омонимии, обладает высоким быстродействием. Для практических целей он может быть настроен на частичное снятие неоднозначности только в наиболее уверенно определяемых случаях, например, при помощи подбора порогового значения вероятности. По-видимому, алгоритм может быть обобщен и на задачу разрешения лексической или грамматической омонимии, например путем увеличения размеров словарей.

Приложение 1. Технические характеристики алгоритма

Реализация — 300 операторов Perl

Быстродействие — 2500 слов/сек

Словарь контекстов — 120987 записей

Словарь псевдоомонимов — 3324 записи

Словарь наиболее вероятных лемм — 4092 записи

Приложение 2. Примеры снятия омонимии

Пример 1. Д.Б. Эльконин. Психология игры.

```
   Прошли века{век=1.00|веко=0.00},
существенно{существенно:0.97|существенный:0.03} изменились
орудия и способы добывания огня и сверления дыр. {\s}Кубари и
жужжалки{жужжалка=1.00|жужжалкий=0.00} не
стоят{стоять:0.94|стоить:0.06}
уже{уже=1.00|уж=0.00|узко=0.00|узкий=0.00}
больше{больше:0.74|много:0.26|большой:0.00} в прямом отношении к
труду взрослых и к будущей трудовой деятельности ребенка. {\s}И
```

для{для=1.00|длить=0.00} ребенка они больше{больше:0.81|много:0.19|большой:0.00} не являются{являться:1.00|являть:0.00} уменьшенными дрелями и даже не изображают их{они:0.75|их:0.25}. {\s}Кубари и жужжалки{жужжалка=1.00|жужжалкий=0.00} превратились из "образных{образный:0.64|образной:0.36} игрушек" в "двигательные" или "звуковые", по терминологии Е. А. {\s}Аркина{аркин:0.76|аркина:0.24}. {\s}Однако действия с ними еще продолжают поддерживаться взрослыми, и они еще бытуют среди детей{ребенок=1.00|дитя=0.00}.

Пример 2. М.А. Булгаков. Мастер и Маргарита.

Виноват, -- мягко{мягко=1.00|мягкий=0.00} отозвался неизвестный, -- для{для=1.00|длить=0.00} того{то:0.84|тот:0.13|того:0.03}, чтобы управлять, нужно{нужно:0.96|нужный:0.04}, как-никак, иметь точный план на некоторый, хоть сколько-нибудь приличный срок. {\s}Позвольте же вас спросить, как же может управлять человек, если он не только лишен{лишенный=1.00|лишать=0.00} возможности составить какой-нибудь план хотя{хотя=1.00|хотеть=0.00} бы на смехотворно{смехотворно=1.00|смехотворный=0.00} короткий срок, ну, лет{год:0.95|лет:0.05|лета:0.00}, скажем, в тысячу, но не может ручаться даже за свой собственный завтрашний день{день=1.00|девать=0.00}? {\s}И, в самом{самый:0.74|сам:0.26} деле, -- тут неизвестный повернулся к Берлиозу, -- вообразите, что вы, например, начнете управлять, распоряжаться и другими и собою, вообще, так сказать, входить во вкус, и вдруг у вас... кхе... кхе... саркома легкого{легкое:0.64|легкий:0.36}... -- тут иностранец сладко{сладко=1.00|сладкий=0.00} усмехнулся, как будто мысль о саркоме легкого{легкое:0.52|легкий:0.48} доставила ему{он:1.00|оно:0.00} удовольствие, -- да, саркома, -- жмурясь, как кот, повторил он звучное слово, -- и вот ваше управление закончилось! {\s}А бывает и еще хуже{плохо:0.93|плохой:0.07|худо:0.00|худой:0.00}: только что человек соберется съездить в Кисловодск, -- тут иностранец прищурился на Берлиоза, -- пустяковое, казалось{казаться:0.97|казать:0.03} бы, дело{дело=1.00|девать=0.00}, но и этого{это:0.87|этот:0.13} совершить не может, потому что неизвестно{неизвестно:0.93|неизвестный:0.07} почему вдруг возьмет -- поскользнется и попадет под трамвай! {\s}

Литература

1. Л.Л. Цинман, В.Л. Сизов. Лингвистический процессор ЭТАП: дескрипторное соответствие и обработка метафор, 2000 http://www.dialog-21.ru/Archive/2000/Dialogue%202000-2/366.htm
2. И. Ножов. Синтаксический анализ, 2002 http://www.computerra.ru/offline/2002/446/18250/
3. Daniel Jurafsky, James H. Martin. Speech and Language Processing, 2000
4. Б.П. Кобрицов. Методы снятия семантической неоднозначности. НТИ, Сер.2, Вып. 3, 2004
5. Cristopher D. Manning, Hinrich Schutze. Foundation of Statistical Natural Language Processing, 1999
6. И. Сегалович, М. Маслов. Русский морфологический анализ и синтез с генерацией моделей словоизменения для незнакомых слов, 1998 http://company.yandex.ru/articles/article1.html
7. Национальный корпус русского языка http://www.ruscorpora.ru/