

А. М. Федотов^{1,2}, **Д. А. Тусупов**³, **М. А. Самбетбаева**³, **А. С. Еримбетова**³
А. М. Бакиева¹, **А. И. Идрисова**¹

¹ Новосибирский государственный университет
ул. Пирогова, 2, Новосибирск, 630090, Россия

² Институт вычислительных технологий СО РАН
пр. Акад. Лаврентьева, 6, Новосибирск, 630090, Россия

³ Евразийский национальный университет им. Л. Н. Гумилева
ул. К. Мунайтынова, 5, Астана, 010000, Республика Казахстан

fedotov@sbras.ru, madina_jgtu@mail.ru

МОДЕЛЬ ОПРЕДЕЛЕНИЯ НОРМАЛЬНОЙ ФОРМЫ СЛОВА ДЛЯ КАЗАХСКОГО ЯЗЫКА

Рассматриваются модели существующие алгоритмы нормализации слов естественных языков. Описаны алгоритмы автоматического выделения основ для ряда естественных языков и возможные пути синтеза нормальной формы слова для казахского языка. Разработаны правила нормализации слов для казахского языка и алгоритм для обработки как словарных, так и отсутствующих в словаре, в том числе несуществующих, слов. Создан тезаурус научно-технических терминов по информационным технологиям на казахском языке и для него реализована система нормализации, доказывающая работоспособность разработанного алгоритма.

Ключевые слова: морфология казахского языка, нормализация слов, алгоритм автоматического выделения основ, синтез нормальной формы слова.

Введение

Казахский язык – тюркский язык кыпчакской группы, который относится к типу синтетических агглютинативных языков¹, обладает богатой и сложной морфологией. Слова в нем обычно состоят из основы и добавляемых к ней аффиксов (суффикс + окончание), которых бывает, по крайней мере, два или три.

Необходимость приведения слов к нормальной форме (построения морфологического анализатора) возникла при работе с информационно-поисковыми тезаурусами с учетом морфологии казахского языка в полнотекстовых базах данных поинформационным технологиям. Приведение слов в анализируемом тексте к нормальной форме сильно упрощает работу с ним: индексацию, последующий поиск информации по построенному индексу, а также решение задач классификации (кластеризации) и автоматического реферирования документов научно-технической тематики [1].

¹ Агглютинативный язык (от лат. *agglutinatio* – приклеивание) – язык, имеющий строй, при котором доминирующим типом словоизменения является агглютинация («приклеивание») различных суффиксов или префиксов, причём каждый из них несёт только одно значение.

Данная статья посвящена анализу проблемы синтеза (поиска) основы слова казахских научно-технических терминов и построению алгоритмов ее решения. Отметим, что ориентация на тезаурус достаточно сильно упрощает рассматриваемую проблему.

Различные языки имеют разные семантические и грамматические особенности, поэтому часто алгоритмы, успешно используемые для обработки одного языка, показывают очень низкую эффективность на другом языке. Для анализа была выбрана морфологическая модель, которая анализирует наибольшее количество словоформ из нормальной формы существительного, прилагательного и нормальной формы глагола, как составляющую научно-технических терминов казахского языка.

Сложности обработки естественного языка, однако, не исключают возможности выделения более узких задач, которые уже можно решить алгоритмически: например, определение частей речи или дробление текстов на логические группы. Однако некоторые особенности естественных языков значительно снижают эффективность данных решений. Так, учет всех словоформ для каждого слова в казахском языке на порядок увеличивает сложность обработки текстов.

Обзор существующих решений

Рассмотрим методы и средства морфологического анализа в задачах нормализации слов в научно-технических терминах для казахского языка. Выделяют два принципиально разных подхода к морфологическому анализу: методы, основанные на словарях, и бессловарные морфологические анализаторы.

Остановимся на морфологических моделях, заложенных подходах к построению алгоритмов нормализации слов. Существующие подходы делятся на два класса: алгоритмы стемминга и лемматизации.

Стемминг – процесс нахождения основы слова для данного исходного слова. Основа слова не всегда совпадает с корнем.

