Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка (скрытая модель Маркова и синтаксический анализатор именных групп)

А.В. Сокирко

Интегрум-техно, Москва, sokirko@yandex.ru

С.Ю.Толдова

МГУ, toldova@pisem.net

Аннотация

В работе описывается серия экспериментов по снятию морфологической и лексической (лемматической) неоднозначности с использованием скрытых моделей Маркова. Для обучения модели используется Национальный корпус русского языка. Проводятся сравнения модели Маркова с программой, построенной на правилах, составленных вручную и с моделями, построенными на других формализмах (вероятностных или гибридных). Авторы приходят к выводу о перспективности использования скрытых моделей Маркова для разрешения морфологической неоднозначности. Вместе с тем точность снятия лексической неоднозначности, с которой работает модель Маркова, ниже, чем у других вероятностных моделей.

1. Постановка задачи

Задача снятия лексической и морфологической неоднозначности("tagging") актуальна для многих прикладных систем. За рубежом этой теме уделяется много внимания, ей посвящены сотни научных статей, изданных на протяжении нескольких десятков лет. Большинство систем анализа текста, коммерческих или академических, используют те или иные методы снятия неоднозначности. В отечественных проектах автоматической обработки текста эта проблема до сих пор не получила должной разработки. Причиной тому послужило распространенное мнение, пришедшее, по-видимому, еще из 60-х годов, что неоднозначность на одном языковом уровне должна решаться за счет следующего уровня, в частности, морфологическая неоднозначность должна решаться на синтаксисе. Согласившись, в принципе, что такой подход теоретически оправдан, заметим однако, что анализаторы уровня X обычно либо на порядок быстрее анализатора уровня X+1, либо проще его на тот же порядок. Можно провести аналогию, что решать неоднозначность одного уровня на следующем уровне все равно, что буксировать сломавшуюся машину с помощью вертолета. Эффектно - но дорогостояще. Учитывая непрактичность этого решения, за рубежом стали активно развиваться специализированные модули разрешения омонимии, которые можно разделить на:

1. Системы, построенные на правилах, составленных ручным способом.
2. Системы, построенные на вероятностных моделях и обученные на размеченных корпусах.
3. Гибридные системы, включающие как вероятностные модели, так и правила.

Конечно, системы, построенные только на большом количестве правил, начинают с некоторого уровня больше походить на анализаторы последующего уровня, т.к. для разрешения морфологической неоднозначности системе в любом случае приходится строить именные группы, которые являются уже синтаксической сущностью. Однако в отличие от анализаторов следующего уровня, системы снятия неоднозначности, построенные на правилах, обычно работают с линейной скоростью. Системы, построенные на вероятностных моделях, работают медленней, но реализовать их проще, и главное, методы, которые их усовершенствуют, часто переносимы с одного естественного языка на другой. Все вероятностные модели тренируются на размеченных

корпусах, т.е. на текстах, где словам вручную приписана интерпретация. Размер корпуса играет важную роль. Так некоторые модели могут обучаться только на небольших корпусах (полмиллиона слов или меньше), и обучение на больших корпусах не улучшает их, а иногда и вредит качеству алгоритма.

Один из вероятностных методов - метод скрытой марковской модели (HMM - Hidden Markov Model). Для английского языка, имеющего бедную морфологию, данный метод достигает достаточно высокой точности: порядка 98% (см.[12]). Простое перенесение этого метода на материал языков с более развитой морфологией дает обычно более низкие результаты, так, первые эксперименты на материале чешского языка давали точность около 95% (см.[11]). Однако низкие результаты первых экспериментов не привели чешских исследователей к выводу о том, что метод скрытых марковских моделей не применим к языкам с сильно развитой морфологией, а скорее стимулировали поиск путей усовершенствования данного метода. Так в работе [11] было показано, как для чешского языка с 95% можно дойти до 95,38%. Насколько мы понимаем, чешские исследователи не собираются останавливаться на достигнутых результатах. Чешская модель относится к гибридному типу, т.е. к типу, где совмещены две технологии - технология, основанная на правилах, и статистическая.

