

Localization of new textual variants within an existing stemma using neural networks & LLMs in stemmatology

Armin Hoenen



Outline

1 Historical Context

- Before Computers
- An LLMperiment
- Before DeepLearning
- DEEP LEARNING

2 Experiment

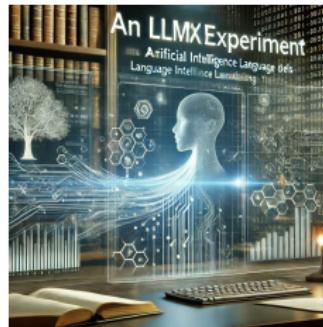
- Introduction & Related Work
- Methods & Materials
- Results
- Discussion & Conclusion

Some Historical Context

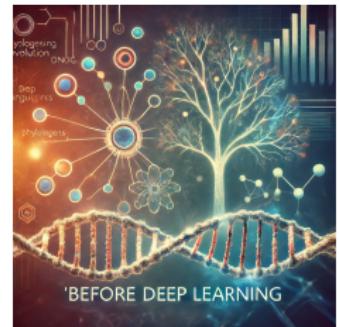
Before Computers



An LLMxperiment



Before DeepLearning



Some Historical Context

Before Con



To know from where we start

Before Computers

Before Print:

Stemmata unlikely; variation normal; focus on the authorial ideal text rather than the original.

Before Computers

Before Print:

Stemmata unlikely; variation normal; focus on the authorial ideal text rather than the original.

After Print:

- **Knowledge Explosion:** Print and colonialism helped amass knowledge at an unprecedented scale. In case of written works also variation.
- **Evolutionary Theory:** The rise of evolutionary theory influenced scientific and textual studies.

Before Computers

Before Print:

Stemmata unlikely; variation normal; focus on the authorial ideal text rather than the original.

After Print:

- **Knowledge Explosion:** Print and colonialism helped amass knowledge at an unprecedented scale. In case of written works also variation.
- **Evolutionary Theory:** The rise of evolutionary theory influenced scientific and textual studies.

Challenge:

Keeping track of all developments in a scientific field became increasingly demanding.

Narratives (before mid-19th c.) vs. Stemma (after)

Forcedly sequential Narratives:

Narrative 1

"From a now lost root *r*, two lost copies were made: *a* and *e*. *a* was copied twice into *A* and *O*, the latter of which was itself copied into *B* and *C*..." SEQ: r>a&e>A&O>B&C

Narratives (before mid-19th c.) vs. Stemma (after)

Forcedly sequential Narratives:

Narrative 1

"From a now lost root *r*, two lost copies were made: *a* and *e*. *a* was copied twice into *A* and *O*, the latter of which was itself copied into *B* and *C*..." SEQ: r>a&e>A&O>B&C

Narrative 2

"From a now lost root *r*, there were two hypothetical descendants: *a* and *e*. Whereas *e* has one copy, *D*, the subtree of *a* was more complex, leading to *A*, *O*, *B*, and *C*..." SEQ: r>a&e>D>A&O&B&C

Narratives (before mid-19th c.) vs. Stemma (after)

Forcedly sequential Narratives:

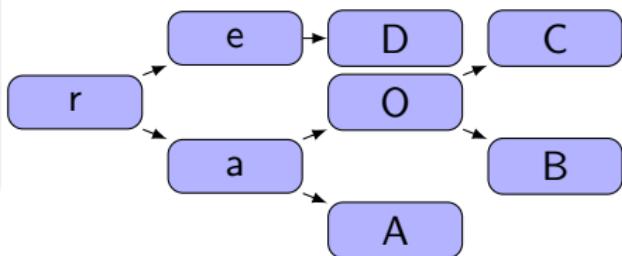
Narrative 1

"From a now lost root *r*, two lost copies were made: *a* and *e*. *a* was copied twice into *A* and *O*, the latter of which was itself copied into *B* and *C*..." SEQ: r>a&e>A&O>B&C

Narrative 2

"From a now lost root *r*, there were two hypothetical descendants: *a* and *e*. Whereas *e* has one copy, *D*, the subtree of *a* was more complex, leading to *A*, *O*, *B*, and *C*..." SEQ: r>a&e>D>A&O&B&C

Freely multi-informative stemma:



Advantage: Visualization (tree) displays all relationships at once, makes instantly comparable.

An experiment?