Лемматизация – процесс привода слова (словоформы) к лемме (нормальной форме).

Дадим некоторые поясняющие определения.

Лемма – нормальная (словарная, каноническая) форма слова. К примеру, в казахском языке леммой для научно-технической терминологии являются:

- 1) существительные–именительный падеж, единственное число;
- 2) прилагательные выступают в роли определений и не приобретают окончаний, а изменение прилагательных, выступающих в роли существительных, не отличается от изменения существительных;
- 3) глаголы – начальная форма глагола.

Словоформа – это слово, представленное в определенной грамматической форме.

Лексема – слово как абстрактная единица естественного языка. В одну лексему объединяются разные парадигматические формы (словоформы) одного слова. Например, *ақпараттандыру* (сообщить) – лемма, *ақпарат тандырылған* (информационный), *ақпарат тандырылатын* (информированный) – лексема.

Морфологический анализ дает решение двух основных задач:

- задачи анализа – определение нормальной формы слова по произвольной словоформе,
- задачи синтеза – построение всех словоформ по нормальной форме.

Большинство популярных алгоритмов реализует лемматизацию (приведение к нормальной форме) с использованием основы слова (алгоритма стемминга). Однако здесь сокрыты две проблемы, характерные для казахского языка: во-первых, синтез нормальной формы сильно зависит от способа получения основы слова, а во-вторых, большинство реализаций синтезирует все возможные леммы, не выбирая из них единственного результата, либо останавливается на определении основы слова.

Проанализируем два наиболее популярных алгоритма лемматизации, основывающиеся на различных принципах: алгоритм Портера и алгоритм Яндекса.

Алгоритм Портера. Алгоритм стемминга Портера был опубликован в 1980 г. Мартином Портером для английского языка. В нем была описана последовательность шагов, в каждом

из которых при определенных правилах может происходить одно из определенных преобразований окончаний.

Это правило имеет следующую структуру [2]:

<условие> <окончание> → <новое окончание>.

Основная идея алгоритма Портера заключается в том, что существует ограниченное количество формо- и словообразующих суффиксов, и основа слова преобразуется без использования каких-либо баз (словарей) основ: только множество существующих суффиксов (при этом сложные составные суффиксы разбиваются на простые) и вручную заданные правила.

То, что алгоритм Портера не использует никаких словарей и баз основ, является плюсом для быстрой работы и спектра применения (он неплохо справляется с несуществующими словами) и одновременно минусом с точки зрения точности выделения основы. Кроме того, к минусам алгоритма Портера часто относят человеческий фактор: то, что правила для проверки задаются в ручную и иногда связаны с грамматическими особенностями языка, увеличивает вероятность ошибки [3].

В качестве примера приведем одно из правил, предложенное Портером и используемое в казахских словоформах: $(m > 0)$ ЫЛҒАН → У, которое означает, что если в словоформе есть данного рода окончание, то окончание заменяется на –у. Например, применение этого правила к словоформе «ақпарат тандырылған» приводит ее к нормальной форме «ақпарат тандыру».

Алгоритм Яндекса. (Mystem – это разработка Ильи Сегаловича (Яндекс, 1998) [4]. Данный алгоритм морфологического анализа является словарным. Основная особенность алгоритма состоит в том, что для словоформы, не описанной в словаре, или несуществующего слова, алгоритм генерирует её предположительную модель словоизменения² – один или несколько вариантов нормальной формы слова (например, стакелках {стакелк?!стакелка?!стакелок?!}), затем, пополняя словарь новыми лексемами, сгенерированные гипотетические статьи можно сохранить в этом словаре (или в другом словаре такого же типа) для дальнейшего использования [5].

Алгоритм обрабатывает анализируемый текст пословно. Каждое слово проверяется на принадлежность списку лексически самостоятельных слов, неимеющих в языке номинативной функции, пример на казахском языке: және, әрі, да, ғой, қанаи т. п. В этот список входят предлоги, частицы, некоторые междометия и наречия казахского языка из списка наиболее используемых. Кроме того, все слова, имеющие длину менее трех символов, так же не обрабатываются. Если слово является лексически самостоятельным словом, оно записывается в строку результата без изменений, алгоритм переходит к обработке следующего слова.

