Text-Induced Corpus Clean-up: current state-of-the-art

Martin Reynaert[1 2], Maarten van Gompel[3], Ko van der Sloot[3] and Antal van den Bosch[2 3]

TiCC - Tilburg University[1] / Meertens Institute - Amsterdam[2] / Center for Language Studies - Radboud University Nijmegen[3]

Workshop Processing of Historical Corpora
IXA research group Donostia, Spain June 11, 2018

Faculty of Arts
Centre for Language and Speech Technology
Radboud University Nijmegen

Text-Induced Corpus Clean-up: Introduction

TICCL for TYPOS and OCR-errors

- Tool to perform large scale, unsupervised spelling correction of corpora.
- Spelling correction = reduction of lexical variation caused by typos, OCR-errors, historical orthographical changes...
- Prototype developed during a pilot project by invitation of the National Library, The Hague in 2008.
- Development continues, TICCL code available in Open Source at `https://github.com/martinreynaert/TICCL`
- TICCL has been made multilingual.

TEXT-INDUCED CORPUS CLEAN-UP: BASIC RETRIEVAL MECHANISM

- Represent identical bags of characters (i.e. word strings sharing the same bag of characters) by an identifying numerical value,
- Use this value as the index key to the word strings in a database
- Perform simple calculations to retrieve variants from the database.

ANAGRAM HASHING

$$Key(w) = \sum_{i=1}^{|w|} f(c_i)^n$$

- A bad hashing function: produces collisions.
- Lines up ANAGRAMS: strings consisting of the same bag of characters.
- In practice, we used to use the code value of each character in the string raised to the power 5.
- Values obtained for the string are summed.

ANAGRAM HASHING II

- CAT = anagram of ACT and TAC
- $A + C + T = 65^5 + 67^5 + 84^5 = 6{,}692{,}535{,}156$
- $C + A + T = 67^5 + 65^5 + 84^5 = 6{,}692{,}535{,}156$
  - ALLOWS FOR ADDITION AND SUBTRACTION
  - SAME APPLIES FOR WORD COMBINATIONS, PHRASES, SENTENCES...
  - Great for discovering anagrams: citric critic, cosmic comics, pentatonische pistachenoten
  - BASIS FOR TISC: Text-Induced Spelling Correction

