Old or new, we repair, adjust and alter (texts)

Cvetana Krstev

University of Belgrade, Faculty of Philology

The Serbian Unitex Day
Belgrade, 11 March, 2019

Outline

Outline

1 Text mending - what can that be?

2 OCR errors

3 Diacritics restoration

4 Ekavian ⟨⟩ Ijekavian

5 Conclusion

Outline

Outline

1. Text mending - what can that be?
2. OCR errors
3. Diacritics restoration
4. Ekavian ⟺ Ijekavian
5. Conclusion

Outline

1. Text mending - what can that be?
2. OCR errors
3. Diacritics restoration
4. Ekavian () Ijekavian
5. Conclusion

Introduction to problems

When texts need mending?

Various reasons, usually because they are in some way incorrect:

- orthography;
- grammar;
- formatting;
- ...

What kind of mending are we doing?

We solve problems that occur systematically in a text, not occasionally;

and that can be solved locally.

Introduction to problems

When texts need mending?

Various reasons, usually because they are in some way incorrect:

- orthography;
- grammar;
- formatting;
- ...

What kind of mending are we doing?

We solve problems that occur systematically in a text, not occasionally;

and that can be solved locally.

Description of problems

Which problems occur systematically?

OCR errors;

Diacritics omission;

Inconsistent use of language variants

Why are these problems solved locally?

words are incorrect , and words not more complex structures are corrected.

Description of problems

Which problems occur systematically?

OCR errors;

Diacritics omission;

Inconsistent use of language variants

Why are these problems solved locally?

words are incorrect , and words not more complex structures are corrected.

Occurrence of OCR errors

The frequency of OCR errors depends on:

- used software;
- quality of the paper and print;
- language and alphabet of a text.

Characteristics of OCR errors:

- They are rarely occasional, the same type of errors tend to repeat;
- they are very local: one or two characters are confused by one or two other characters;
- characters confused can be letters, punctuation, digits.

Occurrence of OCR errors

The frequency of OCR errors depends on:

- used software;
- quality of the paper and print;
- language and alphabet of a text.

Characteristics of OCR errors:

- They are rarely occasional, the same type of errors tend to repeat;
- they are very local: one or two characters are confused by one or two other characters;
- characters confused can be letters, punctuation, digits.

OCR of Cyrillic texts

Problems

OCR cannot be successfully done on a text that uses bot Latin alphabet and Cyrillic alphabet   Cyrillic `à' is confused with a Latin `a', etc.

No solution to that problem yet.

Frequently occurring confusions

Letters `ï', `è', `í' are confused, letters `ñ' and `å', letters `ñ' and `î' (but not `å' and `î');

Two letters `ãà' and one letter `ø'; two letters `øï' and two letters `èø';

A digit `0' and a letter `Î'; a letter `È' and two digits 11 (but not vice versa); a letter `§' and a pair `ë>' (but not vice versa).

The idea of the solution

A dictionary based solution

- An OCR text is processed with dictionaries;
- All words not detected by dictionaries are marked (as potentially due to an OCR error);
- In these words one or two letters that were detected as a potential confusion are replaced with other letter(s); this process is repeated for all letters that are possible sources of confusion;
- only those corrections are accepted that represent words in (same) dictionaries; others are rejected.

Examples (1)

One error - one correction

(***êíøà)*_*êèøà*_*êøèà*_*êïøà*_*êíøà

(***êíøà)*_* êèøà *_*êøèà*_*êïøà*_*êíøà

Replacement in a text: ***êíøà) +++êèøà+++

Two errors - one correction

(***ïðåêííóòï)*_*ïðñêïèóãè*_*ïðñêïèóòè*_*ïðåêïèóãè*_...
*ïðåêèíóãè*_*ïðåêèíóòè*_*ïðñêíèóãè*...
*èðåêèèóòè*_*èðñêííóãí*_*èðñêííóòí*... (360 corrections, some duplicates)

(***ïðåêííóòï)*_*ïðñêïèóãè*_*ïðñêïèóòè*_*ïðåêïèóãè*_...
*ïðåêèíóãè*_* ïðåêèíóòè *_*ïðñêíèóãè*...
*èðåêèèóòè*_*èðñêííóãí*_*èðñêííóòí*...