Следующее действие – поиск слова, которое является самостоятельной частью речи, т. е. для каждого слова запускается алгоритм выделения основы. При помощи дерева суффиксов от слова отсекается суффикс и происходит поиск предполагаемой основы в дереве основ. Если основа находится в дереве – проверяем, возможно ли сочетание данной основы и данного суффикса, удовлетворяет ли полученная модель необходимой части речи. Если да – возвращается лемма, соответствующая данной модели.

Плюсы данного алгоритма в том, что для каждого варианта нормальной формы предлагается вся грамматическая информация (синтезируемая и для несуществующих слов), эти данные можно использовать в дальнейшем для выбора одной нормальной формы из множества, предложенного программой.

Минусы данного алгоритма заключаются в том, что при отсутствии введенного слова он не всегда может справиться с поставленной задачей. Также он не справляется с уменьшительно-ласкательной формой слова. В качестве примера мы апробировали слово «машинка» – алгоритм с этой задачей не справился, тем самым пришлось добавить это слово сочетание в пользовательский словарь (-fixlist). Иначе говоря, алгоритм не всегда справляется

² Однако здесь алгоритм споткнулся на слове Скоропечатник – пишущей машине Михаила Ивановича Алисова URL: http://adm.rkursk.ru/index.php?id=13&mat_id=15963

ся с словами которые не присутствуют в словаре. Например, было проверено слово «скоропечатник».

Выбор решения

Как было указано выше, разделяют два принципиально различающихся подхода к морфологическому анализу. Первый метод, который мы рассмотрели, это алгоритм Портера, работает в бессловарном режиме. Второй метод – алгоритм от Яндекса, работающий со словарем. Наиболее полно задачам морфологического анализа решается словарными анализаторами, позволяющими определять грамматические характеристики для словоупотреблений в текстах на естественных языках, без которых невозможно, например, проводить фрагментацию текстов, выделение именных и предложных групп, однородных членов предложения. Однако этот метод с помощью словаря обладает одним главным недостатком: если анализируемой словоформы нет в словаре, то получить какую-либо морфологическую информацию о ней невозможно.

Поэтому для решения проблемы морфологического анализа при построении поискового индекса для полнотекстового поиска ставится задача разработать методы выделения основ, использующие лингвистические словари. Методы такого типа не решают полностью задачу морфологического анализа: трудно определить часть речи и грамматические признаки словоформ. Однако такие алгоритмы оказываются эффективны для задач индексации текстовых массивов, создания процедур работы со словарями научно-технических терминов для казахского языка.

Для понимания возможностей применения рассмотренных методов анализа к разным языкам необходимо рассмотреть лингвистическую классификацию казахского языка. С точки зрения типов морфологической структуры казахский язык агглютинативный (морфемы семантически отделены, но реально объединены в слова). Для агглютинативных языков (в данном случае для казахского языка) характерна достаточно развитая система словообразовательной и словоизменительной аффиксации, грамматическая однозначность аффиксов, отсутствие чередований.

Исходя из анализа существующих решений были выработаны модель и правила алгоритма получения нормальной формы слова для казахского языка, соединяющая по описаниям приведенные выше алгоритмы.

Морфологическая модель казахского языка

В казахском языке словоформы образуются путем конкатенации корня и аффиксов (суффиксов и окончаний). При этом каждый аффикс связан с наборами семантических признаков и порядок добавления аффиксов строго определен. Например, для имен существительных к основе слова сначала добавляется суффикс и далее окончание множественного числа, затем притяжательное окончание, далее следует падежное окончание и только после него – окончание формы спряжения (добавляется только к одушевленным существительным) [6].

