

УДК 004.89:004.93

В. Ю. Шелепов, А. В. Ниценко

Государственное учреждение «Институт проблем искусственного интеллекта», г. Донецк
283048, г. Донецк, ул. Артема, 118-б

О ТЕКСТОВОЙ КОРРЕКЦИИ И ЕЕ ПРИМЕНЕНИИ ПРИ РАСПОЗНАВАНИИ РЕЧИ

V. Ju. Sheleпов, A. V. Nicenko

Public institution "Institute for Problems of Artificial Intelligence", Donetsk
283048, Donetsk, Artema st., 118-b

ABOUT THE TEXT CORRECTION AND ITS APPLICATIONS IN RECOGNITION OF SPEECH

В. Ю. Шелепов, А. В. Ниценко

Державна установа «Інститут проблем штучного інтелекту», м. Донецьк
283048, м. Донецьк, вул. Артема, 118-б

ПРО ТЕКСТОВУ КОРЕКЦІЮ І ЇЇ ЗАСТОСУВАННЯ ДО РОЗПІЗНАВАННЯ МОВИ

Основная цель настоящей статьи – механизм распознавания произносимых русских слов при сокращенном количестве базовых речевых элементов с последующим уточнением полученных обобщенных результатов в рамках работы с текстом. Аналогичный подход предлагается также при априорном распознавании первого звука. Отдельные разделы относятся к проблеме фонетических вложений и использованию глухих фрагментов при распознавании слитной речи.

Ключевые слова: текстовая коррекция, фонетические вложения, преобразование дерева эталонов, глухие фрагменты, замена заданных букв, распознавание первого звука.

The main purpose of this article - the mechanism of spoken Russian words recognition with a reduced number of basic speech elements and subsequent refinement of the obtained generalized results within the framework of working with the text. A similar approach is also proposed for a priori recognition of the first sound. Separate sections relate to the problem of phonetic nests and the use of unvoiced fragments in the recognition of speech.

Key words: text correction, phonetic nest, transformation of the patterns tree, unvoiced fragment, replacement of given letters, recognition of the first sound.

Головна мета цієї статті – механізм розпізнавання вимовлених російських слів при скороченій кількості базових мовних елементів з наступним уточненням отриманих узагальнених результатів у рамках роботи з текстом. Аналогічний підхід пропонується також під час априорного розпізнавання першого звуку. Окремі розділи відносяться також до проблеми фонетичних вкладень та застосування глухих фрагментів під час розпізнавання зливої мови.

Ключові слова: текстова корекція, фонетичні вкладення, перетворення дерева еталонів, глухі фрагменти, заміна букв, які задаються, розпізнавання першого звуку.

При распознавании речи с большими и сверхбольшими словарями чрезвычайно актуальной становится проблема скорости работы системы. Поэтому **целью данной статьи** является обоснование способа повышения скорости распознавания отдельно произносимых русских слов за счет сокращения количества базовых речевых элементов с последующим уточнением полученных обобщенных результатов в рамках работы с текстом. Предлагается также применение аналогичного подхода при априорном распознавании первого звука.

Вопросам, обсуждаемым в данной работе, посвящены статьи [1-6].

1 Система организации текстовой коррекции

Нам удобно изложить применяемую нами систему текстовой коррекции с помощью описания отдельно предлагаемой программы, которую можно использовать при ручном наборе текста. Эта программа отслеживает ошибки пользователя при наборе русскоязычного текста и позволяет быстро исправлять их, используя только клавиатуру, и не требуя отвлечения на работу с мышью. Окно программы представлено на рис. 1.