For comparison and the automatic analyses of large corpora of philological literature, one could ask, are LLMs able to **extract the relations from text** and ultimately stemmata. Not from the alignment, but from the meta-literature (possible also for descriptions of lost works).

An experiment?

For comparison and the automatic analyses of large corpora of philological literature, one could ask, are LLMs able to **extract the relations from text** and ultimately stemmata. Not from the alignment, but from the meta-literature (possible also for descriptions of lost works).

- **Task:** Extract a stemma (a tree) from a textual description of manuscript relationships.
- This is a specialization of relation extraction aimed at manuscript studies.
 - So far no literature in bio-sciences, papers publish the trees
 - relation-extraction for ontologies, species interactions, etc.
- **Goal:** Automatically extract the Newick tree format from text using LLMs.

Experimental Setup: Texts

Generate informationally equivalent textual descriptions which differ in sequence and wording. Input texts are built from ...

- 4 sentences with copy relations (1 for each parent): e.g. 'O was copied twice, resulting in C and B.', $O > C, B$)

Experimental Setup: Texts

Generate informationally equivalent textual descriptions which differ in sequence and wording. Input texts are built from ...

- 4 sentences with copy relations (1 for each parent): e.g. 'O was copied twice, resulting in C and B.', $O > C, B$)
- 5 or 7 alternatives for each sentence,

Experimental Setup: Texts

Generate informationally equivalent textual descriptions which differ in sequence and wording. Input texts are built from ...

- 4 sentences with copy relations (1 for each parent): e.g. 'O was copied twice, resulting in C and B.', $O > C, B$)
- 5 or 7 alternatives for each sentence,
- randomized sequence of the relations, randomized choice of the alternatives

Experimental Setup: Texts

Generate informationally equivalent textual descriptions which differ in sequence and wording. Input texts are built from ...

- 4 sentences with copy relations (1 for each parent): e.g. 'O was copied twice, resulting in C and B.', $O > C, B$)
- 5 or 7 alternatives for each sentence,
- randomized sequence of the relations, randomized choice of the alternatives
- 100 distinct permutations, 44 of which include **distractors** at random positions.

Experimental Setup: Texts

Generate informationally equivalent textual descriptions which differ in sequence and wording. Input texts are built from ...

- 4 sentences with copy relations (1 for each parent): e.g. 'O was copied twice, resulting in C and B.', $O > C, B$)
- 5 or 7 alternatives for each sentence,
- randomized sequence of the relations, randomized choice of the alternatives
- 100 distinct permutations, 44 of which include **distractors** at random positions.

Additionally, introductory sentences and a final sentence, which carry no relations.

Example Text

Text Description:

This text treats the tradition of Rabanus Testus Textus. The text has been transmitted in handwriting. We have located 5 extant copies in various libraries. e was copied, the copy is D. O and A are closely related, probably they have been copied from the same lost manuscript a. O was not copied from e. archetype r was copied into e and a O has been copied into B and C. The tradition is thus a limited one in size and scope but the relations are quite clear leading to a wonderful stemma albeit with descriptii and chains of hypothetical nodes.

Expected Tree:

$$r(e(D), a(O(C, B), A))$$

Prompt Engineering for GPT4-o

System role: You are a philologist...

Prompt Engineering for GPT4-o

System role: You are a philologist...

Zero-shot:

- Extract the Newick tree from this text:<text>

Prompt Engineering for GPT4-o

System role: You are a philologist...

Zero-shot:

- Extract the Newick tree from this text:<text>

One-shot: iteration of zero-shot with fake text and assistant stemma

Prompt Engineering for GPT4-o

System role: You are a philologist...

Zero-shot:

- Extract the Newick tree from this text:<text>

One-shot: iteration of zero-shot with fake text and assistant stemma

Chain of thought (CoT) Prompting

- Step 1: Extract the relations between witnesses mentioned in this text:<text>
- Step 2: Generate a tree in Newick fromat from:<res:step1>

Prompt Engineering for GPT4-o

System role: You are a philologist...

Zero-shot:

- Extract the Newick tree from this text:<text>

One-shot: iteration of zero-shot with fake text and assistant stemma

Chain of thought (CoT) Prompting

- Step 1: Extract the relations between witnesses mentioned in this text:<text>
- Step 2: Generate a tree in Newick format from:<res:step1>

In-context Learning:

- Added: For instance 'The archetype x has been copied into f and B. C and D are copies from B.' should result in $x(f, B(C, D))$.