Новые словоформы образуются с учетом морфологических и семантических признаков начальных форм следующим образом: сначала к начальной форме слова добавляются суффиксы; затем, двигаясь слева направо, определяется категория (глухие, звонкие и т. п.) последней буквы (последнего звука) начальной формы слова для добавления того или иного окончания [7].

Общая морфологическая форма определения состава выглядит так: Түбір (корень) + жұрнақ (суффикс) + жалғау (окончание) [8].

На основании анализа и грамматики казахского языка можно выделить следующие основные правила казахского языка³.

- В казахском языке слово не может оканчиваться на звонкие согласные: «б», «в», «г», «ғ», «д», «ж». В этом языке имеют место исключения, в которых удаляется суффикс,

³ Грамматика казахского языка. URL: <http://kaz-tili.kz/>

начинающийся нагласную, а стоящие в конце «б», «г», «ғ» преобразуются в следующие буквы: «п», «к», «к». Например, буква «п» на «б», «к» на «ғ», «к» на «г».

- После твердого слога следует твердое окончание, после мягкого слога – мягкое окончание.

- Мягкость и твердость слов в казахском языке определяются наличием определенной гласной в последнем слоге слова. Например, слово является твердым, если присутствуют гласные а, о, ұ, ы, я; а мягким оно становится, если присутствуют гласные ә, ө, ү, і, е. Твердость или мягкость слов коррелирует также с наличием некоторых согласных: слово твердое, если в нем присутствуют согласные қ и ғ, и мягкое, если присутствуют к и г.

- Каждое следующее окончание зависит от предыдущего по нескольким параметрам. По твердости: если последний слог слова твердый, то каждое следующее окончание будет твердым, так как твердость очередного окончания зависит от предыдущего. Таким образом, если слово твердое, то все окончания твердые, если мягкое, то мягкие.

- Как известно, морфемы являются наименьшими значащими (семантическими) единицами языка, из которых составляется словоформа, а далее, соответственно, и лексема. В казахском языке окончания делятся на четыре вида окончаний. Описанные ниже окончания непосредственно будут использоваться в разрабатываемом алгоритме определения основы слова.

Обозначим через P_i – следующие множества окончаний (аффиксов), для $i = 1, 2, 3, 4$.

P_1 – множество трех буквенных окончаний (окончание множественного числа);

P_2 – множество окончаний (притяжательные окончания);

P_3 – множество окончаний (личные окончания);

P_4 – множество окончаний (падежные окончания).

В табл. 1 описаны определения морфемного состава (P_i , где $i = 1, 2, 3, 4$):

Таблица 1

Определение морфемного состава

№	Виды окончаний	Окончания
1	Окончание множественного числа – P_1	'лар', 'лер', 'дар', 'дер', 'тар', 'тер'
2	Притяжательные окончания – P_2	'ым', 'ім', 'м', 'ың', 'ің', 'ң', 'ыңыз', 'іңіз', 'ңыз', 'ңіз', 'сы', 'сі', 'ы', 'і', 'ымыз', 'іміз', 'мыз', 'міз'
3	Личные окончания – P_3	'мын', 'мін', 'бын', 'бін', 'пын', 'пін', 'сын', 'сін', 'сыз', 'сіз', 'мыз', 'міз', 'быз', 'біз', 'пыз', 'піз', 'сындар', 'сіндер', 'сыздар', 'сіздер', 'м', 'н', 'ңыз', 'ңіз', 'к', 'к', 'ндар', 'ндер', 'ңыздар', 'ңіздер'
4	Падежное окончание – P_4	'ның', 'нің', 'дың', 'дің', 'тың', 'тің', 'ға', 'ге', 'қа', 'ке', 'ны', 'ні', 'ды', 'ді', 'ты', 'ті', 'да', 'де', 'та', 'те', 'нан', 'нен', 'тан', 'тен', 'дан', 'ден', 'мен', 'бен', 'пен'

С учетом всех комбинаций соединения окончания аффиксной группы казахского языка были определены более 750 словоизменяемых аффиксов с указанием алгоритмов синтеза слов, что предоставляет творческие возможности по расширению круга используемых слов и словосочетаний [8].