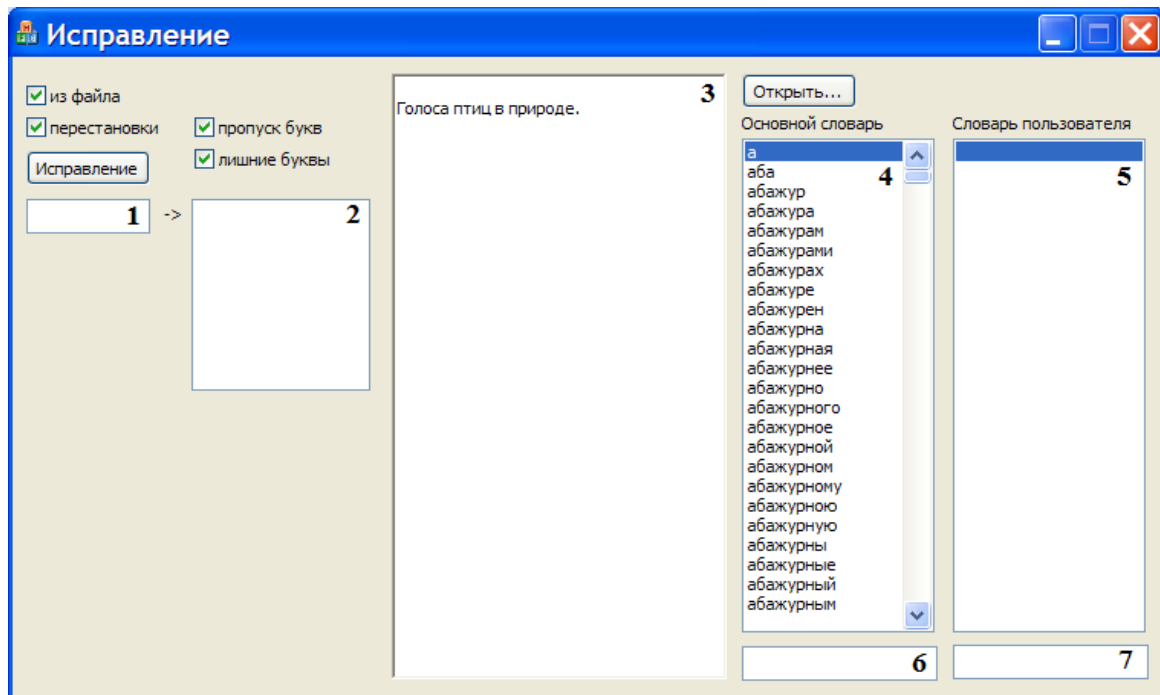


Рисунок 1 – Окно программы коррекции текста при ручном вводе

Пользователь вводит очередное слово в поле 1 и нажимает клавишу пробела. Программа проверяет наличие слова в основном словаре (список 4) и в дополнительном пользовательском словаре (список 5). (Вводя слово в поле 6 или 7, можно, при желании, вручную проверить его наличие в основном или вспомогательном словаре.) В качестве основного словаря нами используется словарь русских словоформ, содержащий около 2 миллионов единиц. Для обеспечения быстрого поиска мы представляем его в виде древовидной структуры. При этом сложность поиска оказывается пропорциональной длине слова и не зависит от размера словаря. В результате поиск слова заканчивается в момент нажатия последней нужной клавиши. Если слово обнаруживается, оно заносится в поле 3, предназначенное для набираемого текста. Содержимое поля 1 при этом очищается.

Слова может не оказаться в словарях по двум причинам:

1) это специфическое слово, например, имя собственное или специальный термин, не входящий в число общеупотребительных слов. В этом случае с помощью комбинации клавиш «Ctrl+Enter» оно заносится в дополнительный словарь пользователя, и тут же попадает в текстовое поле 3.

2) при вводе слова допущена орфографическая ошибка.

Для исправления ошибок, которое требует замены одних букв другими, используется файл «Исправление.txt», строки которого содержат буквы для взаимной замены. В обсуждаемом нами случае ввода с клавиатуры, для устранения ошибок, связанных со случайным нажатием соседних клавиш, в каждой строке указанного файла содержится буква и один из ее ближайших соседей. Это сделано вначале для верхнего, а затем – для среднего и нижнего рядов клавиатуры:

йц йф цу цы цф ук ув уы ке ка кв ен еп еа нг нр нп гш го гр шщ шл шо
щз щд щл зх зж зд хь хэ хж зь фы фя ыв ыч ыя ва вс вч ап ам ас пр пи
пм ро рт ри ол оь от лд лб ль дж дю дб жэ ж. жю э. яч чс см ми ит ть ьб
бю ю.