Prompt Engineering for GPT4-o

System role: You are a philologist...

Zero-shot:

- Extract the Newick tree from this text:<text>

One-shot: iteration of zero-shot with fake text and assistant stemma

Chain of thought (CoT) Prompting

- Step 1: Extract the relations between witnesses mentioned in this text:<text>
- Step 2: Generate a tree in Newick format from:<res:step1>

In-context Learning:

- Added: For instance 'The archetype x has been copied into f and B. C and D are copies from B.' should result in $x(f, B(C, D))$.

Format augment:

- Provide only the tree

Results

Method	Hits	Misses	Accuracy (%)
Zero-shot	6	94	0.06
One-shot	89	11	0.89
CoT	9	81	0.09
Zero-shot with in-context	67	33	0.67
One-shot with in-context	94	6	0.94
CoT with in-context	82	18	0.82

Table: Performance of different approaches

Key Findings:

- cot-1shot and cot-1shot-incontext failed (0.08, 0.12) due to prompt engineering

Results

Method	Hits	Misses	Accuracy (%)
Zero-shot	6	94	0.06
One-shot	89	11	0.89
CoT	9	81	0.09
Zero-shot with in-context	67	33	0.67
One-shot with in-context	94	6	0.94
CoT with in-context	82	18	0.82

Table: Performance of different approaches

Key Findings:

- cot-1shot and cot-1shot-incontext failed (0.08, 0.12) due to prompt engineering
- In-context learning/few-shot significantly improved performance.

Results

Method	Hits	Misses	Accuracy (%)
Zero-shot	6	94	0.06
One-shot	89	11	0.89
CoT	9	81	0.09
Zero-shot with in-context	67	33	0.67
One-shot with in-context	94	6	0.94
CoT with in-context	82	18	0.82

Table: Performance of different approaches

Key Findings:

- cot-1shot and cot-1shot-incontext failed (0.08, 0.12) due to prompt engineering
- In-context learning/few-shot significantly improved performance.
- Chain-of-Thought prompting likewise improved performance.

Results

Method	Hits	Misses	Accuracy (%)
Zero-shot	6	94	0.06
One-shot	89	11	0.89
CoT	9	81	0.09
Zero-shot with in-context	67	33	0.67
One-shot with in-context	94	6	0.94
CoT with in-context	82	18	0.82

Table: Performance of different approaches

Key Findings:

- cot-1shot and cot-1shot-incontext failed (0.08, 0.12) due to prompt engineering
- In-context learning/few-shot significantly improved performance.
- Chain-of-Thought prompting likewise improved performance.
- Presence of distractors did not significantly affect accuracy.

Results

Method	Hits	Misses	Accuracy (%)
Zero-shot	6	94	0.06
One-shot	89	11	0.89
CoT	9	81	0.09
Zero-shot with in-context	67	33	0.67
One-shot with in-context	94	6	0.94
CoT with in-context	82	18	0.82

Table: Performance of different approaches

Key Findings:

- cot-1shot and cot-1shot-incontext failed (0.08, 0.12) due to prompt engineering
- In-context learning/few-shot significantly improved performance.
- Chain-of-Thought prompting likewise improved performance.
- Presence of distractors did not significantly affect accuracy.
- The format augment worked extremely well

Results

Method	Hits	Misses	Accuracy (%)
Zero-shot	6	94	0.06
One-shot	89	11	0.89
CoT	9	81	0.09
Zero-shot with in-context	67	33	0.67
One-shot with in-context	94	6	0.94
CoT with in-context	82	18	0.82

Table: Performance of different approaches

Key Findings:

- cot-1shot and cot-1shot-incontext failed (0.08, 0.12) due to prompt engineering
- In-context learning/few-shot significantly improved performance.
- Chain-of-Thought prompting likewise improved performance.
- Presence of distractors did not significantly affect accuracy.
- The format augment worked extremely well
- With incontext or one-shot, almost all examples adhered to format from examples

Future Work

Encouraging result:

- Expand dataset beyond toy data.
- Explore alternative LLM approaches (more LLMs) and fine-tuning.
- Investigate binary classification: are two textual descriptions equivalent.
- Experiment with inversion tasks: Generating descriptions from trees.

Future Work

Encouraging result:

- Expand dataset beyond toy data.
- Explore alternative LLM approaches (more LLMs) and fine-tuning.
- Investigate binary classification: are two textual descriptions equivalent.
- Experiment with inversion tasks: Generating descriptions from trees.