Replacement in a text: ***ïðåêííóòï) +++ïðåêèíóòè+++

Examples (1)

One error - one correction

(***êíøà)*_*êèøà*_*êøèà*_*êïøà*_*êíøà

(***êíøà)*_* êèøà *_*êøèà*_*êïøà*_*êíøà

Replacement in a text: ***êíøà) +++êèøà+++

Two errors - one correction

(***ïðåêííóòï)*_*ïðñêïèóãè*_*ïðñêïèóòè*_*ïðåêïèóãè*_...
*ïðåêèíóãè*_*ïðåêèíóòè*_*ïðñêíèóãè*...
*èðåêèèóòè*_*èðñêííóãí*_*èðñêííóòí*... (360 corrections, some duplicates)

(***ïðåêííóòï)*_*ïðñêïèóãè*_*ïðñêïèóòè*_*ïðåêïèóãè*_...
*ïðåêèíóãè*_* ïðåêèíóòè *_*ïðñêíèóãè*...
*èðåêèèóòè*_*èðñêííóãí*_*èðñêííóòí*...

Replacement in a text: ***ïðåêííóòï) +++ïðåêèíóòè+++

Examples (2)

One error - two corrections

(***ïîãðåøïî)*_*íîòðñíèî*_*íîòðñøíî*_*íîãðñíèî*...
*ïîãðåíèî*_*ïîãðåøíî*_*ïîòðñøèî*_*èîòðåøèî*...
*ïîòðåøèî*_*ïîãðåøèî*_*ïîòðñíèî*_*ïîòðñøíî* (146 corrections, some duplicates)

(***ïîãðåøïî)*_*íîòðñíèî*_*íîòðñøíî*_*íîãðñíèî*...
*ïîãðåíèî*_* ïîãðåøíî *_*ïîòðñøèî*_*èîòðåøèî*...
*ïîòðåøèî*_* ïîãðåøèî *_*ïîòðñíèî*_*ïîòðñøíî*

Replacement in a text: ***ïîãðåøïî)
+++ïîãðåøíî+++ïîãðåøèî+++

Examples (3)

Errors in hyphenated words

(***‡ó-áîìíð)*_*‡óáîìïð*_*‡óáîìèð*_*‡óáîìíð

(***‡ó-áîìíð)*_*‡óáîìïð*_* ‡óáîìèð *_*‡óáîìíð

Replacement in a text: ***‡ó-áîìíð) +++‡óáîìèð+++

Errors in compounds

(***Íáðàõîïì-Õàñàíà)*_*ÍáðàõîìÕàåàïà*_*ÍáðàõîìÕàñàïà*_
*ÍáðàõèìÕàåàïà*_*ÍáðàõèìÕàñàïà*_*ÍáðàõîìÕàåàíà*...

no acceptable corrections; next try

(***Íáðàõîïì)*_*Íáðàõîì*_*Íáðàõèì*_*Íáðàõèì*_...
*Èáðàõîì*_*Èáðàõèì*_* Èáðàõèì *...-Õàñàíà

Replacement in a text: Íáðàõîïì-Õàñàíà)
+++Èáðàõèì+++-Õàñàíà

Examples (3)

Errors in hyphenated words

(***‡ó-áîìíð)*_*‡óáîìïð*_*‡óáîìèð*_*‡óáîìíð

(***‡ó-áîìíð)*_*‡óáîìïð*_* ‡óáîìèð *_*‡óáîìíð

Replacement in a text: ***‡ó-áîìíð) +++‡óáîìèð+++

Errors in compounds

(***Íáðàõïì-Õàñàíà)*_*ÍáðàõíìÕàåàïà*_*ÍáðàõíìÕàñàïà*_
*ÍáðàõèìÕàåàïà*_*ÍáðàõèìÕàñàïà*_*ÍáðàõíìÕàåàíà*...

no acceptable corrections; next try

(***Íáðàõïì)*_*Íáðàõíì*_*Íáðàõèì*_*Íáðàõèì*_...
*Èáðàõíì*_*Èáðàõèì*_* Èáðàõèì *...-Õàñàíà

Replacement in a text: Íáðàõïì-Õàñàíà)
+++Èáðàõèì+++-Õàñàíà

The System for OCR correction

The Unitex implementation

- Lexical resources for Serbian dictionaries and dict. transducers;
- Cascades of transducers:
  - The rst cascade produces corrections;
  - The second cascades lters corrections - only those representing the correct words remain.