Если в набранной буквенной последовательности содержится, например, символ «а», то программа одну за другой образует последовательности, где вхождения этого символа заменяются ближайшими соседями «в», «к», «е», «п», «м», «с». Те последовательности, которые обнаруживаются в словарях, заносятся в промежуточный список 2. Двигаясь по этому списку с помощью клавиш «↑,↓», выбираем нужный вариант и снова нажимаем пробел. Выбранный вариант передается в текст, а список очищается. Иногда требуется принудительная очистка этого списка, это делается с помощью клавиши «Backspace».

Возможны также еще некоторые виды ошибок.

Пропуск нужных букв устраняется путем автоматического образования всех последовательностей, когда между первой парой букв поочередно вставляются все буквы русского алфавита. Затем, оставив первую пару букв прежней, переходим ко второй паре и так далее. Вставка букв производится также в начале и в конце (исключения: ь,ъ в начале слова и ь в конце слова).

Еще один вид ошибки – лишние буквы – устраняется путем последовательного исключения одного из набранных символов.

Наконец, весьма распространенной при наборе является ошибка, когда перепутан порядок двух набираемых букв. Она устраняется образованием последовательностей с перестановками соседних букв.

Те из образованных последовательностей, которые обнаруживаются в одном из словарей, заносятся в список 2. Выбор нужного варианта, сопровождаемый нажатием пробела, приводит к появлению его в тексте и сопровождается очисткой списка 2.

Таким образом, отсутствие набранной последовательности в словарях воспринимается программой как возможная ошибка. Это сопровождается специальным звуковым сигналом, который побуждает пользователя взглянуть на экран монитора, проверить содержимое списка 2, выбрать необходимый вариант и дополнительно нажать пробел. Все остальное время он может смотреть на клавиатуру, что является более естественным, чем слепой набор, когда глаза прикованы к экрану. Как известно, слепой набор становится возможным только после специального обучения.

Ввиду наличия в некоторых из описанных алгоритмов полного перебора букв, они рассчитаны на случай единственной ошибки в слове.

Дальнейшие разделы относятся к распознаванию устной русской речи.

2 О проблеме фонетических вложений

В работе [7] для распознавания слитной речи был предложен принцип минимума DTW-расстояния. Там же отмечалось, что препятствием для его эффективного применения является случай, когда словарь содержит такие слова, для которых транскрипция одного получается из транскрипции другого дописыванием одного или нескольких транскрипционных символов (фонетическое вложение). В этом случае DTW-расстояние до более короткого слова, как правило, оказывается меньше, и именно оно будет выдаваться в качестве результата при произнесении более длинного слова. Для борьбы с этим недостатком предлагается поместить более короткое слово и его продолжение в 2 разных словаря. Например, если поместить слово «балет» в 1-й словарь, а продолжение «оман» – во 2-й, то программа успешно распознает слово «балет». Она также распознает слово «балетоман» в виде последовательности слов «балет», «оман», из которых результат «балетоман» легко склеивается программно (рис. 2).