Основная цель данной работы заключается в том, чтобы начать эксперименты по применению скрытых марковских моделей к разрешению лексической и морфологической омонимии для русского языка. Актуальность данной работы обуславливается использованием для обучения модели Национального корпуса русского языка (см. [3]), который стал доступен в таком объеме (5 млн. слов) только в 2005 году. Мы будем сравнивать систему, построенную на правилах, с разными модификациями системы, построенной на марковской модели.

2. Морфологический анализ и наборы тегов

Используемый морфологический анализ слова в общих чертах основан на морфологическом словаре Зализняка[2], где каждой словоформе приписан некоторый набор граммем, которые являются значениями морфологических категорий (род, число, падеж и т.д.). Этот набор обычно называется морфологической интерпретацией слова. Кроме этого, в словаре каждой морфологической интерпретации приписана нормальная форма слова (лемма). Таким образом, для каждой словоформы S словарь выдает набор пар

<M,L>, где M - морфологическая интерпретация S, а L - лемма словоформы S. В заданном входном тексте, как правило, только одна морфологическая интерпретация является верной. Задачей нашего алгоритма - нахождение этой интерпретации, используя непосредственный контекст слова, или, по крайней мере, удаление некоторых неверных вариантов. Пусть Output(W) - множество пар <M,L>, которое осталось для слова W после работы алгоритма. Оценивать работу алгоритма мы будем по трем целевым параметрам:

- Уровень оставшейся неоднозначности: число элементов в Output(W) для всех слов текста, поделенное на число слов в тексте. Если алгоритм работает однозначно, то этот параметр равняется 1.
- Лексической точностью алгоритма мы называем число слов текста, для которых правильная лемма осталась в Output(W), поделенное на общее число слов в тексте.
- Точностью назовем число слов текста, для которых в Output(W) осталась правильная морфологическая интерпретация, поделенное на общее число слов в тексте.

Для работы с моделью Маркова мы будем использовать понятие тега. Тегом мы называем строковую константу, которая соответствует некоторому множеству пар <M,L>. Набором тегов мы называем множество тегов, которые полностью и однозначно покрывают все множество возможных пар <M,L>. Набором полных тегов мы называем такой набор, где каждый тег соответствует множеству пар $<M_{max},L>$, где M_{max} - одна из максимально полных морфологических интерпретаций слова, а L - лемма, у которой есть словоформа с интерпретацией M_{max}. В нашем случае набор полных тегов состоит из 900 штук.

От размера набора тегов сильно зависят скорость работы модели Маркова и ее размер. Кроме того, при увеличении набора тегов, модели требуется более чем линейное увеличение размера обучающего корпуса. Поэтому остается открытым вопрос о том, какой набор тегов нужно использовать для максимизации значений целевых параметров алгоритма. Можно ли сформулировать какие-то теоретические законы, которые определят достаточность данного набора тегов для решения данного типа неоднозначности? Наша работа не дает ответа на этот вопрос, мы лишь предлагаем результаты некоторых базовых экспериментов с разными наборами тегов.

3. Предыдущие работы

К сожалению, авторам не известны работы по применению марковских моделей к русскому языку, кроме [13] и [8]. В работе Ножова[13] была создана модель в рамках программы русификации продуктов компании Inxight. Модель обучалась на небольшом корпусе текстов (40 тыс. слов). Использовался средний набор тегов - 80 штук. Результат был достигнут довольно высокий (94,5% - это точность приписывания тега слову). Однако, к сожалению, в этой работе не указаны размеры и жанр тестируемого корпуса, кроме того, остаются неизвестными параметры модели Маркова, которые использовались в работе, поскольку эта программа является коммерческим продуктом.