Для удобства реализации была исследована систематизация окончаний. Порядок правил имеет следующий вид:

А – окончание множественного числа + падежное окончание.

В – множественное число + личное окончание.

С – множественное число + притяжательное окончание.

Д – множественное число + притяжательное окончание + падежное окончание.

Е – множественное число + притяжательное окончание + личное окончание.

F – личное окончание + окончание множественного числа.

G – притяжательное окончание + падежное окончание.

H – притяжательное окончание + личное окончание.

Пусть

Q – произвольный одноместный предикат;

W – множество нормальной формы слова.

Каждое слово z представим в виде $z = y \wedge x$ как конкатенация двух (или более) слов y и x .

Если слово $x \in P_i$, то обозначим как $P_i(x)$ для всех $i = 1, \dots, 4$.

Если слово $x \in W$, то обозначим как $W(x)$.

Если слово $x \in Q$, то обозначим как $Q(x)$.

Тогда наши правила А–Н аналитического выделения основы по шагам удовлетворяют следующим формулам.

Пусть произвольное слово $z = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_k$, где x_i – максимальное количество букв в окончании слова z . Положим $i = k$, $x = x_i$.

Шаг 1.

$$A = \begin{cases} P_4(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_1(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \text{Положим } z = z \setminus x, i = i - 1. \end{cases}$$

Шаг 1 проверяется на применимость (условия сочетаемости), и если он не применимо, то переходим к шагу 2.

Шаг 2.

$$B = \begin{cases} P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_1(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \end{cases}$$

Шаг 3.

$$C = \begin{cases} P_4(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_1(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \end{cases}$$

Шаг 4.

$$D = \begin{cases} P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_1(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \end{cases}$$

Шаг 5.

$$E = \begin{cases} P_1(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \end{cases}$$

Шаг 6.

$$F = \begin{cases} P_4(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \end{cases}$$

Шаг 7.

$$G = \begin{cases} P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \end{cases}$$

Шаг 8.

$$H = \begin{cases} P_2(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \\ P_1(x) \rightarrow Q(z \setminus x), \text{ где } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1}. \\ \text{Положим } z = z \setminus x, i = i - 1. \end{cases}$$

Применение выбранного правила (шаг 1–8) и проверка его конечного символа, в зависимости от которого алгоритм либо останавливается, либо осуществляется переход к шагу 1.

Шаг 9.

$$\bigwedge_{i=1}^4 \neg P_i(x) \rightarrow W(z).$$

На выходе мы получаем основу анализируемой словоформы.

Приведем пример тестирования написанного алгоритма. Ниже в табл. 2 представлен образец словаря с отсеченными последних k букв для словоформ, которые составлены с помощью нашего алгоритма.

Таблица 2

Словарь научно-технических терминов

Лексема (примеры форм одного слова)		Основа слова	Лемма (нормальная форма слова)
для казахского языка	для русского языка		
Ақпарат	информация	ақпарат	информация
ақпаратпен	информацией	ақпарат	информация
Ақпаратқа	информации	ақпарат	информация
Ақпаратты	информации	ақпарат	информация
ақпараттық	информационный	ақпаратт	информационный
ақпараттандыру	информатизация	ақпараттандыру	информатизация
ақпараттандырылған	информационный	ақпараттандыру	информатизация
Жүйе	система	жүйе	система
Жүйенің	системы	жүйе	система
Жүйелік	системный	жүйелік	системный
жүйелерді	систем	жүйе	система
Жүйеде	всистеме	жүйе	система
Жүйеге	всистему	жүйе	система
жүйелерінің	систем	жүйе	система
жүйесінің	систем	жүйе	система
мәліметтер	данные	мәлімет	данные
мәліметке	сданными	мәлімет	данные
мәліметтің	данных	мәлімет	данные
Іздеуші	поисковый	іздеуші	поисковый
іздеулерден	поиска	іздеу	поиск
Іздеуге	поиска	іздеу	поиск
бағдарлама	программа	бағдарлама	программа
бағдарламалау	программирование	бағдарламалау	программирование
бағдарламаларды	программ	бағдарлама	программа
бағдарламаға	программу	бағдарлама	программа
бағдарламасы	программы	бағдарлама	программа