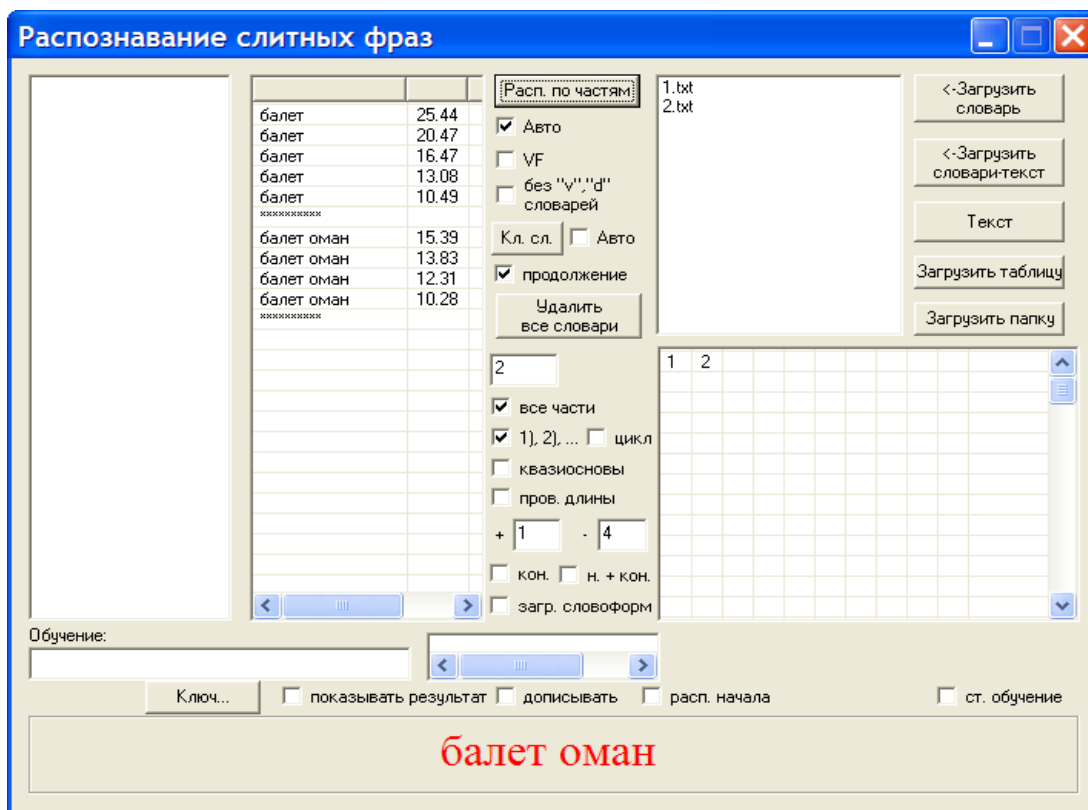


Рисунок 2 – Окно программы с результатом распознавания слов «балетоман»

3 Об использовании глухих отрезков

В работе [8] был предложен способ использования при распознавании слитной речи отрезков, отвечающих глухим звукам. Такие отрезки с высокой надежностью выделяются при сегментации непосредственно после записи речевого сигнала. Способ состоял в предварительном разбиении всего распознаваемого словаря на части, состоящие из слов с одинаковым числом глухих фрагментов и обращении к соответствующему словарю при распознавании очередного отрезка сигнала в зависимости от количества в нем глухих фрагментов. В настоящей статье предлагается заменить этот способ работой с ранее применявшейся нами VF-транскрипцией, кото-

рая отражает чередование в слове голосовых (V) и глухих (F) участков и записывается при создании дерева полных транскрипций в вершинах, соответствующих концам слов. В зависимости от чередования в рассматриваемом отрезке сигнала голосовых и глухих фрагментов результат распознавания ищется только среди слов с соответствующей VF-транскрипцией.

4 Пословное удлинение распознанного отрезка от начала слитной речи с помощью последовательной модификации дерева эталонов

Ранее применявшийся нами метод распознавания слитной речи состоял в том, что на основе принципа минимума DTW-расстояния определялось первое слово и место, где оно заканчивается в слитном речевом отрезке. Начиная с этого места, распознавалось второе слово и так далее. В настоящей работе предлагается иной способ действий. Мы как бы строим новый словарь, склеивая распознанное первое слово и все остальные слова словаря, и ведем распознавание на полученном словаре снова от самого начала слитного речевого отрезка. При этом на самом деле нам достаточно преобразовать уже существующее дерево эталонов, присоединив известный эталон распознанного первого слова к корню ранее существовавшего дерева. Это почти мгновенная операция. Затем распознавание аналогичным образом продолжается с использованием эталонов первых двух слов и так далее.

Опыт показывает, что такая организация распознавания является более эффективной, поскольку отпадает необходимость определения конца первого слова, что в общем случае затруднено возможными ошибками сегментации. К тому же отрезок двух первых слов, разумеется, длиннее, чем одно только второе слово, а DTW-алгоритм, как известно, работает тем надежнее, чем длиннее распознаваемый речевой отрезок.