Transducers in the rst cascade

Only a few di erent transducers are needed:

- One letter replaced by one letter and vice versa;
- One letter replaced by one letter;
- Two letters replaced by one or two letters;
- Two same letters occurring in one word replaced by the same letter;

The concrete replacements just invoke one of these transducers.

The System for OCR correction

The Unitex implementation

- Lexical resources for Serbian   dictionaries and dict. transducers;
- Cascades of transducers:
  - The rst cascade produces corrections;
  - The second cascades lters corrections - only those representing the correct words remain.

Transducers in the rst cascade

Only a few di erent transducers are needed:

- One letter replaced by one letter and vice versa;
- One letter replaced by one letter;
- Two letters replaced by one or two letters;
- Two same letters occurring in one word replaced by the same letter;

The concrete replacements just invoke one of these transducers.

One correction transducer

Two letters into one or two letters (no vice versa)

Letters øí to íè

The use of the system

Project 100 novels 1840 1920

- The system is used for the correction of the OCR of old books (published 100 years ago or before) in Cyrillic;
- The success of the process depends on:
  - The coverage of the text by lexical resources;
  - the type of OCR errors the smaller number of types of errors give better results;
- the system can reduce the number of unrecognized words up to 12% of the initial number.

Diacritics in Serbian (using Latin script)

Diacritics and their omission; Digraphs and their replacements

- c stands for c and c (and c);
- z stands for z (and z);
- s stands for s (and s).
- dj stands for đ (and dj);
- dz stands for dz (and dz).

Comparison to OCR

- Repertoire of `errors' and their possible corrections is limited and known in advance;
  - Candidates are not o ered randomly, but only those that are represented in dictionaries;
- Similar to OCR each letter without diacritic or digraph can also be correct.

Diacritics in Serbian (using Latin script)

Diacritics and their omission; Digraphs and their replacements

- c stands for c and c (and c);
- z stands for z (and z);
- s stands for s (and s).
- dj stands for đ (and dj);
- dz stands for dž (and dz).

Comparison to OCR

- Repertoire of `errors' and their possible corrections is limited and known in advance;
  - Candidates are not o ered randomly, but only those that are represented in dictionaries;
- Similar to OCR each letter without diacritic or digraph can also be correct.

Problem de nition

Tokenization

We are interested only in tokens that are simple- or multi-word units that we separate in three groups:

- Type W_a words: they will not be taken into account because they do contain neither lettersc, z, s nor clusterdj. For instance,majka `mother';
- Type W_b words: they will be processed because they do contain either lettersc, z, or s or clusterdj;
  - Type W_b words are all those containing the critical letters and/or clusters, regardless whether a diacritic sign or a digraph is really missing (npr. zvono `bell' correct and zvaka `chewing gum' incorrect zvaka);
- Type W_c words: words containing diacritics, đ, c , c , z our procedure supposes that there are no such words.

Ideas for problem solving (1)

1. Retrieving the candidates

For each W_b word in a text our procedure will o er the list of all possible candidates:

- This list may contain the original word liscem) liscem (lisce `foliage'), liscem (lisce `face'), liscem (lisac `male fox');
- It need not contain the original word lucice) lucice (lucica `port'), lucce (luciti `to separate');
- If the list contains only the original word, and no other candidates, it will be accepted immediately.

2. Candidate ranking

For each W_b word, all its candidates should be ranked according to the probability of their occurrence: for instance, in the Corpus of Contemporary Serbian (SrpKor) the candidates for liscem occur with following frequencies: liscem ! 265, liscem ! 10, liscem ! 2.

Ideas for problem solving (1)

1. Retrieving the candidates

For each W_b word in a text our procedure will o er the list of all possible candidates:

- This list may contain the original word (liscem) liscem (lisce `foliage'), liscem (lisce `face'), liscem (lisac `male fox');
- It need not contain the original word (lucice) lucice (lucica `port'), lucice (luciti `to separate');
- If the list contains only the original word, and no other candidates, it will be accepted immediately.

2. Candidate ranking

For each W_b word, all its candidates should be ranked according to the probability of their occurrence: for instance, in the Corpus of Contemporary Serbian (SrpKor) the candidates for liscem occur with following frequencies: liscem ! 265, liscem ! 10, liscem ! 2.

