Computational stemmatology - Project ANR SHERBET

Using artificial intelligence to reconstruct the stemmatology of the Hebrew Bible

Sophie Robert-Hayek - Frédérique Rey

University of Lorraine
Écritures - *Maison des Sciences de l'Homme*

15 Décembre 2023

1. The SHERBET project
2. Computational stemmatology
   - Distance based algorithms
   - Probability algorithms
3. Motivation
4. Introducing StemmaBench
5. Next steps

The SHERBET project

Outline

1. The SHERBET project
2. Computational stemmatology
3. Motivation
4. Introducing StemmaBench
5. Next steps

The SHERBET project

SHERBET (Stemmatology for the HEbRew BiblE Transmission)

4 years funding (French ANR grant) to reconstruct the genealogical linkage of Qumran and Cairo Genizah manuscripts using **computational tools**, led by **Frédérique Rey**.

Consortium of philology laboratory (Ecritures), computer science laboratory (LORIA) and applied mathematics laboratories (IECL, LJK).

The SHERBET project

3 anticipated work packages:

1. **Benchmarking** and **calibrating stemmatology algorithms** to the Qumran and Cairo Genizah textual traditions;

The SHERBET project

3 anticipated work packages:

1. **Benchmarking** and **calibrating stemmatology algorithms** to the Qumran and Cairo Genizah textual traditions;
2. Development of **novel** computational stemmatology algorithms:
   - Using a precise probability transition model;
   - Leveraging recent advances in Natural Language Processing;
   - Outperforming current algorithms;

The SHERBET project

3 anticipated work packages:

1. **Benchmarking** and **calibrating stemmatology algorithms** to the Qumran and Cairo Genizah textual traditions;
2. Development of **novel** computational stemmatology algorithms:
   - Using a precise probability transition model;
   - Leveraging recent advances in Natural Language Processing;
   - Outperforming current algorithms;
3. Applications of these algorithms to **build the genealogical lineage of several traditions**, starting with Hebrew manuscripts of Ben Sira.

Computational stemmatology

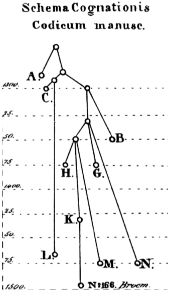
Outline

1. The SHERBET project
2. Computational stemmatology
    - Distance based algorithms
    - Probability algorithms
3. Motivation
4. Introducing StemmaBench
5. Next steps

Stemmatology

Stemmatology

Stemmatology consists in building the **genealogical lineage** of a set of textual witnesses by analyzing the textual **variants**, to better understand textual transformations and scribal behavior.



Usual method rely on **manual variant analysis**:

- Paul Maas conjunctive/separative errors (Maas 1958)

Computational stemmatology

The improvements in computational algorithms and computer speed has led to:

- the development of **automatic algorithms**;
- inspired from **biology**;

Computational stemmatology

The improvements in computational algorithms and computer speed has led to:

- the development of **automatic algorithms**;
- inspired from **biology**;

Computational stemmatology

Computational stemmatology uses **computational techniques and algorithms** to reconstruct the evolutionary relationships between the witnesses.

Computational stemmatology

Many algorithms have been designed over the last 50 years:

- Encoding "standard" stemmatology algorithms: Poole's algorithm (Camps 2015), RHM algorithm (Roos and Heikkila 2009) ...

Computational stemmatology

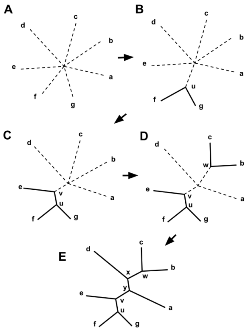
Many algorithms have been designed over the last 50 years:

- Encoding "standard" stemmatology algorithms: Poole's algorithm (Camps 2015), RHM algorithm (Roos and Heikkila 2009) ...
- Borrowing from **philogeny** (study of the evolutionary history among organisms):
  - Distance based algorithms;
  - Probabilistic based algorithms.