В работе американских исследователей Hana и Feldman[8] был проведен эксперимент по переносу модели, обученной на чешском корпусе, на русский язык. Был использован набор полных тегов (900 штук). Полученная точность - 72,6%. Исследователи использовали лексикон (морфологический словарь), построенный на размеченном корпусе и снабженный довольно широкой функцией предсказания. Именно эта функции предсказания и была, как нам представляется, причиной столь низкого результата. Подтверждением тому служит факт, что уровень входной морфологической неоднозначности у них составляет 3,1, тогда как в наших экспериментах - только 2,1.

4. Синтаксический анализатор именных групп Synan

Чтобы оценить качество работы модели Маркова по разрешению морфологической омонимии, мы сравнивали ее с поверхностным синтаксическим анализатором Synan (см [4]). Мы учли, что модуль Synan использует те же морфологический и графематический анализаторы, которые использовались при создании размеченной части Национального корпуса. Исследователи часто пишут о невозможности сравнения разных систем разрешения неоднозначности, поскольку они используют разные по наполнению словари. В эксперименте, который описывается в этой статье, такой проблемы нет.

Модуль Synan не предназначался напрямую для разрешения омонимии. Главная его цель состояла в том, чтобы построить на предложении набор клауз (фрагментов), внутри которых нужно было выделить подлежащие со сказуемым и именные группы. Этот модуль не строил полное синтаксическое

дерево, поэтому в нем не использовалось, например, глагольное управление для разрешения морфологической неоднозначности.

В нормальном режиме модуль Synan разрешал морфологическую неоднозначность только на 30 процентов (с входного уровня неоднозначности 2,1 до выходного - 1,7). Затем мы использовали еще одну модификацию Synan, когда после основной работы модуль выбирал самую частую для данного слова морфологическую интерпретацию, а остальные удалял. Последняя модификация выдавала однозначную интерпретацию, но с очень низкой точностью - 88.80% (см. ниже таблицу 3).

5. Модель Trigram

В основу нашей модели (рабочее название “Trigram”) была положена модель, предложенная для чешского языка в работе [11] . Ниже будут кратко определены основные параметры этой модели (детальное объяснение теории моделей Маркова можно найти, например, в работе L.R.Rabiner[6]). Модель Trigram, как и работе [11], состоит из двух частей:

- Трехграммная модель для тегов: $p(t_i \mid t_{i-2}, t_{i-1})$ – вероятность того, что некоторый тег t_i может следовать во входном тексте за тегами t_{i-1} и t_{i-2}.
- Биграммная модель для словоформ: $p(w_i \mid t_i, t_{i-1})$ – вероятность того, что некоторое слово w_i может иметь тег t_i, если предыдущему слову было приписан тег t_{i-1}. Эта модель еще называется лексической вероятностью.

Для каждого входного предложения Trigram определяет наиболее вероятные теги каждого слова по следующим формулам:

$$T = \mathrm{argmax}_T \, P(W|T)P(T),$$
$$P(T) = \Pi_{i=3..n} \, p_{smooth}(t_i \mid t_{i-2}, t_{i-1}) \quad \text{и} \qquad (1)$$
$$P(W|T) = \Pi_{i=3..n} \, p_{smooth\_lex}(w_i \mid t_i, t_{i-1}).$$

Вероятности p_{smooth} строятся с помощью сглаживания (”smoothing”):

$$p_{smooth}(t_i \mid t_{i-2}, t_{i-1}) = \lambda_3 \, p(t_i \mid t_{i-2}, t_{i-1}) + \lambda_2 \, p(t_i \mid t_{i-1}) + \lambda_1 \, p(t_i), \qquad (2)$$

где $\lambda_1+\lambda_2+\lambda_3=1$. Если не использовать сглаживания, тогда в тех случаях, когда некая триграмма $<t_i, t_{i-2}, t_{i-1}>$ ни разу не встречалась в обучающем корпусе, «сырая» вероятность $p(t_i \mid t_{i-2}, t_{i-1})$ обращается в ноль, и модель не может приписать слову тег t_i, если перед ним стояли теги t_{i-2} и t_{i-1}.