Ideas for problem solving (2)

3. Selection of one candidate

For all W_b words for which more than one candidate exists, our procedure chooses one with the help of:

- dictionaries,
- heuristics,
- rules (in the form of shallow parsing).

4. W_b words with no candidates

- Words unrecognized by the system missing from the dictionary, proper names, regular derivation, errors;
- Presently, we do not tackle this problem.

Ideas for problem solving (2)

3. Selection of one candidate

For all W_b words for which more than one candidate exists, our procedure chooses one with the help of:

- dictionaries,
- heuristics,
- rules (in the form of shallow parsing).

4. W_b words with no candidates

- Words unrecognized by the system missing from the dictionary, proper names, regular derivation, errors;
- Presently, we do not tackle this problem.

The dictionary for diacritic restoration (1)

The transformation of Serbian Morphological Dictionary (SMD) into the new form (SMD_DR)

- All word forms of type W_c (with diacritics) and W_b (containing critical letters and/or digraphs) are extracted from SMD;
- All type W_c words are transformed into type W_b words (diacritic deletion);
- All unnecessary information is deleted (lemma, POS, etc.);
- All obtained forms are merged.

An example

liscem,lisac.N+Zool:ms6v liscem,.X+CR=liscem
liscem,lisce.N+Conc:ns6q liscem,.X+CR=liscem ⟹ SMD_DR recnik
liscem,lisce.N+Dem:ns6q liscem,.X+CR=liscem
liscem,.X+CR=liscem_liscem_liscem

The dictionary for diacritic restoration (1)

The transformation of Serbian Morphological Dictionary (SMD) into the new form (SMD_DR)

- All word forms of type W_c (with diacritics) and W_b (containing critical letters and/or digraphs) are extracted from SMD;
- All type W_c words are transformed into type W_b words (diacritic deletion);
- All unnecessary information is deleted (lemma, POS, etc.);
- All obtained forms are merged.

An example

liscem,lisac.N+Zool:ms6v	liscem,.X+CR=liscem		
liscem,lisce.N+Conc:ns6q	liscem,.X+CR=liscem	)	SMD_DR recnik
liscem,lisce.N+Dem:ns6q	liscem,.X+CR=liscem		

liscem,.X+CR=liscem_liscem_liscem

The dictionary for diacritic restoration (2)

The size of the dictionary for diacritic restoration

SMD_DR has 943.804 entries:

- one candidate 897,077 (95.05%);
- two candidates 41,444(4.38%);
- three candidates 3,585 (0.38%);
- more than 3 candidates 569 (0.06%).

The longest list has 8 candidates

SMD_DR entry with the longest list of candidates:

Celice,.N+CR= Celice_Celice_ Celice_ Celice_
celice_celice_celice_celice

Forms of di erent surnames Celik, Celic, Celic, Celic, nouns celik , celica, celica and verbs celiciti , celiti.

The dictionary for diacritic restoration (2)

The size of the dictionary for diacritic restoration

SMD_DR has 943.804 entries:

- one candidate 897,077 (95.05%);
- two candidates 41,444(4.38%);
- three candidates 3,585 (0.38%);
- more than 3 candidates 569 (0.06%).

The longest list has 8 candidates

SMD_DR entry with the longest list of candidates:

Celice,.N+CR= Celice_Celice_ Celice_ Celice_
celice_celice_celice_celice

Forms of di erent surnames Celik, Celic, Celic, Celic, nouns celik, celica, celica and verbs celiciti , celiti.

The dictionary for candidate ranking

Information about frequencies

- This information was calculated on the bases of an excerpt of the SrpKor containing approx. 108 million words;
- the total number of word tokens was calculated ($totalNumTokens$);
  - for tokens written in upper case only frequencies of tokens written in the same way were calculated;
  - for token written in lower case the total frequencies were calculated.

Relative frequencies

$$relFreq = Round\left(\frac{freq \quad 10000000}{totalNumTokens + 0.5}; 0\right)$$

Relative frequency of 0 is 0, for 1 10 is 1, for 11 21 is 2, for 22 32 is 3,...
Relative frequencies were calculated in order to facilitate calculations.
For MWUs frequencies were not calculated.

The dictionary for candidate ranking

Information about frequencies

- This information was calculated on the bases of an excerpt of the SrpKor containing approx. 108 million words;
- the total number of word tokens was calculated ($totalNumTokens$);
  - for tokens written in upper case only frequencies of tokens written in the same way were calculated;
  - for token written in lower case the total frequencies were calculated.