Distance based algorithms

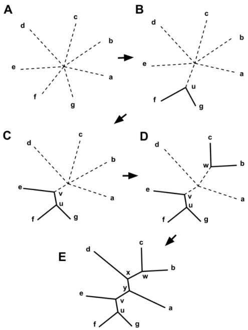
Computational stemmatology: distance based algorithms

- Define a distance **matrix** between manuscripts;
- Iteratively group together the closest manuscripts;



Computational stemmatology: distance based algorithms

- Define a distance **matrix** between manuscripts;
- Iteratively group together the closest manuscripts;



Example algorithms

UPGMA (Sokal and Michener 1958) and Neighbor Joining (Saitou N 1987).

Probability algorithms

Computational stemmatology: probability based algorithms

- Define a probability model of transition between manuscripts : probability of going from manuscript P_i to manuscript P_j given the **variants**;
- Select the tree that is the **most likely true** given the data.

The likelihood, $L(T)$ of observing the given manuscript data D, under the tree T and the tradition parameters Θ, can be calculated as:

$$L(T) = \mathbb{P}(D|T, \Theta)$$

Computational stemmatology: probability based algorithms

- Define a probability model of transition between manuscripts : probability of going from manuscript P_i to manuscript P_j given the **variants**;
- Select the tree that is the **most likely true** given the data.

The likelihood, $L(T)$ of observing the given manuscript data D, under the tree T and the tradition parameters Θ, can be calculated as:

$$L(T) = \mathbb{P}(D|T, \Theta)$$

Example algorithms

Bayesian Inference (Drummond and Bouckaert 2015), Maximum Likelihood trees (Felsenstein 1981) ...

Motivation

Outline

1. The SHERBET project
2. Computational stemmatology
3. Motivation
4. Introducing StemmaBench
5. Next steps

Selecting the right algorithms

Faced with as many possible choices ...

Selecting the right algorithms

Faced with as many possible choices ...

What algorithm should we select given a textual tradition ?

Selecting the right algorithms

Faced with as many possible choices ...

What algorithm should we select given a textual tradition ?

There is no single optimum algorithms that **will outperform all others** and the algorithms **should be selected given the particularity of each tradition**.

(Machine Learning/Deep Learning community refers to this as the "no free lunch" theorem)

Benchmarking of stemmatology algorithms

Benchmarking

Benchmarking refers **to the process of evaluating the performance** of a new model, algorithm, or technique by comparing it against established and standardized datasets, metrics, or existing models.

Benchmarking of stemmatology algorithms

Benchmarking

Benchmarking refers **to the process of evaluating the performance** of a new model, algorithm, or technique by comparing it against established and standardized datasets, metrics, or existing models.

Benchmarking is required to:

- Suggest new algorithms and compare them to the state of the art;
- Select the optimum algorithm given the particularity of a tradition.

Why should we benchmark stemmatology algorithms ?

Suggesting a new algorithm: A new variation of algorithms should **perform at least as well** on at least **one case study** to be an acceptable.

Why should we benchmark stemmatology algorithms ?

Suggesting a new algorithm: A new variation of algorithms should **perform at least as well** on at least **one case study** to be an acceptable.

Example

When suggesting to use a **new distance** in a distance based stemmatology algorithms such as Neighbor Joining, we should show that in practice it outperforms other textual distances to be **suggested as an alternative**.

Why should we benchmark stemmatology algorithms ?

Selecting the algorithm:

Select the best performing algorithms given **the characteristics of the variants within the studied textual tradition**.

Why should we benchmark stemmatology algorithms ?

Selecting the algorithm: **Select the best performing** algorithms given **the characteristics of the variants within the studied textual tradition**.