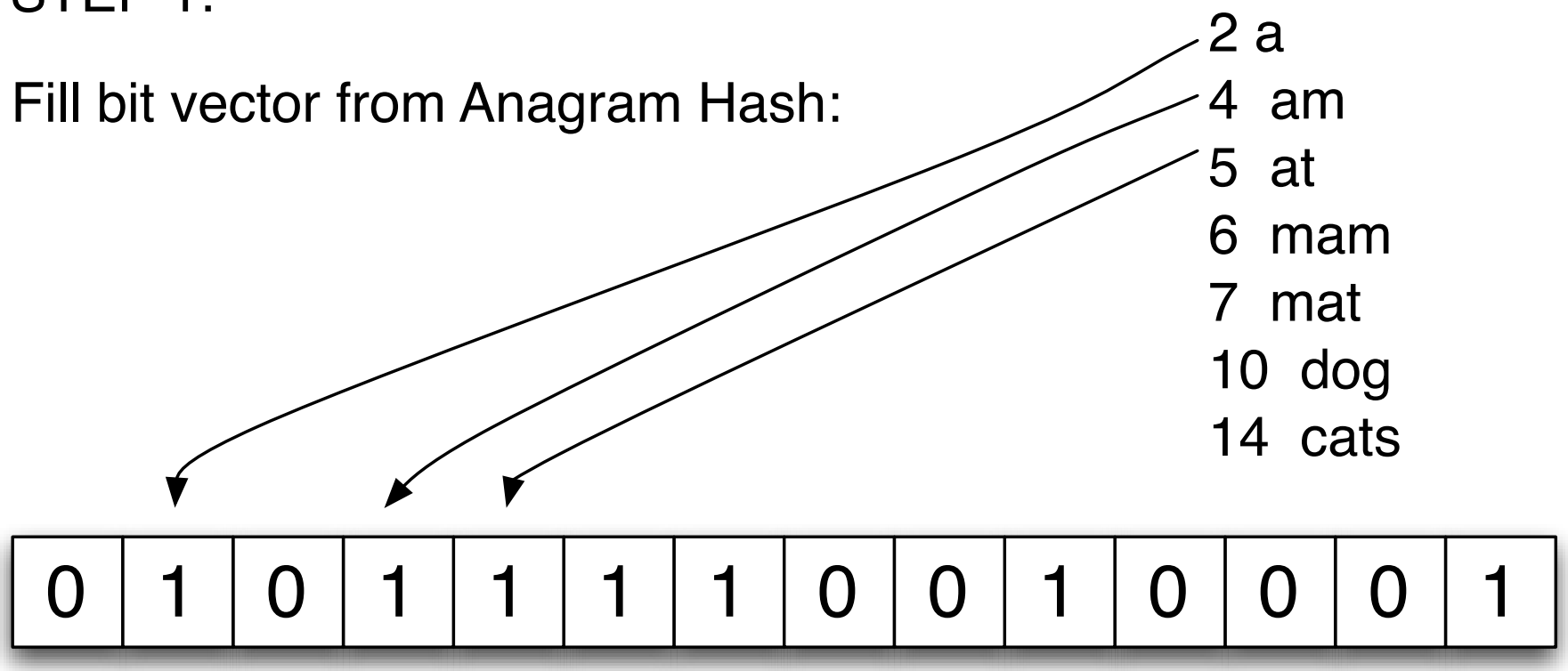
ANAGRAM HASHING III

- Given ANAGRAM VALUE (AV): 6,692,535,156
- AV(ACT) + 84^5 (plus T) = TACT
- AV(ACT) - 67^5 (minus C) = AT, TA
- AV(ACT) - 84^5 + 82^5 (minus T, plus R) = CAR
- AV(ACT) - 84^5 + 78^5 + 83^5 (minus T, plus N, plus S) = CANS/SCAN
- Focus word approach: take a word and systematically search for its variants, then take the next word..., etc.
- OR:
- Character Confusion approach: systematically search for all word pairs in the corpus that display a particular difference in characters for all possible confusions given a particular edit distance

Anagram Hashing Visualized - Step 1

STEP 1:

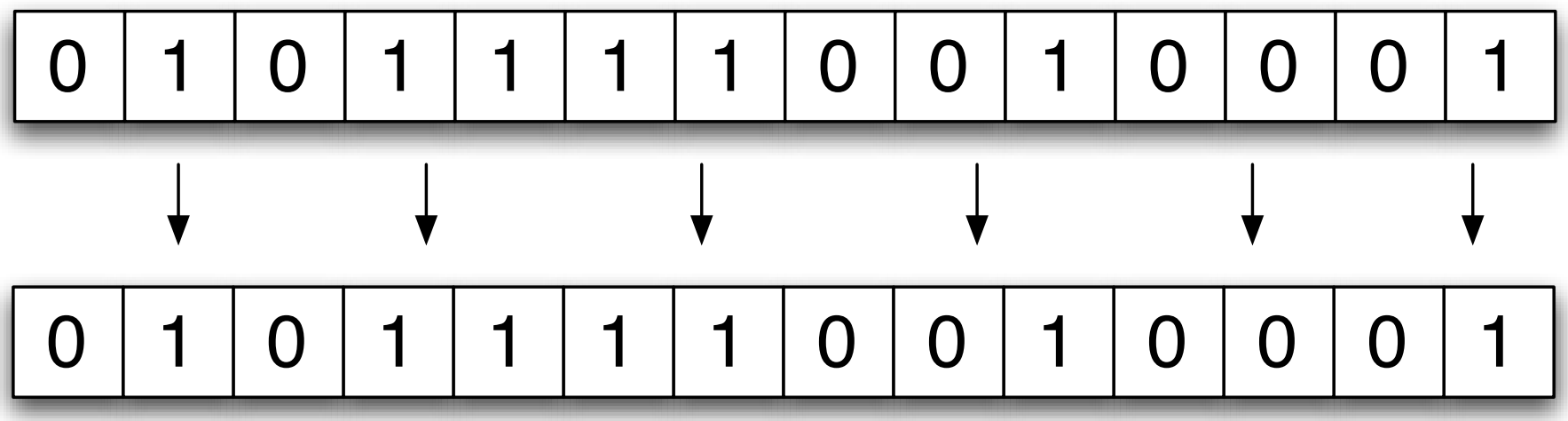
Fill bit vector from Anagram Hash:



Anagram Hashing Visualized - Step 2

STEP 2:

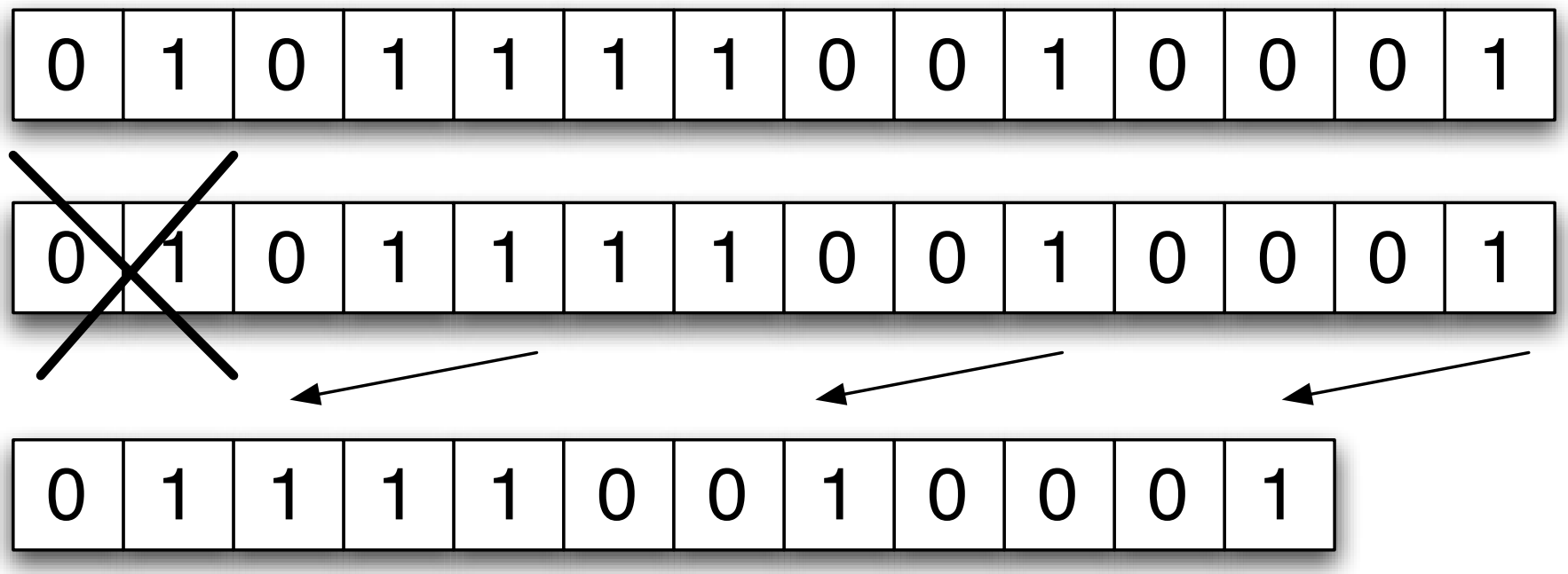
Clone the bit vector:



Anagram Hashing Visualized - Step 3

STEP3:

Truncate the cloned bit vector: AV = 2



Anagram Hashing Visualized - Step 4

STEP4:

Perform Boolean AND:

0	1	0	1	1	1	1	0	0	1	0	0	0	1

AND:

0	1	1	1	1	0	0	1	0	0	0	1

=

0	1	0	1	1	0	0	0	0	0	0	0

Anagram Hashing Visualized - Step 5

STEP5:

Retrieve confusion pairs from anagram hash:

Bit number - Bit number + AV char. confusion

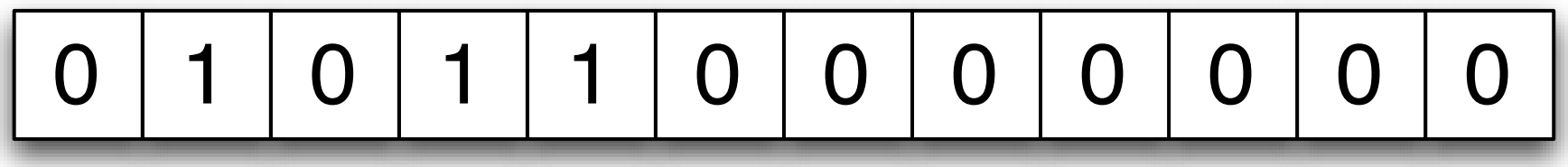
Result = confusion pairs:

a - am

am - mam

at - mat

2 a
4 am
5 at
6 mam
7 mat
10 dog
14 cats



Counting with Words in Euclidean and Multi-Dimensional Spaces

- In the one-dimensional space we can easily and exhaustively search for lexical variation between words up to a specific ‘edit distance’, e.g. words differing by up the 3 characters.
- Chosen metric: Euclidean distance.
- In multi-dimensional space we will look for words that share meaning, regardless of superficial character resemblance.
- Chosen metric: Cosine distance.