Relative frequencies

$$relFreq = Round\left(\frac{freq \quad 10000000}{totalNumTokens + 0{:}5}; 0\right)$$

Relative frequency of 0 is 0, for 1 10 is 1, for 11 21 is 2, for 22 32 is 3,...
Relative frequencies were calculated in order to facilitate calculations.
For MWUs frequencies were not calculated.

Ambiguity of word forms without diacritics

The most di cult cases

- 147 entries with two candidates, both having relFreq 100;
- 12 entries with two candidates, both having relFreq 1000;
- 4 entries with three candidates, all of them having relFreq 100.

Examples

- reci,.X+CR=reci(237)_reci(2607)_reci(1448)
- sto,.X+CR=sto(36850)_sto(1268)
- nas,.X+CR=nas(5623)_nas(3528)

Ambiguity of word forms without diacritics

The most di cult cases

- 147 entries with two candidates, both having relFreq 100;
- 12 entries with two candidates, both having relFreq 1000;
- 4 entries with three candidates, all of them having relFreq 100.

Examples

- reci,.X+CR=reci(237)_reci(2607)_reci(1448)
- sto,.X+CR=sto(36850)_sto(1268)
- nas,.X+CR=nas(5623)_nas(3528)

The procedure for diacritic restoration

Text processing - using Unitex and SMD

- tokenization;
- assignment of sets of grammatical information to each word form (lemma, POS, etc.)

A dictionary information assigned to each word form

- empty word form is not in the dictionary;
- it can have one to several members, as in the case of:
  - reci,reka.N:fs7q:fs3q `river';
  - reci,redak.N:mp5q:mp1q `text line';
  - reci,reci.V:Yys `to say';
  - (and still missing the right one, due to omitted diacritics).

The procedure for diacritic restoration

Text processing - using Unitex and SMD

- tokenization;
- assignment of sets of grammatical information to each word form (lemma, POS, etc.)

A dictionary information assigned to each word form

- empty word form is not in the dictionary;
- it can have one to several members, as in the case of:
  - reci,reka.N:fs7q:fs3q `river';
  - reci,redak.N:mp5q:mp1q `text line';
  - reci,reci.V:Yys `to say';
  - (and still missing the right one, due to omitted diacritics).

The candidate assignment

Used resources and tools

- Unitex and SMD_DR (dictionaries for diacritic restoration).
- The list of candidates is retrieved from SMD_DR for each W_b word form and assigned to it.

How it works?

- Let reci be a W_b word form found in a text that is being processed;
- It will obtain from SMD various dictionary interpretations;
- In addition to that, W_b word form reci is looked for and found in SMD_DR and the value of the attribute +CR is assigned to it, in this case reci(237)_reci(2607)_reci(1448) .

The candidate assignment

Used resources and tools

- Unitex and SMD_DR (dictionaries for diacritic restoration).
- The list of candidates is retrieved from SMD_DR for each W_b word form and assigned to it.

How it works?

- Let reci be a W_b word form found in a text that is being processed;
- It will obtain from SMD various dictionary interpretations;
- In addition to that, W_b word form reci is looked for and found in SMD_DR and the value of the attribute +CR is assigned to it, in this case reci(237)_reci(2607)_reci(1448) .

Two cascades of Finite-State Transducers

The rst cascade

- Works on a tokenized text with lexical information assigned to each token word;
- Extracts information about candidates for correction from SMD_DR for each W_b word form and embeds it in a processed text;
- Solves some straightforward cases (cases in which incorrect replacement is not possible).

The second cascade

- Works on a tokenized text with lexical information assigned to each token word and list of candidates embedded in it;
- Some candidates are accepted, others are refused with the aim to reduce the number of candidates for each W_b word form to one;
- Some steps in this can be omitted or the order of step can be changed which depends on a text or user's preferences.

Two cascades of Finite-State Transducers

The rst cascade

- Works on a tokenized text with lexical information assigned to each token word;
- Extracts information about candidates for correction from SMD_DR for each W_b word form and embeds it in a processed text;
- Solves some straightforward cases (cases in which incorrect replacement is not possible).