Example

Given a tradition with a :

- 1% word omission rate;
- 5% letter inversion;
- 2% word inversion;
- 10% of missing manuscripts;

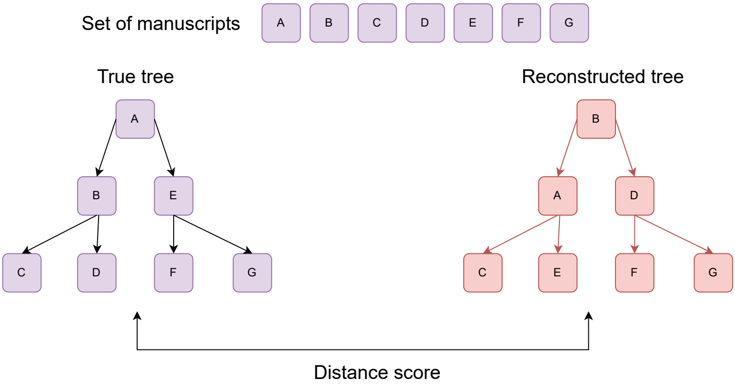
probability at each generation,
What algorithm should I select given these characteristics ?

Benchmarking of stemmatology algorithms

To perform benchmarking we need:

1. A golden standard (**ground truth**): a tradition where we know the true stemma;
2. A set of stemmatology algorithms to compare;
3. A metric to compare the different results between them.

Benchmarking of stemmatology algorithms



Set of manuscripts

True tree

Reconstructed tree

Distance score

Benchmarking of stemmatology algorithms

Two possible approaches:

Benchmarking of stemmatology algorithms

Two possible approaches:

- Use “real” **handwritten benchmarking data** or traditions with known ground truth.

Benchmarking of stemmatology algorithms

Two possible approaches:

- Use “real” **handwritten benchmarking data** or traditions with known ground truth.
- Use **computer generated traditions** that imitates observed traditions and simulates variants over time.

Using handwritten data

Landmark study of Roos et al. (Roos and Heikkila 2009) that compare 22 different variations of stemmatology algorithms on 4 traditions (3 synthetic, 1 “real”):

Data	**Number of manuscripts**
Heinrichi	67
Parzival	21
Notre besoin	14
Legend	52*

Using handwritten data (Roos and Heikkila 2009)

	Data		
Method	***Heinrichi* (%)**	***Parzival* (%)**	***Notre besoin* (%)**
RHM	**76.0**	79.9	76.9
PAUP*			
Parsimony	74.4	77.8	74.5
Parsimony BS[b]	73.6	85.4	77.3
Neighbour Joining	64.4	81.5	76.2
Neighbour Joining BS[b]	62.9	**87.1**	77.4
Least squares	64.2	81.5	70.2
Least squares BS[b]	62.6	79.8	73.0
n-Gram clustering	64.4	79.3	66.4
SplitsTree4			
NeighborNet	59.1	77.8	70.2
SplitDecomp.	53.1	74.5	73.1
ParsimonySplits	56.8	83.7	71.6
CompLearn	52.7	81.5	70.6
Hierarchical clustering	51.4	72.6	60.2
‘Classical’ method A[a]			74.4
‘Classical’ method B[a]			**85.1**
Weighted support method			66.3
Neighbour joining A			76.0
Neighbour joining B			75.0
Parsimony			74.4
Data compression			62.0

Limits

Handwritten texts in conditions that resemble the working conditions of scribes, but:

Limits

Handwritten texts in conditions that resemble the working conditions of scribes, but:

1. Very expensive;
2. No guarantee of being representative;
3. Very dependent on experimental parameters;
4. Hard to fine tune.

Benchmarking of stemmatology algorithms

Suggestion of a complementary approach **based on simulation** to generate **representative textual traditions**.

Benchmarking of stemmatology algorithms

Suggestion of a complementary approach **based on simulation** to generate **representative textual traditions**.

We present the **StemmaBench** Python library, a set of utilities to **generate ground truth traditions for benchmarking of stemmatology algorithms**.

Introducing StemmaBench

Outline

1. The SHERBET project
2. Computational stemmatology
3. Motivation
4. Introducing StemmaBench
5. Next steps

Purpose

StemmaBench is a Python library that allows you to quickly generate an artificial textual traditions given:

- An **input text**;
- A **configuration file**.