Коэффициенты $\lambda_1,\lambda_2,\lambda_3$ из формулы (2) могут быть вычислены для всего множества триграмм, или же множество всех триграмм может быть поделено на группы, и для каждой из них отдельно вычисляются оптимальные значения $\lambda_1,\lambda_2,\lambda_3$. В последнем случае говорится, что применяется группировка ("bucketing") триграмм. Повышение точности алгоритма при использовании группировки объясняется тем, что одни теги зависят больше от предыдущих тегов, чем от тегов, которые стоят через один от них. В таких случаях значение λ_2 должно быть выше, чем значение λ_3. Понятно, что возможны и обратные случаи. Алгоритм деления множества тегов на группы описан в работе Chen[10]. Этот алгоритм там образно назван "построением стены из кирпича" (wall of bricks process) и базируется он на сортировке всех триграмм по частоте встречаемости с дальнейшей разделением триграмм на N групп. Внутри каждой группы вычисление оптимальных значений $\lambda_1,\lambda_2,\lambda_3$ осуществляется с помощью интерполяции удаления ("deleted interpolation" см. [16]) .

Для сглаживания лексической вероятности мы использовали схожую формулу из работы Thede&Harper [12]:

$$p_{smooth\_lex}(w_i \mid t_i, t_{i-1}) = \left(\frac{N_{\log}}{N_{\log}+1}\right)p(w_i \mid t_i, t_{i-1}) + \left(\frac{1}{N_{\log}+1}\right)p(w_i \mid t_i) \quad (3)$$

где $N_{log} = \log(N_3+1)+1$, а N_3 – количество вхождений слова w_i с тегом t_i, таких что до слова w_i стояло слово с тегом t_{i-1}.

Формула (3) применялась только в том случае, если слово с таким тегом встречалось в обучающем корпусе ($p(w_i \mid t_i) > 0$). Если же этого не произошло, тогда

$$p_{smooth\_lex}(w_i \mid t_i, t_{i-1}) = 1/M, \quad (4)$$

где M – число вхождений слова w_i с тегом t, таком что $p(w_i \mid t)$ максимально. Таким образом новой интерпретации приписывается минимально низкая вероятность (см. Jurish[14] о других возможных способах интеграции новых морфологических интерпретаций слова, не вошедших в обучающий корпус).

В последнем рабочем варианте мы умножали значение, полученное по формуле (3), на длину входного слова, что очень незначительно увеличивало точность алгоритма (чем длиннее слово, тем меньше в среднем оно зависит от общего трехграммного контекста, а больше зависит от лексической вероятности).

5. Условия эксперимента

Для обучения модели мы использовали Национальный корпус русского языка. Морфологическая разметка корпуса была переведена в стандарт aot.ru[5]. При переводе в морфологических интерпретациях исчезали граммемы, которых нет в стандарте aot.ru. В некоторых случаях из-за того, что неоднозначность в корпусе сознательно не была разрешена полностью, программе приходилось строить новую морфологическую интерпретацию, которую принципиально не может построить морфологический словарь или модуль Synan. Таким образом, наша попытка уровнять шансы модуля Synan и Trigram не увенчалась полным успехом. Однако нам представляется, что уровень несоответствия здесь не может считаться значительным.

Для тестирования были созданы три непересекающихся подкорпуса, каждый на 3300 предложений (1/100 корпуса). Каждый раз при тестировании один подкорпус исключался из обучения, и на нем проводилось тестирование. В результирующих таблицах 1-3 приведены средние значения по трем тестируемым подкорпусам.

Отдельный вопрос составляет то, какое определение слова (токена) нужно использовать для вычисления точности алгоритмов. В большинстве работ при оценке точности не учитываются знаки препинания, цифробуквенные комплексы. Это делается потому, что знаки препинания и числа почти всегда однозначны. В нашем эксперименте, кроме вышеперечисленного, не учитывались еще слова, записанные латиницей. Однако в работе[11] знаки препинания учитывались, хотя это не говорится в тексте (нам пришлось напрямую связываться с один из авторов, чтобы выяснить этот вопрос). Разница здесь может быть существенной, например, для набора полных тегов с учетом знаков препинания и чисел лучшая точность модели Trigram – 95,39%, а без учета 94,46%.