The second cascade

- Works on a tokenized text with lexical information assigned to each token word and list of candidates embedded in it;
- Some candidates are accepted, others are refused with the aim to reduce the number of candidates for each W_b word form to one;
- Some steps in this can be omitted or the order of step can be changed which depends on a text or user's preferences.

Steps of the first cascades

An Example: input and output of the first cascade

input text	output text
Jer je imao dovoljno vremena da spreci zlocin cak i nakon reci kojima je podstrekivao sina.	Jer je imao dovoljno vremena da 5a_(spreci(302)_spreci(0)) 5a_(zlocin(456)) 3_(cak i) 1_(nakon reci) kojima je podstrekivao 5b_(sina(518)_sina(54)).
U novinama vise nije bilo ni reci o ratnoj steti.	U novinama 5b_(vise(35)_vise(17628)) nije bilo 2_(ni reci o) 4_(ratnoj steti(0)).

Steps of the second cascades

An Example: input and output of the second cascade

intermediate text	output text
Jer je imao dovoljno vremena da 5a_(spreci(302)_spreci(0)) 5a_(zlocin(456)) 3_(cak i) 1_(nakon reci) kojima je podstrekivao 5b_(sina(518)_sina(54)).	Jer je imao dovoljno vremena da spreci zlocin cak i nakon reci kojima je podstrekivao (sina(518)_sina(54)).
U novinama 5b_(vise(35)_vise(17628)) nije bilo 2_(ni reci o) 4_(ratnoj steti(0)).	U novinama vise nije bilo ni reci o ratnoj steti.

Evaluation

Procedures

- 65 di erent (and new) texts of di erent sizes were used;
- Diacritics were automatically removed;
- Diacritics were automatically restored using our procedure;
- Input and output text were compared word by word.

Results - calculations based only on W_b word forms

		P	R	Acc	F_1
tokens	average	0.986	0.939	0.969	0.962
	maximal	0.997	0.961	0.981	0.977
	minimal	0.916	0.895	0.944	0.930
types	average	0.989	0.949	0.978	0.968
	maximal	1.000	0.984	0.994	0.992
	minimal	0.958	0.885	0.950	0.929

Evaluation

Procedures

- 65 di erent (and new) texts of di erent sizes were used;
- Diacritics were automatically removed;
- Diacritics were automatically restored using our procedure;
- Input and output text were compared word by word.

Results - calculations based only on M_b word forms

		P	R	Acc	F_1
tokens	average	0.986	0.939	0.969	0.962
	maximal	0.997	0.961	0.981	0.977
	minimal	0.916	0.895	0.944	0.930
types	average	0.989	0.949	0.978	0.968
	maximal	1.000	0.984	0.994	0.992
	minimal	0.958	0.885	0.950	0.929

Graphical representation of results

Assessment of our system for diacritics restoration

Positive aspects

- Transparency every o ered solution has logical explanation which facilitates corrections;
- The improvement of lexical resources will improve system performances (e.g. longer and more reliable lists of bigrams and trigrams, more comprehensive dictionary of MWEs);
- Modularity - some steps can be omitted, new steps can be added , steps can be re-ordered.

Negative aspects

- Development time is substantial certainly more than if machine learning were used;
- Execution time is substantial making the solution unsuitable for interactive applications (for instance, mobile).

Assessment of our system for diacritics restoration

Positive aspects

- Transparency every o ered solution has logical explanation which facilitates corrections;
- The improvement of lexical resources will improve system performances (e.g. longer and more reliable lists of bigrams and trigrams, more comprehensive dictionary of MWEs);
- Modularity - some steps can be omitted, new steps can be added , steps can be re-ordered.

Negative aspects

- Development time is substantial certainly more than if machine learning were used;
- Execution time is substantial making the solution unsuitable for interactive applications (for instance, mobile).

Necessary improvements

At least

- Unrecognized word forms - solution needed;
- Better solution for cases with several candidates with balanced frequencies;
  - Hybrid approach;
- Solution for text with some, but not all diacritics missing:
  - Enhancement of the dictionary for diacritic restoration SMD_DR.

Ekavian and Ijekavian variants of Serbian

Brie y about Ekavian and Ijekavian variants

In Serbian, two standard variants of pronunciation are in use, Ekavian and Ijekavian.

They di er in the re ection of the old Proto-Slavic phoneme jat (t):

- in the Ekavian variant it is replaced predominantly by e;
- in the Ijekavian variant its is replaced by syllables ije/je , sometimes i.