Example

Input text: Extract from *the Lion, The Witch and The Wardrobe*, by C.S. Lewis

> *Once there were four children whose names were Peter, Susan, Edmund and Lucy.*

Configuration file:

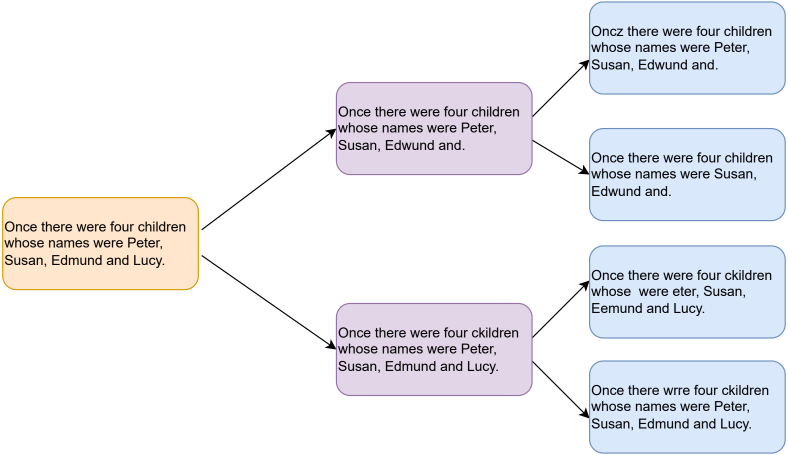
```
meta:
  language: eng

variants:
  words:
    synonym:
      law: Bernouilli
      rate: 0.01
    mispell:
      law: Bernouilli
      rate: 0.1
    omit:
      law: Bernouilli
      rate: 0.05
  sentences:
    duplicate:
      args:
        nbr_words: 2
      law: Bernouilli
      rate: 0.1

stemma:
  depth: 2
  width:
    law: Uniform
    min: 2
    max: 4
```

Example

StemmaBench will generate the following artificial tradition:



Once there were four children whose names were Peter, Susan, Edmund and Lucy.

Once there were four children whose names were Peter, Susan, Edwund and.

Oncz there were four children whose names were Peter, Susan, Edwund and.

Once there were four children whose names were Susan, Edwund and.

Once there were four ckildren whose names were Peter, Susan, Edmund and Lucy.

Once there were four ckildren whose were eter, Susan, Eemund and Lucy.

Once there wrre four ckildren whose names were Peter, Susan, Edmund and Lucy.

Main features

- **Easy install** using Python;

Main features

- **Easy install** using Python;
- **Fast** and **easy** tradition generation;

Main features

- **Easy install** using Python;
- **Fast** and **easy** tradition generation;
- **Effortless** result vizualization and manipulation;

Main features

- **Easy install** using Python;
- **Fast** and **easy** tradition generation;
- **Effortless** result vizualization and manipulation;
- **Flexible scribal modelization**: possibly to define your own probabilities for:
  - Mispells
  - Synonym insertions
  - Word omission
  - Word repetition
  - Word order change

Using stemmabench

Input:

- **Define the text**: `input_text.txt`
- **Define the wanted configuration**: `config.yml`

```
generate narnia.txt output_folder config.yaml
```

Using stemmabench

Input:

- **Define the text**: `input_text.txt`
- **Define the wanted configuration**: `config.yml`

```
generate narnia.txt output_folder config.yaml
```

Output:

- `edges.txt`: A display of the generated tree.
- A text file per generated manuscript within the tradition.

Supported languages

For now, supported languages for synonym generation are:

- English;
- Koiné Greek.

Incoming supported language: **biblical hebrew**.

Next steps

Outline

1. The SHERBET project
2. Computational stemmatology
3. Motivation
4. Introducing StemmaBench
5. Next steps

3 improvement direction

- **Improving** scribal modelization;
- **Estimating** the parameters of the tradition;
- Dealing with **contamination** / horizontal transmission.