Кроме модулей Synan и Trigram, в эксперименте был использован модуль Accopost (см. [9]). Реализация модели Маркова в модуле Accopost почти полностью следует известной реализации TnT (см. [7]). Таким образом модуль Accopost является представителем классической базовой реализации моделей Маркова, которая применяется для немецкого, английского и французского языков.

Нами было использованы три набора тегов:

- Частеречный набор(Таблица 1) состоит из 19 тегов, которые соответствуют частям речи. На самом деле, с учетом того, что частеречная омонимия не везде была снята, этот набор тегов

вырастал до 150 (один тег мог состоят из двух и более частей речи, например, СОЮЗ_ЧАСТ).

- Inxight набор (Таблица 2) состоял из 91 тега (тоже вырастал до 228 тега из-за недоснятой омонимии). Этот набор тегов использовался для сравнения нашей модели с моделью компании Inxight (см. Ножов[13])
- Набор полных тегов (Таблица 3) состоял из 829 тега.

В следующих таблицах собраны данные о проведенных экспериментах:

Название модуля	Частичное снятие омонимии	Средний уровень оставшейся неоднозначности[1]	Точность	Лексическая точность
Synan	Да	1.14	99.13%	99.26%
Synan	Нет	1.00	96.87%	99.26%
Trigram	Да	1.14	99.07%	99.76%
Trigram	Да	1.08	98.67%	99.63%
Trigram	Нет	1.00	97.26%	99.17%
Accopost	Нет	1.00	96.62%	-

Таблица 1: Сравнение модуля Synan и Trigram на частеречном наборе тегов

Теггер	Точность
Inxight (согласно [13])	94,5%
Trigram	94.6%

Таблица 2: Сравнение модели HMM из работы Ножова[13] и Trigram на наборе тегов Inxight (полное разрешение неоднозначности)

Теггер	Частичное снятие омонимии	Средний уровень оставшейся неоднозначности	Точность	Лексическая точность
Trigram	Да	1.63	98.34%	99.71%
Trigram	Да	1.13	97.04%	99.42%
Trigram	Нет	1.00	94.46%	99.10%
Trigram без группировки	Нет	1.00	94.41%	99.02%

[1] Этот параметр определялся как отношение T/C, где T – число оставшихся тегов, приписанных словам, а C – число слов в тексте.

Trigram без группировки и без зависи-мости лекси-ческой вероятности от предыдущего тега	Нет	1.00	93.81%	98.96%
Synan	Да	1.69	98.65%	99.06%
Synan	Нет	1.00	88.80%	99.06%

Таблица 3: Сравнение Trigram и модуля Synan на наборе полных тегов

6. Анализ результатов: полное или частичное снятие неоднозначности

При полном снятии неоднозначности Synan явно проигрывает модулю Trigram. Для набора полных тегов (Таблица 3) эта разница составляет 94,46% – 88,80% = 5,66%. Однако, как уже говорилось, Synan не проектировался специально для этой задачи. С другой стороны, в случае частичного снятия неоднозначности модуль Trigram немного уступает модулю Synan. Для набора полных тегов эта разница составляет 98,65% – 98,34% = 0.3%.

Следовательно, система, работающая на правилах, составленных ручным способом, работает в той области, для которых были сделаны эти правила, лучше, чем вероятностная модель. Этот вывод можно подтвердить тем, что, например, лучший модуль для различения частеречной омонимии для английского языка работает на правилах (см. [1]), а не на вероятностной модели, причем этот модуль значительно опережает все чисто вероятностные модели (99,5% точности). Вместе с тем остается открытым вопрос, возможно ли построить такую же модель на правилах для набора полных тегов. Ведь полный набор тегов в 40 раз больше частеречного набора тегов. Нам представляется, что ответ на этот вопрос скорее отрицательный, чем положительный, т.е., по нашему мнению, для набора полных тегов в случае полного разрешения неоднозначности нужно использовать вероятностную модель.

