

5. Золотарев В.В. Параллельное кодирование в каналах СПД // Вопросы кибернетики. 1986. Вып.120. С.56–58.
6. Веб-сайт www.mtdbest.iki.rssi.ru.
7. Мессу Дж. Пороговое декодирование / Пер. с англ.; под ред. Э.Л. Блоха. М.: Мир, 1966.-208 с.
8. Золотарев В.В., Овечкин Г.В. Сложность реализации эффективных методов декодирования помехоустойчивых кодов // 6-я Межд. конф. и выст. «Цифровая обработка сигналов и ее применение». М.: 2004. Т.1. - С.220-221.
9. Форни Д. Каскадные коды // Пер. с англ.; под ред. Самойленко С.И. М.: Мир, 1970.-208 с.
10. Berrou C., Glavieux A., Thitimajshima P. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes // Proc. of the Intern. Conf. on Commun. 1993. May. P.1064–1070.
11. Press Release, AHA announces Turbo Product Code Forward Error Correction Technology. 1998.
12. Williams D. Turbo Product Code FEC Contribution // IEEE 802.16.1pc-00/35. 2000.
13. MacKay D.J.C., Neal R.M. Near Shannon limit performance of low density parity check codes // IEEE Electronics Letters. Aug. 1996. V.32. №18. P.1645–1646.
14. Richardson T., Shokrollahi M., Urbanke R. Design of capacity-approaching irregular low-density parity-check codes // IEEE Trans. Inform. Theory. 2001. V.47.-P.638–656.
15. Jin H., Khandekar A., McEliece R. Irregular repeat-accumulate codes // Proc. 2nd Int. Symp. on Turbo Codes and Related Topics. 2000, Sept.-P.1–8.

А.В. Пруцков

ОПРЕДЕЛЕНИЕ И ГЕНЕРАЦИЯ СЛОЖНЫХ ФОРМ СЛОВ ЕСТЕСТВЕННЫХ ЯЗЫКОВ ПРИ МОРФОЛОГИЧЕСКОМ АНАЛИЗЕ И СИНТЕЗЕ

За последние десятилетия вычислительная техника совершила большой скачок и заняла важное место в жизни современного общества. Одной из причин такого скачка является развитие средств взаимопонимания человека и машины, что сделало компьютерные системы более удобными для использования. Однако взаимодействие человека и машины осуществляется пока с помощью команд, подаваемых с клавиатуры или манипулятора типа «мышь» через графический интерфейс пользователя (graphic user interface, GUI). Системы общения с человеком на естественном языке позволяют ускорить и упростить работу с компьютерными системами. В настоящее время такие системы получают все большее развитие не только в компьютерной, но и в бытовой технике.

Понимание речи включает два этапа:

- 1) распознавание речи: перевод речи в текстовый вид;
- 2) понимание текста: перевод текста в смысловое содержание.

В процессе понимания текста в свою очередь можно выделить последовательность следующих этапов:

- 1) лексический – выделение в тексте слов;
- 2) морфологический – определение грамматических характеристик (падеж, число, время) выделенных слов;

- 3) синтаксический – объединение слов в группы существительного и глагола;
- 4) семантический – определение смыслового содержания выявленных групп.

Результаты работы этапов нижнего уровня являются исходными данными для следующего этапа. При синтезе речи решаются обратные задачи, т. е. по смысловому содержанию генерируется текст, а затем и речевой поток (генерация речи). Понимание и генерация текста являются задачами искусственного интеллекта.

Задачей морфологического этапа является классификация слов по частям речи в соответствии с их грамматическими характеристиками (при распознавании речи, морфологический анализ) и получение форм слов с заданными характеристиками (при синтезе речи, морфологический синтез). Морфологический анализ и синтез далее будем называть задачами определения и генерации форм слов естественных языков соответственно.

Существует несколько подходов [1-3] к решению задач генерации и определения форм слов, однако они обладают следующими недостатками:

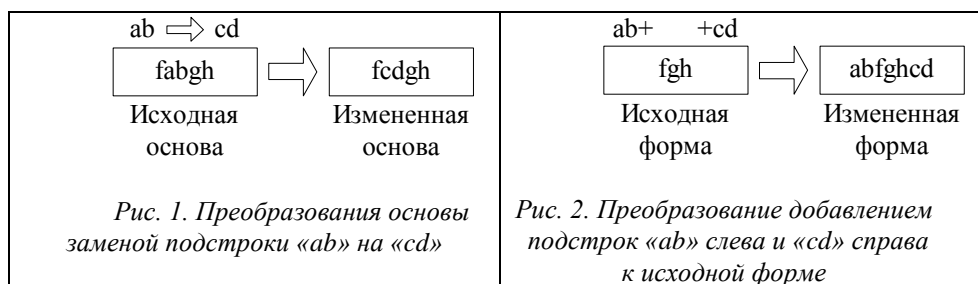
- 1) структуры данных систем и словарей – данных, входящих в их состав, различны для задач генерации и определения;
- 2) системы не применимы для языков, относящихся к различным семьям и группам языков;
- 3) сложные и запутанные схемы заполнения или модификации статей в словарях;
- 4) обрабатываются только простые формы слов;
- 5) алгоритмы обработки зависят от конкретного типа флективного преобразования (как правило, это окончание).

Предлагается система генерации и определения форм слов [4-5] (далее просто система), которая позволяет устранить перечисленные недостатки. Структура системы включает следующие словари:

- 1) словарь основ слов: содержит основы нормальных форм слов, а также грамматический тип слова;
- 2) словарь преобразований основ: содержит тип преобразования основ и его параметры;
- 3) словарь правил образования форм слов: содержит для каждого слова правила получения из нормальной формы необходимой формы слова с помощью цепочек преобразований; правила обратимы и позволяют получать из произвольной формы нормальную;
- 4) словарь слов-исключений: содержит полные формы слов, которые нерационально описывать с помощью преобразований.

В основе данной системы лежит принцип, что определение и генерацию форм слов можно представить как конечную последовательность преобразований – операций замены и добавления подстрок. Все преобразования форм слов можно свести к следующим типам:

- (A+): преобразования основы, т. е. замена одной подстроки на другую (рис. 1);
- (B+): добавление подстрок к основе справа или слева (рис. 2).



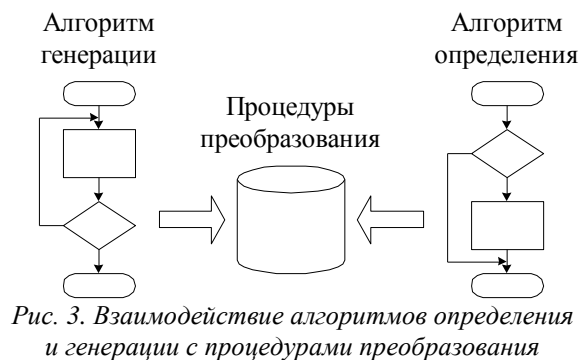
Данные преобразования являются прямыми. Для каждого преобразования существует обратное, позволяющее получить исходную форму. Для данных преобразований обратными являются соответственно:

- (A-): обратная замена подстрок;
- (B-): отделение подстроки от основы справа или слева.

Перечисленные преобразования будем называть простыми. Комбинация простых преобразований образует составное или сложное преобразование. Для каждого естественного языка определяется одно ключевое, флективное преобразование (обычно добавление окончания), которое присутствует в любой цепочке преобразований. Однако алгоритмы генерации и определения данной системы не зависят от флективного преобразования.

Алгоритм генерации [5-6] заключается в выборе цепочки прямых преобразований из словаря правил образования форм слов и применении этих преобразований к нормальной форме. В алгоритме определения происходит перебор всех цепочек обратных преобразований из словаря правил образования форм слов и поиск полученных основ в словаре основ слов.

Особенностью предлагаемой системы является то, что алгоритм не зависит от конкретного типа преобразований за счет того, что процедуры преобразования вынесены за рамки алгоритма (рис. 3).



Такой подход позволяет реализовать практически любые преобразования форм слов, специфичные для отдельных языков. Рассмотрим примеры таких преобразований для различных языков, реализованных в предлагаемой системе, но отсутствующих в аналогичных системах.

В немецком языке существуют глаголы с отделяемой приставкой, когда при спряжении приставка переносится после глагола. Рассмотрим пример спряжения такого глагола *auf|machen* – «открывать, оформлять» в настоящем времени *Präsens*:

ich mach ***auf***;
du mach ***st auf***;
er mach ***t auf***;
wir mach ***en auf***;
ihr mach ***t auf***;
sie mach ***en auf***.

Пусть необходимо получить (сгенерировать) форму третьего лица единственного числа ***macht auf*** из нормальной формы *aufmachen*. Для этого выполним следующую комбинацию простых преобразований:

- отделим окончание «-en» от нормальной формы (преобразование B-): ***aufmach***;
- добавим окончание «-t» (преобразование B+): ***aufmacht***;
- отделим приставку «auf-» (преобразование B-): ***macht***;
- присоединим частицу «auf» справа от формы (преобразование B+): ***macht auf***.