По табл. 2 мы выделяем два примера: это нормализация слов и приведение в неизменяемую часть слова. Первый пример «ақпараттандырылған» приобретает вид «ақпарат тандыру». Здесь программа отсекает суффикс «ыл + ған» и к основе слова добавляет суффикс «+у». Тем самым мы получаем нормальную форму слова. Второй пример в большинстве случаев это лексемы, которые используются в программе, выполняются отсечением окончаний и выделения основы слова. Например, слова «жүйелерді» приобретает вид «жүйе». Здесь отбрасывается с правой стороны падежное окончание «-ді», затем окончание множественного числа «-лер».

Практическая реализация и тестирование алгоритма

В качестве языка реализации был выбран PHP, являющийся кросс платформенным языком, подходящим для разработки встраиваемого в систему модуля Kaz Term Analyzer для обработки текста.

Разработано веб-приложение для преобразования словоформ казахского языка, работающее с полнотекстовой базой публикаций ИВТ СОРАН по информационным технологиям. Пример работы приложения показан на рисунке (для краткости приведена только часть словоформ) – выведены термины, преобразованные в исходные словоформы

Была проведена проверка правильности преобразования словоформ. За исходные данные были взяты термины из словаря (тезауруса), по которому определялась конкретная словоформа.

В итоге для 1 000 произвольно выбранных словоформ было получено 100 % правильно сгенерированных, из чего можно утверждать, что алгоритм работает корректно.

```

array(1000){[0]=>string(22)"Информатика1"[Информатика]=>string(8)"және2"·
[және]=>string(18)"Жана3"[Жана]=>string(28)"технологиялардын4"[технология]=>
string(20)"Ақпараттық5"[Ақпаратт]=>string(24)"қауіпсіздіктер6[қауіпсіздік]=>string(20)
"технологиялар7"[технология]=>string(10)"Базалық8"[Базалық]=>string(16)·
"станциялардын9"[станция]=>string(12)"Ғаламдық10"[Ғаламд]=>string(16)"желілеге"·
[желі]=>string(14)"Тезаурусының11"[Тезаурус]=>string(15)"мақалаларға12"[мақала]=>
string(8)"Роже13"[Роже]=>"Автоматтық14"[Автоматт]=>string(12)"сөйлеу15"·
[сөйлеу]=>string(17)"синтезі16"[синтезі]=>"Компьютерлер17"[Компьютер]=>string()·

```

Как видим, по экспериментам с данным алгоритмом мы получаем отдельные слова в смешанной форме (отсеченная основа и нормальная форма слова). Например, в слове «ақпараттық» при отсечении окончаний осталась основа слова «ақпаратт». А в других случаях при отсечении окончаний оставались уже нормальные формы слова. Например, «компьютерлер» → «компьютер», «тезаурусының» → «тезаурус». Однако данная проблема неоднозначности снимается наличием словаря терминов.

Заключение

Исследованы существующие алгоритмы выделения основ и приведение в нормальную форму слов для казахского языка. Разработан алгоритм для морфологического анализатора казахского языка и создан прототип системы нормализации, подтверждающий работоспособность данного алгоритма. В процессе работы над алгоритмом:

- создана база данных начальных форм слов объемом более 1 000 слов с разметкой частей речи и других признаков, необходимых для генерации словаря словоформ; получена формальная модель словоизменения и словообразования казахского языка с учетом семантики;
- автоматически сгенерирована база данных казахских словоформ объемом более 2 000 словарных статей с полной морфологической информацией.

Отличительными особенностями построенного алгоритма является его понятность и достаточно легкая воспроизводимость, что позволяет, в частности, без особых трудозатрат применить его в семантических поисковых системах.