7.Анализ результатов: группировка

Для набора полных тегов экспериментальным путем было получено, что число групп должно быть около 20 (для чешского языка[11] оптимальное число было 32). Однако в нашем эксперименте группировка не дала такого эффекта, как для чешского языка (улучшение с 94.97% до 95.16%, т.е. 4% относительного прироста). А в нашем эксперименте - улучшение с 94.41% до 94.46%, т.е. 0,8% относительного прироста.

Возможно, столь низкий прирост происходит из-за того, что в нашей модели отсутствует группировка для лексической вероятности (для чешского языка работали обе группировки, а в нашей модели группы лексической вероятности фактически состояли из отдельных триграмм, т.е. сколько триграмм, столько и групп). Еще более вероятным представляется утверждение, что группировка не дает большого результата, если обучающий корпус достаточно велик (у нас 5 млн. словоформ, а для чешского языка – 1,8 млн.).

8.Анализ результатов: лексическая точность

Из сравнения данных из таблиц 1 и 3 следует, что лексическая точность в частеречной модели(99,17%) выше, чем лексическая точность для набора полных тегов (99.10%), в том случае, если модели выдают однозначную интерпретацию. Оказывается, что добавление новой информации о словах (падеж, число) не улучшает распознание леммы, а даже немного ухудшает. Одно из объяснений может заключаться в так называемом проблеме разреженности (sparsedness problem), которая заключается в том, что размер обучающего корпуса должен кубически (количество разных триграмм) зависеть от размера набора тегов. Т.е. если мы считаем, что 5 млн. словоформ достаточно для частеречного набора, то для набора полных тегов нужен корпус более 1трлн. словоформ. В противном случае программа будет работать с очень низкими частотами триграмм, что приводит к невозможности статистических обобщений.

9.Анализ результатов: другие модели

Нами было проведено сравнение программы Trigram с вероятностной моделью, предложенной Ю.Г.Зеленковым и др. (см. [15]). Эта модель строит лемму омонимичного слова, учитывая окончания слов, входящих в левые и правые контексты

омонимичного слова. Модель предварительно должна быть обучена на корпусе со снятой омонимией. Оценка качества этой модели проводилась в том числе на некотором специальном корпусе (взятом из проекта ЭТАП), состоящем из 22548 словоформ, из которых 3549 являлись омонимами. Модель Зеленкова и др. правильно разрешила неоднозначность в 3457 случаях (точность - 97.42%). Программа Trigram правильно разрешила неоднозначность в 3449 случаях (точность - 97.18%). Учитывая, что модель Зеленкова и др. имеет более высокую скорость обработки входного текста, следует признать, что для русского языка при снятии неоднозначности по леммам предпочтительней использовать модель Зеленкова и др.

Кроме этого, в работе Ножова[13] приводится информация, что модель Маркова для русского языка, разработанная в компании Inxight, правильно выбирала лемму в 99% случаях (учитывая только словоформы с разными леммами). К сожалению, автор работы не приводит описание процедуры проверки точности. К тому же возможно, что в наших словарях были по-разному определены леммы для некоторых частотных омонимов. Например, нами было подсчитано, что, если не различать омонимию хотя бы для словоформ

его/их/ее: (*Я увидел его.* vs. *Его дом стоял на горе.*)
все: (*Я знаю все*. vs. *Я знаю все дома*)
это: (*Я это знаю*. vs. *Я знаю это слово*)
точность Trigram по выбору омонимичных лемм вырастет с 97,18% до 97,80%.

10. Заключение и будущие работы

В работе было показано, что предложенная модель может точнее приписывать однозначную морфологическую интерпретацию словам, чем это делает синтаксический модуль Synan.

Вместе с тем было показано, что предложенная модель хуже справляется с задачей разрешения лексической неоднозначности, чем модель Зеленкова и др.