5 Сокращение числа базовых речевых единиц при распознавании речи. О текстовой коррекции при голосовом вводе

Два разных слова должны различаться в написании хотя бы одной буквой. Ниже приводится алгоритм выделения групп слов, которые отличаются только заданными двумя буквами в одних и тех же позициях.

Алгоритм поиска в словаре слов, отличающихся только заданными двумя буквами.

Пусть задан массив строк (словарь) V размера n и некоторая пара букв a и b . Необходимо для каждой строки из словаря V найти строки, отличающиеся от нее только буквами a и b , то есть такие, которые можно получить из данной строки $V[i]$ всеми возможными вариантами замен a на b , либо b на a .

В общем виде алгоритм решения этой задачи выглядит следующим образом:

Шаг 1. Выбираем очередную строку $V[i]$, $i=1\dots n$.

Шаг 2. Проверяем, есть ли строка $V[i]$ в списке уже отобранных слов W . Если есть, то переход к шагу 1.

Шаг 3. Для $V[i]$ генерируем список C всех вариантов замен букв a и b , которые присутствуют в словаре V .

Шаг 4. Если в списке C только одно слово (то есть не найдено ни одного слова словаря V , отличающегося от $V[i]$ только заданными буквами), то переход к шагу 1.

Шаг 5. Добавляем все слова из C в список отобранных слов W .

Шаг 6. Если все строки в словаре исчерпаны, то переход к шагу 7, иначе переход к шагу 1.

Шаг 7. Алгоритм завершает работу.

Алгоритм получения всех вариантов замен заданных букв в слове основан на рекурсивной генерации из исходной строки всех возможных вариантов замен с последующим поиском каждого варианта в словаре. Опишем рекурсивную функцию, которая для строки S длины l генерирует все варианты строк с заменой a на b и b на a .

Функция Замена (p, S, a, b)

Шаг 1. Устанавливаем $k := p$.

Шаг 2. Если $k = l$, то переход к шагу 9.

Шаг 3. Если $S[k] = a$ или $S[k] = b$, то переход к шагу 4, иначе переход к шагу 8.

Шаг 4. $S[k] := a$. Если S есть в словаре V и отсутствует в списке C , добавляем S в C .

Шаг 5. Вызов Замена ($k+1, S, a, b$).

Шаг 6. $S[k] := b$. Если S есть в словаре V и отсутствует в списке C , добавляем S в C .

Шаг 7. Вызов Замена ($k+1, S, a, b$).

Шаг 8. $k := k+1$, переход к шагу 2.

Шаг 9. Функция завершает работу.

Вот все пары слов словаря А. А. Зализняка, которые различаются только буквами А и Э:

а, э, ай, эй, аль, эль, ан, эн, ангармонический, энгармонический, анергия, энергия, ар, эр, ара, эра, арготизм, эрготизм, ас, эс, аффект, эффект, аффективный, эффективный, афферентный, эфферентный, ах, эх, бар, бэр, дуалист, дуэлист, пар, пэр, пленарный, пленэрный, уда, удэ.

Мы имеем 19 таких пар, множество которых обозначим через (АЭ).

Для букв Б и Д подобное множество (БД) состоит из 203 пар:

абдуктор, аддуктор, абдукторный, аддукторный, .., узбечка, уздечка, хлебор\об, хлебор\од

Словарь Зализняка насчитывает около 94 тысяч слов. Таким образом, вероятность наткнуться при распознавании речи на слово, которое имеет Б-Д – двойника, приблизительно равна

$$203/94000=0,002,$$

то есть весьма мала. Для (БГ) и (ГД) эта вероятность еще меньше. Поэтому является целесообразным следующий способ действий: в словаре для распознавания заменить слова, содержащие Б и Г «квазисловами», в которых эти буквы заменены на Д. Восстановление правильного результата (или группы возможных результатов) будет осуществляться при работе уже не с речевым распознаванием, а с помощью коррекции распознанного квазислова путем обращения к словарю всех «правильных» слов. Файл «Исправление.txt» будет в данном случае содержать единственную строку:

бгд

При работе со словарем, представленным в виде дерева, исправление будет происходить очень быстро и с минимальными вычислительными затратами. Таким образом, часть работы с распознаванием переносится на работу с текстом, которая вычислительно является значительно менее трудоемкой и в результате осуществляется во много раз быстрее. Конечно, придется разбираться с двойниками, если мы наткнулись-таки на них, но вероятность этого мала.