Список литературы

1. Шокин Ю. И., Федотов А. М., Барахнин В. Б. Проблемы поиска информации. Новосибирск: Наука, 2010. 196 с.
2. Porter M. F. An algorithm for suffix stripping // Program. 1980, Т. 14, №3. Р. 130–137.
3. Willett P. The Porter stemming algorithm: then and now // Program: Electronic Library and Information Systems. 2006. В. 3, Vol. 40, С. 219–223.
4. Segalovich I. «A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine» MLMFA, 2003. Р. 273–280.
5. Сегалович И. В., Маслов М. А., Русский морфологический анализ и синтез с генерацией моделей словоизменения для неописанных в словаре слов. М.: Диалог, 1998. Т. 2. С. 547–552.

6. Казахская грамматика. Фонетика, словообразование, морфология, синтаксис. Астана, 2002.
7. Бектаев К. Большой казахско-русский, русско-казахский словарь. Алматы, 1995, 703с.
8. Шарипбаев А. А., Бекманова Г. Т., Ергеш Б. Ж., Бурибаева А. К., Карабалаева М. Х. Интеллектуальный морфологический анализатор, основанный на семантических сетях // Материалы междунар. науч.-технич. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2012). Минск: БГУИР, 2012. С. 397–400.

Материал поступил в редколлегию 25.02.2015

**A. M. Fedotov, D. A. Tussupov, M. A. Sambetbayeva
A. S. Yerimbetova, A. M. Bakieva, A. I. Idrisova**

*Institute of Computational Technologies SBRAS
6 Lavrentiev Ave., Novosibirsk, 630090, Russian federation*

*Novosibirsk State University
2, Pirogov Str., Novosibirsk, 680090, Russian Federation*

*L. N. Gumilyov Eurasian National University
5, K. Munaitpasov Str., Astana, 010000, Republic of Kazakhstan*

fedotov@sbras.ru, madina_jgtu@mail.ru

THE IMPLEMENTATION OF THE ALGORITHM GENERATING WORD FORMS OF THE KAZAKH LANGUAGE

In this work considers the methods and research of the existing algorithms of normalization of words of natural languages and its realization for the Kazakh language. Details are described algorithms automatically extract the foundations for a number of natural languages and the possible ways of determining the normal form of the word. The algorithm and the rules of normalization of words for the terms included in the thesaurus domain. Implemented a system of normalization of scientific and technical terms in the Kazakh language, proving performance of the proposed algorithm.

Keywords: the morphology of the Kazakh language, normalization of words, algorithms of allocation basics, synthesis of a normal form.

References

1. Shokin Yu. I., Fedotov A. M., Barakhnin V. B. Problemy poiska informacii [Problems of information retrieval]. Novosibirsk, Nauka, 2010, 196 p. (in Russ.)
2. Porter M. F. An algorithm for suffix stripping // *Program*. 1980, vol. 14, №3, p. 130–137.
3. Willett P. The Porter stemming algorithm: then and now // *Program: Electronic Library and Information Systems*. 2006. Bd. 3, vol. 40, p. 219–223
4. Segalovich I. V. «A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine». 2003. P. 273–280.
5. Segalovich I. V., Maslov M. A., Russian Morphological Analysis and Synthesis With Automatic Generation of Inflection Models For Unknown Words. Moscow: Dialog, 1998, vol. 2, p. 547–552 (in Russ.)
6. Kazakh grammar. Phonetics, vocabulary, morphology, syntax. Astana, Astana Polygrafia, 2002. (in Kazakh)
7. Bektaev K. Big Kazakh-Russian, Russian-Kazakh dictionary. Almaty, 1995. 703 p. (in Kazakh)
8. Sharipbaev A. A., Bekmanova G. T., Ergesh B. J., Buribaeva A. K., Karabalaeva M. H. Intellektual'nyj morfologicheskij analizator, osnovannyj na semanticheskij setyah [The intellectual morphological analyzer based on semantic networks]. *Proceedings of the International Scientific – Technical Conference «Open Semantic Technology of Intelligent Systems» (OSTIS-2012)*. Minsk, 2012, p. 397–400. (in Russ.)