Данная комбинация образует сложное преобразование. При определении формы порядок преобразований обратный.

В испанском языке существуют местоименные или возвратные глаголы, окончание которых «-se» переносится перед глаголом и преобразуется в личное местоимение. Рассмотрим пример спряжения местоименного глагола *lavarse* – «умываться» в настоящем времени *Presente*:

yo me ***lavo***;
tu te ***lavas***;
el se ***lava***;
nosotros nos ***lavamos***;
vosotros os ***laváis***;
ellos se ***lavan***.

Получим форму первого лица единственного числа ***me lavo*** из нормальной формы *lavarse*. Чтобы сгенерировать данную форму, необходимо выполнить следующую цепочку преобразований:

- отделим окончание «-arse» от нормальной формы (преобразование B-): ***lav***;
- добавим окончание «-o» к основе (преобразование B+): ***lavo***;
- добавим частицу «me» слева от формы (преобразование B+): ***me lavo***.

Последовательность преобразований при определении формы будет обратной.

Все цепочки преобразований хранятся в словаре правил образования форм слов. Если просматривать цепочку с начала, применяя преобразования к нормальной форме, то в результате получим производную форму. Если просматривать цепочку с конца, обращая преобразования в обратные, то происходит процесс определения формы и получение нормальной формы. Цепочки преобразований можно представить в виде графа [7], что позволяет использовать аппарат теории графов и методы поиска в них.

Рассмотренные примеры получения форм слов как цепочки преобразований доказывают возможность реализации практически любых преобразований, что делает разработанную систему и предлагаемые алгоритмы универсальными по этому показателю. В настоящее время показана применимость алгоритмов

генерации и определения форм слов для языков, относящихся к различным группам и семьям языков:

- русский (восточно-славянская группа индоевропейской семьи языков);
- английский (германская группа индоевропейской семьи языков);
- немецкий (германская группа индоевропейской семьи языков);
- испанский (романская группа индоевропейской семьи языков);
- финский (прибалтийско-финская группа финно-угорской семьи языков).

Разработанная система может применяться при решении других актуальных на сегодняшний день задач, таких как:

- машинный перевод;
- поиск и индексация информации в глобальных сетях;
- проверка орфографии и грамматики;
- обучение грамматике естественных языков.

Предполагается использование разработанной системы в автоматизированных обучающих системах для генерации заданий при обучении морфологии естественных языков [8].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мальковский М.Г. Диалог с системой искусственного интеллекта. – М.: Изд-во МГУ, 1985. – 214 с.
2. Белоногов Г.Г., Богатырев В.И. Автоматизированные информационные системы. / Под ред. К.В. Тараканова. – М.: Сов. радио, 1973. – 328 с.
3. Koskeniemi, K. (1983). Two-level Morphology: A General Computational Model for Word-form Recognition and Production. University of Helsinki, Department of General Linguistics, Publications No. 11.
4. Пруцков А.В. Информационно-справочный ресурс по словообразованию естественных языков // Информационные ресурсы России. – 2004. – № 6. – С. 22-24.
5. Пруцков А.В. Морфологический анализ и синтез текстов посредством преобразований форм слов // Вестник РГРТА. Вып. 15 / Рязан. гос. радиотехн. акад. – Рязань, 2005. – С. 70-75.
6. Пруцков А.В. Алгоритмы генерации и определения форм слов // Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. А.Н. Пылькина. – Рязань: Рязан. гос. радиотехн. акад., 2005. – С. 56-61.
7. Пруцков А.В. Методы поиска решений в лингвистических автоматизированных обучающих системах // Научно-техническая информация. Серия 2. Информационные процессы и системы. – 2006. – №4. – С. 15-18.
8. Пруцков А.В. Автоматизация обучения словообразованию иностранных языков // Информатика и образование. – 2005. – № 5. – С. 117-119.

С.В. Василенко

СИНТЕЗ КВАЗИМОДАЛЬНОГО УПРАВЛЕНИЯ ДЛЯ НЕЛИНЕЙНЫХ ДИСКРЕТНЫХ СИСТЕМ

Преобразование переменных состояния часто позволяет привести уравнения системы к простым каноническим формам, что значительно упрощает решение задач анализа и синтеза [1, 2]. Широкое применение получило преобразование,