При распознавании квазислов используется база с меньшим числом элементарных речевых единиц. Это сокращает время обучения последней и будет играть положительную роль при создании многодикторной или дикторонезависимой базы.

Все сказанное для слов с буквами Б, Г, Д целесообразно повторить для слов с буквами, заменив П, К буквой Т. В этом случае в файл «Исправление.txt» должна быть добавлена строка

кпт

Родственный прием – использование вместо слов, начинающихся на К, П или Т, квазислов, в которых эти буквы опущены. Например: «еликан» вместо «пеликан». Восстановление правильного слова достигается с помощью функции добавления пропущенных букв, использованной в текстовом корректоре, и ограниченной применением только к первой букве.

6 Применение текстовой коррекции при распознавании первого звука

В работах [9], [10] предложены способы ускорения распознавания слова сверх-большого словаря за счет предварительного распознавания его первого звука. Особую трудность представляет определение первого звука, если это один из звуков [б], [г], [д], один из звуков [м], [н], один из звуков [к], [п], [т], или мягкий вариант какого-либо из всех вышеперечисленных звуков. Как результат, предлагалось вместо определения конкретного звука ограничиваться определением принадлежности его соответствующему классу D, N или P. Теперь, в соответствии с предложенным подходом, все слова словаря, начинающиеся на Б или Г следует заменить квазисловами, начинающимися на Д, слова, начинающиеся на М, заменить квазисловами, начинающимися на Н, а в словах, начинающихся на К, П или Т, опустить эти буквы. Тогда мы уйдем от ситуации, когда вместо первого звука распознается достаточно широкий класс. После определения начала и распознавания квазислова, правильное слово можно будет восстановить с помощью вышеописанной текстовой коррекции.

Выводы

Предложен механизм текстовой коррекции при распознавании отдельно произносимых русских слов, а также подходы к проблеме фонетических вложений и использованию глухих фрагментов при распознавании слитной речи.

Список литературы

1. Kaki S. A method for correcting errors in speech recognition using the statistical features of character co-occurrence [Текст] / S. Kaki, E. Sumita, H. Iida // COLING-ACL / S. Kaki, E. Sumita, H. Iida. – Montreal, Quebec, Canada, 1998. – P. 653–657.
2. Lewis J. R. Effect of Error Correction Strategy on Speech Dictation Throughput [Текст] / J. R. Lewis // Proceedings of the Human Factors and Ergonomics Society. – 1999. – P.457–461.
3. Mangu L. Error corrective mechanisms for speech recognition / [Текст] / L. Mangu, M. Padmanabhan // Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. – Salt Lake City, UT, 2001. – Vol. 1. – P.29–32.
4. Jung S. Speech recognition error correction using maximum entropy language model [Текст] / S. Jung, M. Jeong, G. G. Lee. // Proceedings of the International Conference on Spoken Language Processing. – Jeju Island, Korea, 2004. – P.2137–2140.
5. Zhou Z. A multi-pass error detection and correction framework for Mandarin LVCSR [Текст] / Z. Zhou, H. M. Meng, W. K. Lo. // Proceedings of the International Conference on Spoken Language Processing. – Pittsburgh, PA, 2006. – P. 1646–1649.