A Serbian text is usually written in one of these variants.

De nition of a problem

In an Ekavian text, for each word containing e it should be decided whether it has an Ijekavian variant containing ije/je/i at that place;

In an Ijkavian text, for each word containing ije/je/i it should be decided whether it has an Ekavian variant containing e at that place;

Ekavian and Ijekavian variants of Serbian

Brie y about Ekavian and Ijekavian variants

- In Serbian, two standard variants of pronunciation are in use, Ekavian and Ijekavian.
- They di er in the re ection of the old Proto-Slavic phoneme jat ( ):
  - in the Ekavian variant it is replaced predominantly by e;
  - in the Ijekavian variant its is replaced by syllables ije/je , sometimes i.
- A Serbian text is usually written in one of these variants.

De nition of a problem

- In an Ekavian text, for each word containing e it should be decided whether it has an Ijekavian variant containing ije/je/i at that place;
- In an Ijkavian text, for each word containing ije/je/i it should be decided whether it has an Ekavian variant containing e at that place;

Description of the system

Comparison with OCR and diacritics restoration

- Like DR `errors' are limited to a small number of letters and/or syllables) a dictionary solution is appropriate;
- Similar to OCR and DR are in an Ekavian text can be a re ection of jat (reka () rijeka) or not (zeka); a syllables ije/je in an Ijekavian text can be a re ection of jat (snijeg () sneg and mjesec () mesec) or not (sujeta and prijem)

Two systems

- One system transforms an Ekavian text to Ijekavian variant, the other transforms an Ijkavian text to Ekavian variant;
- Systems work in the similar way as the system for DR, they only use two separate dictionaries;
- Frequencies incorporated in them for the selection of candidates are obtained from two di erent corpora Ekavian and Ijekavian;

Description of the system

Comparison with OCR and diacritics restoration

- Like DR `errors' are limited to a small number of letters and/or syllables) a dictionary solution is appropriate;
- Similar to OCR and DR an e in an Ekavian text can be a re ection of jat (reka ⇔ rijeka) or not (zeka); a syllables ije/je in an Ijekavian text can be a re ection of jat (snijeg ⇔ sneg and mjesec ⇔ mesec) or not (sujeta and prijem)

Two systems

- One system transforms an Ekavian text to Ijekavian variant, the other transforms an Ijkavian text to Ekavian variant;
- Systems work in the similar way as the system for DR, they only use two separate dictionaries;
- Frequencies incorporated in them for the selection of candidates are obtained from two di erent corpora Ekavian and Ijekavian;

Dictionaries for variant transformation

Variants in the standard SMD

Variants are not connected but they are marked:

Ekavian	Ijekavian	translation
reka,N612+Ek	rijeka,N612+Ijk	`river'
zeka,N741+Zool		`rabbit'
sneg,N291+Ek	snijeg,N291+Ijk	`snow'
prijem,N1		`reception/receipt'
mesec,N9+Ek	mjesec,N9+Ijk	`moon/month'
sujeta,N600		`vanity'

Production of dictionaries

Step 1

- Entries with +Ijk marker (for Ijekavian) are chosen from DELAF dictionaries, eg.rijeka ;
  - Because syllables ije/je are more speci c than e;
- One (or occasionally more) syllables/letters ije/je/i are detected and replaced by e (in the case of more possibilities more candidates are produced);
- If a produced candidate exist in DELAF and has the marker +Ek, entries are connected, eg. reka

Step 2

- Frequencies of Ekavian forms are calculated from the Serbian Ekavian corpus;
- Frequencies of Ijekavian forms are calculated from the Serbian Ijekavian corpus;

Production of dictionaries

Step 1

- Entries with +Ijk marker (for Ijekavian) are chosen from DELAF dictionaries, eg.rijeka ;
    - Because syllablesije/je are more speci c thane;
- One (or occasionally more) syllables/lettersije/je/i are detected and replaced bye (in the case of more possibilities more candidates are produced);
- If a produced candidate exist in DELAF and has the marker +Ek, entries are connected, eg.reka

Step 2

- Frequencies of Ekavian forms are calculated from the Serbian Ekavian corpus;
- Frequencies of Ijekavian forms are calculated from the Serbian Ijekavian corpus;

Entries in dictionaries for variant transformation

Words that are same in both variants are not represented in these dictionaries.