Improving scribal modelization

- Introduce different probabilities depending on letters (make some switches more likely than others);

Improving scribal modelization

- Introduce different probabilities depending on letters (make some switches more likely than others);
- Introduce possible variants within words cut-off;

Improving scribal modelization

- Introduce different probabilities depending on letters (make some switches more likely than others);
- Introduce possible variants within words cut-off;
- Take into account word inflection whenever computing synonyms;

Improving scribal modelization

- Introduce different probabilities depending on letters (make some switches more likely than others);
- Introduce possible variants within words cut-off;
- Take into account word inflection whenever computing synonyms;
- Perform omissions depending on POS (adjectives and conjunctions are most likely to be dropped);

Improving scribal modelization

- Introduce different probabilities depending on letters (make some switches more likely than others);
- Introduce possible variants within words cut-off;
- Take into account word inflection whenever computing synonyms;
- Perform omissions depending on POS (adjectives and conjunctions are most likely to be dropped);
- Include manuscript fragmentation.

Estimating parameters of the traditions

Selecting **realistic configuration** file will require analysis of existing traditions: quite mathematically tricky !

Estimating parameters of the traditions

Selecting **realistic configuration** file will require analysis of existing traditions: quite mathematically tricky !

Current works are being done on **analyzing the statistical distribution of variants** in the:

- Traditions used by Roos et al. for result reproducibility.
- Ben Sira tradition (Genizah, Qumran ...)
- Isaiah tradition (Qumran, ...)

Dealing with horizontal transmission

Horizontal transmission refers to a situation where **two or more textual traditions or manuscripts that are being studied become mixed or intermingled**, leading to uncertainty about the relationships between different versions of a text.

Dealing with horizontal transmission

Horizontal transmission refers to a situation where **two or more textual traditions or manuscripts that are being studied become mixed or intermingled**, leading to uncertainty about the relationships between different versions of a text.

Very little work has been done regarding the impact of horizontal transmission on computational stemmatology algorithms.

Dealing with horizontal transmission

Horizontal transmission refers to a situation where **two or more textual traditions or manuscripts that are being studied become mixed or intermingled**, leading to uncertainty about the relationships between different versions of a text.

Very little work has been done regarding the impact of horizontal transmission on computational stemmatology algorithms.

StemmaBench will generate "contaminated" traditions to quantify its impact on the computational stemmatology algorithms.

Using the software

Link to GitHub project: `https://github.com/metz-theolab/stemmabench`.
Link to project's website: `https://metz-theolab.github.io/stemmabench/`.
Link to PyPi module: `https://pypi.org/project/stemmabench/`.

Questions

Questions ?

Bibliography I

- Camps, Jean-Baptiste (Jan. 2015). "Copie, authenticité, originalité dans la philologie et son histoire". fr. In: *Questes* 29, pp. 35–67. ISSN: 2102-7188, 2109-9472. DOI: 10.4000/questes.3535.
- Drummond, Alexei J. and Remco R. Bouckaert (2015). *Bayesian evolutionary analysis with BEAST*. Cambridge: Cambridge University Press.
- Felsenstein, Joseph (1981). "Evolutionary trees from DNA sequences: A maximum likelihood approach.". In: *Journal of Molecular Evolution* 17, pp. 168–376.
- Maas, Paul (Jan. 1958). *Textual criticism*. London: Oxford University Press. ISBN: 978-0-19-814318-5.
- Roos, Teemu and T. Heikkila (2009). "Evaluating methods for computer-assisted stemmatology using artificial benchmark datasets". In: *Literary and Linguistic Computing* 24, pp. 417–433.

Bibliography II

- Saitou N, Nei M. (July 1987). “The neighbor-joining method: a new method for reconstructing phylogenetic trees.”. In: *Mol Biol Evol.* 4, pp. 406–425.
- Sokal, Robert R. and Charles Duncan Michener (1958). “A statistical method for evaluating systematic relationships”. In: *University of Kansas science bulletin* 38, pp. 1409–1438. URL: `https://api.semanticscholar.org/CorpusID:61950873`.