TICCL: recent advances

- Now allows for word bi- and trigram correction
- Allows for solving split and run-on word problems
- Allows for more precise short word correction
- We have now implemented ’chaining’ to collect OCR-variants beyond LD 2

TICCL: recent advances

- Word ngram correction
- Applied bigram correction to SGD (Political Mashup version)
- Solves word splits, run-ons and short words
  - Almost 200 years of Dutch Acts of Parliament
  - About 775 million word tokens (775,040,652)
  - About 6.6 million word types
  - About 63 million word bigrams
  - About 2.5 million ‘corrections’ on word type level
  - About 12 million ‘corrections’ on word token level

PICCL: Introduction

- Within the Dutch version of CLARIN, we are working on a new corpus building tool called PICCL. It constitutes a complete workflow for corpus building.
- PICCL or Philosophical Integrator of Computational and Corpus Libraries is a CLARIAH project. The system is used in NWO 'Groot' project Nederlab `https://www.nederlab.nl/cms/`.
- PICCL moves beyond demonstrator status and is to be an actual production system.
- It is currently being tested in 4 CLARIAH pilot projects.
- Should be fully deployed at CLARIN Center INT (Leiden) by summer 2018

PICCL: An integrated pipeline

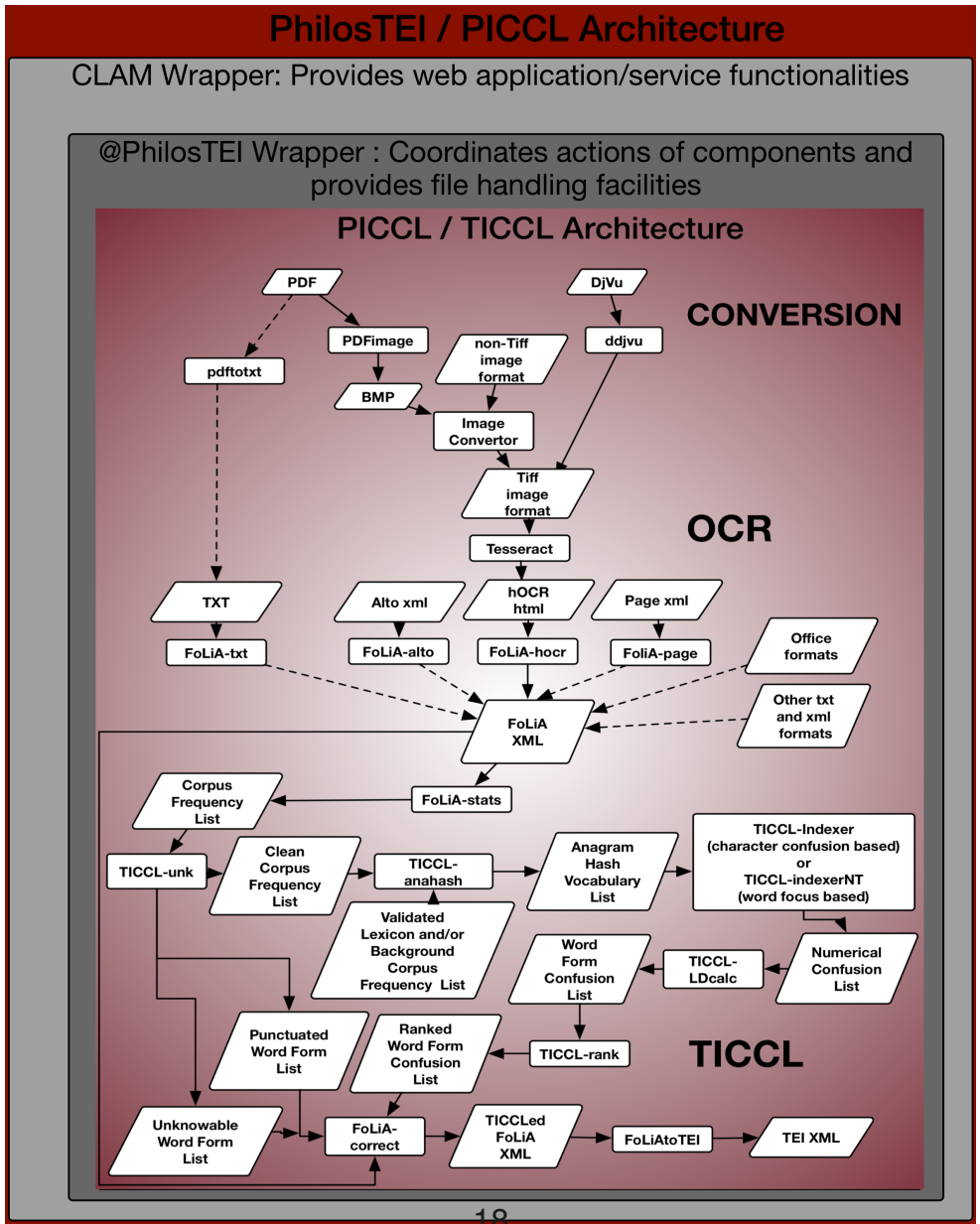
The integrated PICCL pipeline offers:

- a comprehensive range of conversion facilities for legacy electronic text formats
- Optical Character Recognition for text images, i.e. Tesseract OCR engine
- automatic text correction and normalization, i.e. TICCL OCR post-correction tool
- linguistic annotation, i.e. Frog Linguistic Enrichment tool
- and indexing for corpus exploration and exploitation environments (BlackLab and possibly WhiteLab)

Main Work Flow Components for corpus building

- Conversion: a choice selection of available open-source image and text convertors have been incorporated in the work flow.
- Optical Character Recognition: Tesseract is currently the OCR engine of choice.
- Pivot format: the format of choice central to the whole work flow is FoLiA XML.
- OCR post-correction: a new, modular and distributable implementation of Text-Induced Corpus Clean-up (online processing system) or TICCL(ops) provides diachronic and multilingual normalisation and transcription facilities.
- Book collation: The digitised and post-corrected book is finally delivered as a single tome in TEI XML format whatever the number of input files, whatever their original format.

PICCL Overview



PICCL Overview

- PICCL is wrapped in a single efficient CLAM-based web service/application. The Computational Linguistic Application Mediator is also one of our early CLARIN-NL achievements.
- The actual work flow is implemented in Nextflow. URL: `https://www.nextflow.io/index.html`
- The user-friendly system will be made available as a large black box to process a book's images into a digital version with next to no user intervention or prior knowledge required. It will equally well be equipped with the necessary interface options to allow more sophisticated users to address any submodule or combination of submodules individually at will.

PICCL: further functionalities

Output text is in FoLiA XML. The pipeline will therefore offer the various software tools that support FoLiA.

- Language categorization may be performed by the tool FoLiA-langcat at the paragraph level.
- TICCL – Text-Induced Corpus Clean-up – performs automatic post-correction of the OCRed text.
- Dutch texts may optionally be annotated automatically by Frog, i.e. tokenized, lemmatized and classified for parts of speech, named entities and dependency relations.
- The FoLiA Linguistic Annotation Tool (FLAT) will provide for manual annotation of e.g. metadata elements within the text – for later extraction.
- FoLiA-stats delivers n-gram frequency lists for the texts' word forms, lemmata, and parts of speech.

PICCL: future availability

- Will have its own website soon.
- Is available in LaMachine.
  Cf. https://github.com/proycon/LaMachine
  - As a Virtual Machine - easiest, allows you to run our software on any host OS.
  - As a Docker application
  - As a compilation/installation script in a virtual environment for full flexibility

CLARIN in the Low Countries

- Open Access book on Dutch and Belgian CLARIN activities
- DOI: `https://doi.org/10.5334/bbi`

LaMa software

- URL:
  `https://webservices-lst.science.ru.nl/portal/`

PICCL Subservice: AHA

- AHA = Anagram Hashing Application
- A new web service that derives character confusions and error type statistics from lists of word pairs displaying lexical variation (typo's, OCR-errors, diachronic and dialectical variation, ...
- Outputs table in Latex format, copy-and-paste ready
- To be further expanded with evaluation capabilities
- Available from http://ticclops.uvt.nl/AHA/

PICCL Subservices: AHA: Latex formatted AHA-output

Category	Levenshtein Distance							Total	%
	1	2	3	4	5	6	7		
deletion	78	14	4					96	17.55
insertion	69	9	1					79	14.44
substitution	108	29	5	5	2			149	27.24
transposition		28						28	5.12
multisingle		23	12	5				40	7.31
multiple		65	42	19	9	3		138	25.23
space deletion	13								2.38
space insertion									0.00
capitalisation									0.00
TOTAL	268	168	64	29	11	3		547	
%	48.99	30.71	11.70	5.30	2.01	0.55			99.3

ENJOY!!

Thanks for your attention!

`https://webservices-lst.science.ru.nl/portal/`

Text-Induced Corpus Clean-up: current state-of-the-art

Martin Reynaert[1 2], Maarten van Gompel[3], Ko van der Sloot[3] and Antal van den Bosch[2 3]

TiCC - Tilburg University[1] / Meertens Institute - Amsterdam[2] / Center for Language Studies - Radboud University Nijmegen[3]