6. Bassil Y. Post-Editing Error Correction Algorithm For Speech Recognition using Bing Spelling Suggestion [Текст] / Y. Bassil, M. Alwani. // *International Journal of Advanced Computer Science and Applications*. – 2012. – Vol. 3. – № 2. – P. 95–101.
7. Шелепов В. Ю. К проблеме распознавания слитной речи [Текст] / В. Ю. Шелепов, А. В. Ниценко // *Искусственный интеллект*. – 2012. – № 4 – С. 272–281.
8. Shelepov V. Ju. Recognition of the continuous-speech Russian phrases using their voiceless fragments [Текст] / V. Ju. Shelepov, A. V. Nicenko // *Eurasian journal of mathematical and computer applications*. – 2016. – Vol. 4. – № 4. – P. 19–24.
9. Шелепов В. Ю. О распознавании первого звука в слитном речевом отрезке [Текст] / В. Ю. Шелепов, А. В. Ниценко // *Проблемы искусственного интеллекта* – 2015. – № 0(1). – С. 116–122.
10. Shelepov V. Ju. The Refined Identification of Beginning - End of Speech; the Recognition of the Voiceless Sounds at the Beginning-End of Speech. On the Recognition of the Extra-large Vocabularies [Текст] / V. Ju. Shelepov, A. V. Nicenko // *Eurasian journal of mathematical and computer applications*. – 2017. – Vol. 5. – Iss. 4. – P. 70–79.

References

1. Kaki S., Sumita E., Iida H. A method for correcting errors in speech recognition using the statistical features of character co-occurrence. *COLING-ACL*, Montreal, Quebec, Canada, 1998, , pp. 653–657.
2. Lewis J. R. Effect of Error Correction Strategy on Speech Dictation Throughput. *Proceedings of the Human Factors and Ergonomics Society*, 1999, pp. 457-461.
3. Mangu L., Padmanabhan M. Error corrective mechanisms for speech recognition. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, 2001, vol. 1, pp. 29–32.
4. Jung S., Jeong M., Lee G. G. Speech recognition error correction using maximum entropy language model. *Proceedings of the International Conference on Spoken Language Processing*, Jeju Island, Korea, 2004, pp. 2137–2140.
5. Zhou Z., Meng H. M., Lo W. K. A multi-pass error detection and correction framework for Mandarin LVCSR. *Proceedings of the International Conference on Spoken Language Processing*, Pittsburgh, PA, 2006, pp. 1646–1649.
6. Bassil Y., Alwani M. Post-Editing Error Correction Algorithm For Speech Recognition using Bing Spelling Suggestion. *International Journal of Advanced Computer Science and Applications*, 2012, Vol. 3, no. 2., pp. 95-101.
7. Shelepov V. Ju., Nicenko A. V. K probleme raspoznavanija slitnoj rechi [To the problem of recognition of the merged speech]. *Problems of Artificial Intelligence*, 2012, no. 4, pp. 272–281.
8. Shelepov V. Ju., Nicenko A. V. Recognition of the continuous-speech Russian phrases using their voiceless fragments. *Eurasian journal of mathematical and computer applications*, 2016, Vol. 4, no. 4, pp. 19–24.
9. Shelepov V. Ju., Nicenko A. V. O raspoznavanii pervogo zvuka v slitnom rechevom otrezke [On Recognition of the First Sound in Continuous Speech Fragment]. *Problems of Artificial Intelligence*, 2015, no. 0(1), pp. 116 – 122.
10. Shelepov V. Ju., Nicenko A. V. The Refined Identification of Beginning – End of Speech; the Recognition of the Voiceless Sounds at the Beginning-End of Speech. On the Recognition of the Extra-large Vocabularies. *Eurasian journal of mathematical and computer applications*, 2017, Vol. 5, Iss. 4, pp. 70-79.

RESUME

V.Ju. Shelepov, A.V. Nicenko

About the text correction and its applications in recognition of speech

Background: currently, there is a need to build speech recognition systems that work with large dictionaries. When recognizing speech with large and ultra-large dictionaries, the problem of the speed of the system becomes extremely urgent.

Materials and methods: the article describes the method of recognition of spoken Russian words with the use of a reduced number of letters and subsequent text correction. In fact, this version of the work with preliminary recognition in the framework of the generalized transcription. Some sections relate to the problem of phonetic attachments and the use of deaf fragments in the recognition of speech.

Results: a method is proposed to increase the speed of recognition of separately spoken Russian words by reducing the number of basic speech elements, followed by refinement of the generalized results obtained in the framework of working with the text. It is also proposed to use a similar approach in a priori recognition of the first sound.

Conclusion: the developed methods and algorithms can significantly improve the speed of recognition when working with large dictionaries and can be used to create computer recognition systems of speech images.

Статья поступила в редакцию 16.10.2017.