Ijk2Ek	Ek2Ijk
rijeka,.N+EK=reka(865)	reka,.N+IJK=rijeka(235)
rijekama,.N+EK=rekama(121)	rekama,.N+IJK=rijekama(34)
snijeg,.N+EK=sneg(473)	sneg,.N+IJK=snijeg(129)
snijega,.N+EK=snega(298)	snega,.N+IJK=snijega(97)
mjesec,.N+EK=mesec(2282)	mesec,.N+IJK=mjesec(644)
mjeseca,.N+EK=meseca(3922)	meseca,.N+IJK=mjeseca(3955)

Di cult cases (1)

Multiple possibilities for transformation

As with DR, in the case of multiple corrections, they are merged in one entry.

From Ijekavian to Ekavian

An Ijekavian form is a homograh of a form that does not contain Ijekavianije/je :

- njega,njega.N+Ijk+EK=nega(56)_njega(5459) (njega/nega `nursing' vs. njega `him');
- bolje,boljeti.V+Ijk+EK=bole(42)_bolje(4279) (bolje/bole `(they) hurt' vs. bolje `better').

Incorrect connections:

- bijelje,bijel.A+Ijk+EK=belje(8)_bele(1188) ; because of two occurrences ofije/je , only the rst one is correct.

Di cult cases (1)

Multiple possibilities for transformation

As with DR, in the case of multiple corrections, they are merged in one entry.

From Ijekavian to Ekavian

An Ijekavian form is a homograh of a form that does not contain Ijekavianije/je :

njega,njega.N+Ijk+EK=nega(56)_njega(5459) (njega/nega `nursing' vs. njega `him');
bolje,boljeti.V+Ijk+EK=bole(42)_bolje(4279) (bolje/bole `(they) hurt' vs. bolje `better').

Incorrect connections:

bijelje,bijel.A+Ijk+EK=belje(8)_bele(1188) ; because of two occurrences ofije/je , only the rst one is correct.

Di cult cases (2)

From Ekavian to Ijekavian

An Ekavian form is a homograh of a form that does not contain Ekavianije/je :

beg,.N+IJK=bijeg(78)_beg(15) (beg/bijeg `runaway' vs. beg `bey');

An Ekavian form has two or more di erent Ijekavian forms:

cedila,.N+IJK=cjedila(0)_cijedila(5) (cjedila `strainer (genitive)' vs. cijedila `to strain');
posede,.N+IJK=posjede(10)_posijede(0)_posijedje(0) _posjedje(0) (forms of a noun posjed and tree di erent verbs: posijedjeti, posjedjeti, posjesti).

A combination of both:

bega,.N+IJK=bijega(42)_bega(9)_bjega(0) (bega/bijega `runaway (genitive)' vs. bega `bey (genitive)') vs. bjega `to runaway (aorist)'.

What has been achieved?

OCR correction

- The system is operational;
- improvements have to be done to speed up the process etc.;
- The web application is pending (Ranka & co.);

Diacritics restoration

- The system is operational;
- The web application is being done but ... (Ranka & co.);

Ekavian ⟨⟩ Ijekavian

- The system is not operational yet;
- Improvements are necessary: correction of Ijk part of dictionaries, enlargement of Ijk corpus, evaluation (Ijk expert needed);
- The web application same as for DR (Ranka & co.).

What has been achieved?

OCR correction

The system is operational;

improvements have to be done to speed up the process etc.;

The web application is pending (Ranka & co.);

Diacritics restoration

The system is operational;

The web application is being done but ... (Ranka & co.);

Ekavian $\Longleftrightarrow$ Ijekavian

The system is not operational yet;

Improvements are necessary: correction of Ijk part of dictionaries, enlargement of Ijk corpus, evaluation (Ijk expert needed);

The web application same as for DR (Ranka & co.).

What has been achieved?

OCR correction

The system is operational;

improvements have to be done to speed up the process etc.;

The web application is pending (Ranka & co.);

Diacritics restoration

The system is operational;

The web application is being done but ... (Ranka & co.);

Ekavian ( ) Ijekavian

The system is not operational yet;

Improvements are necessary: correction of Ijk part of dictionaries, enlargement of Ijk corpus, evaluation (Ijk expert needed);

The web application same as for DR (Ranka & co.).