Необходимо учитывать, что около 10% несоответствий между тегами, которые выдает программа Trigram, и тегами, которые приписаны в Национальном корпусе, являются ошибками корпуса (проценты, приведенные в таблицах 1-3, были получены автоматически). Это может означать очень многое, учитывая то, что на подобных ошибках модель тренировалась. Мы надеемся, что в будущем нам станет доступно новое издание Национального

корпуса, в котором ошибки будут исправлены, и тогда качество работы Trigram можно будет оценить снова.

Возможно еще дополнение программы Trigram специальным блоком простых контекстных правил, написанных вручную, по образцу тех, что были сделаны для чешского языка.

11. Благодарность

Работа выполнена при поддержке ООО “Яндекс” (грант N 92802 за 2005 год).

12. Литература

[1] Voutilainen A. EngCG tagger, Version 2, In Brondsted T., Lytje I. (eds.). Sprog og Multimedier.Aalborg Universitetsforlag, Aalborg (1997).

[2] Зализняк А.А."Грамматический словарь русского языка" М.: Русский язык, 1980 г.

[3] Национальный корпус русского языка, www.ruscorpora.ru

[4] Гершензон Л.М, Ножов И.М., Панкратов Д.В., Сокирко А.В., Синтаксический анализ в системе РМЛ http://www.aot.ru/docs/synan.html

[5] Сокирко А.В. Морфологические модули на сайте www.aot.ru . Диалог’2004. Верхневолжский, 2-7 июня 2004 г.

[6] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. of the IEEE, Feb. 1989

[7] Thorsten Brants, TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of ANLP-NAACL*,pages 224–231, 2000.

[8] Jiri Hana and Anna Feldman, Portable Language Technology: The case of Czech and Russian. In *Proceedings from the Midwest Computational Linguistics Colloquium, June 25-26, 2004*, Bloomington, Indiana.

[9] Ingo Schröder, A Case Study in Part-of-Speech tagging Using the ICOPOST Toolkit. *Computer Science Memo 314/02*, Department of Computer Science. University of Hamburg. Hamburg, Germany 2002.

[10] Stanley F. Chen, Building Probabilistic Models for Natural Language. PhD thesis Harvard University, 1996.

[11] Jan Hajic, Pavel Krbec, Pavel Kveton, Karel Oliva, and Vladimr Petkevic.. Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001),Toulouse, France, 2001.

[12] S. M. Thede and M. P. Harper. A Second-Order Hidden Markov Model for Part-of-Speech Tagging. In Proceedings of the 37th Annual Meeting of the ACL, 1999
[13] Ножов И.М. Морфологическая и синтаксическая обработка текста (модели и программы), диссертационная работа, 2000, Москва
[14] Bryan Jurish, A Hybrid Approach to Part-of-Speech Tagging, Final Report at Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, 2003.
[15] Зеленков Ю.Г., Сегалович И.В., Титов В.А., Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов. // Компьютерная лингвистика и интеллектуальные технологии. Труды международного семинара Диалог'2005., 2005.
[16] Cristopher D. Manning, Hinrich Schuetze. Foundation of Statistical Natural Language Processing, 1999.

Comparing a stochastic tagger based on Hidden Markov Model with a rule-based tagger for Russian

Alexey Sokirko

Integrum-Techno, Moscow, sokirko@yandex.ru

Svetlana Toldova

Moscow State University, toldova@pisem.net

Abstract

A set of experiments to resolve lexical and morphological ambiguity in Russian using Hidden Markov Model(HMM) is described. The HMM-tagger is trained by Russian National Corpus. Three different tag sets are used. The authors compare the HMM-tagger with a rule-based shallow syntax program(Synan) and also with some other taggers(stochastic or hybrid). The experiments show that that for the same amount of remaining morphological ambiguity, the error rate of the HMM-tagger is almost the same as of Synan program, but if the output morphological interpretation should be unambiguous, then the HMM-tagger is considerably better.

Nevertheless while resolving lexical ambiguity the proposed HMM-tagger yields less accurate results, than the programs which are on other stochastic models.