→ semi-automatic stemma extraction and comparison of large bodies of literatures or single traditions; trends, developments

Future Work

Encouraging result:

- Expand dataset beyond toy data.
- Explore alternative LLM approaches (more LLMs) and fine-tuning.
- Investigate binary classification: are two textual descriptions equivalent.
- Experiment with inversion tasks: Generating descriptions from trees.

→ semi-automatic stemma extraction and comparison of large bodies of literatures or single traditions; trends, developments

→ example question: did stemmata for a certain field grow in average size (for certain literatures) over time, constantly or with event-peaks

Before DeepLearning: Key Advances (1950s-1960s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
Early computers UNIVAC (1951)	Watson DNA (1953) Focus on blood group data (1963-65)	Early stemma models present early computation (Ellison 1957)	Swadesh List (1950) Glottochronology debates

Before DeepLearning: Key Advances (1950s-1960s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
Early computers UNIVAC (1951)	Watson DNA (1953) Focus on blood group data (1963-65)	Early stemma models present early computation (Ellison 1957)	Swadesh List (1950) Glottochronology debates

→ DNA introduces a whole new paradigm

Before DeepLearning: Key Advances (1970s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
Home PCs appear (Apple I, 1976) Early AI systems	Sanger (1977) sequencing	Computational stemma experiments	Lexicostatistics projects
			Indo-European studies

Before DeepLearning: Key Advances (1970s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
Home PCs appear (Apple I, 1976)	Sanger (1977) sequencing	Computational stemma experiments	Lexicostatistics projects
Early AI systems			Indo-European studies

→ matrices of shared features dominant

Before DeepLearning: Key Advances (1980s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
PCs reach MHz range	Felsenstein's MaxL for DNA sequences (1981)	Stemma models	Indo-European homeland debate
Rule-based AI	PAUP gains popularity	Evolutionary models applied to texts	Gimbutas' hypothesis

Before DeepLearning: Key Advances (1980s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
PCs reach MHz range	Felsenstein's MaxL for DNA sequences (1981)	Stemma models	Indo-European homeland debate
Rule-based AI	PAUP gains popularity	Evolutionary models applied to texts	Gimbutas' hypothesis

→ Computing trees in phylogenetics takes off aided by DNA input data

Before DeepLearning: Key Advances (1990s-2010s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
PCs reach 100 MHz	PHYLIP expands MaxL	Collate software	WALS project (1999)
Early ML tools	Bayesian methods (BEAST, MrBayes)	Robinson-Howe advances	Computational phylogenies
Cloud computing begins	Genome-scale phylogenetics		
Multi-core CPUs	Next-gen sequencing	Phylogenetic tools in stemmatology	Computational linguistics grows

Before DeepLearning: Key Advances (1990s-2010s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
PCs reach 100 MHz	PHYLIP expands MaxL	Collate software	WALS project (1999)
Early ML tools	Bayesian methods (BEAST, MrBayes)	Robinson-Howe advances	Computational phylogenies
Cloud computing begins	Genome-scale phylogenetics		
Multi-core CPUs	Next-gen sequencing	Phylogenetic tools in stemmatology	Computational linguistics grows

→ phylogenetics transfers some methodology to stemmatology

Before DeepLearning: Key Advances (1990s-2010s)

Computer Science	Phylogenetics	Stemmatology	Historical Linguistics
PCs reach 100 MHz	PHYLIP expands MaxL	Collate software	WALS project (1999)
Early ML tools	Bayesian methods (BEAST, MrBayes)	Robinson-Howe advances	Computational phylogenies
Cloud computing begins	Genome-scale phylogenetics		
Multi-core CPUs	Next-gen sequencing	Phylogenetic tools in stemmatology	Computational linguistics grows

- phylogenetics transfers some methodology to stemmatology
- computing capacity still grows and **pushes the boundaries** in every respect - larger tree spaces navigable, even for NP-hard problems, less need for heuristics and local maxima/minima. Compute-heavy methods become feasible (Felsenstein MaxL and Parsimony)

The Advent of Deep Learning



DeepLearning Revolution

Since around 2010 neural networks gain popularity, in 2015 AlphaGo beats Fan Hui, in 2017 the so called transformer mechanism leads to heavy improvements especially in NLP. Affected fields:

- machine vision (object recognition etc.)
- machine translation
- speech-recognition
- text classification
- sentiment analysis
- (board-)game AIs

DeepLearning Revolution

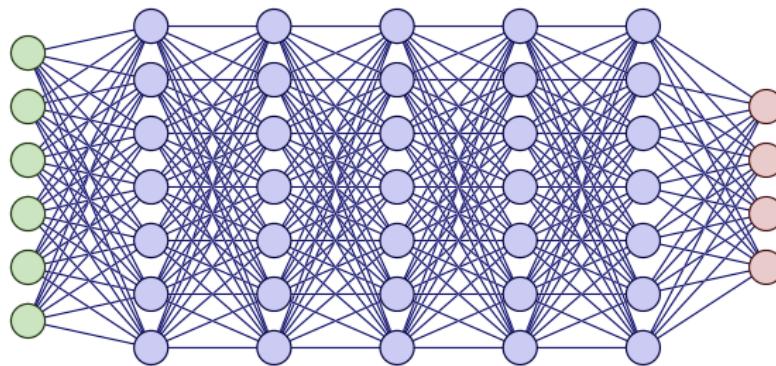
Since around 2010 neural networks gain popularity, in 2015 AlphaGo beats Fan Hui, in 2017 the so called transformer mechanism leads to heavy improvements especially in NLP. Affected fields:

- machine vision (object recognition etc.)
- machine translation
- speech-recognition
- text classification
- sentiment analysis
- (board-)game AIs

Almost all tasks in computer science were 'taken over' by deep-learning. Deep learning at scale, alias LLMs brought a second deeplearning revolution around and after 2020.

Deep Neural Network

based on regression (or perceptrons), $\sum_{i=1}^n w_i x_i$



$$(0.7 \cdot 17) + (0.3 \cdot 2) + (0.8 \cdot 3) + (0.9 \cdot 5) + (0.1 \cdot 7) + (0.3 \cdot 1) [\dots]$$

output classes: horse, elephant, mouse, tree

DL in other disciplines and phylogenetics

2022: nature communications survey on DL in biosciences

Table 1 Impact of Deep Learning on Computational Biology.

	Protein structure prediction	Protein function prediction	Genome engineering	Systems biology and data integration	Phylogenetic inference
Paradigm shifting	✓				
Major success		✓	✓		
Moderate success				✓	
Minor success					✓

"it is difficult to conceive of an end-to-end DL model to directly estimate phylogenetic trees from raw data in the near future"

DL in other disciplines and phylogenetics

2022: nature communications survey on DL in biosciences

Table 1 Impact of Deep Learning on Computational Biology.

	Protein structure prediction	Protein function prediction	Genome engineering	Systems biology and data integration	Phylogenetic inference
Paradigm shifting	✓				
Major success		✓	✓		
Moderate success				✓	
Minor success					✓

"it is difficult to conceive of an end-to-end DL model to directly estimate phylogenetic trees from raw data in the near future"

Well, what is the problem?

Deep Neural Networks -

Deep Learning Party everywhere, but in one small village...



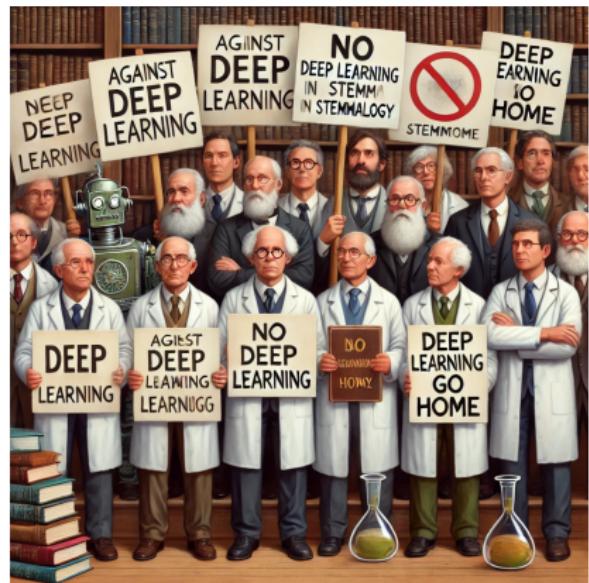
All other fields n tasks

Deep Neural Networks - out! #free_phylogen #old_ways

Deep Learning Party everywhere, but in one small village...



All other fields n tasks

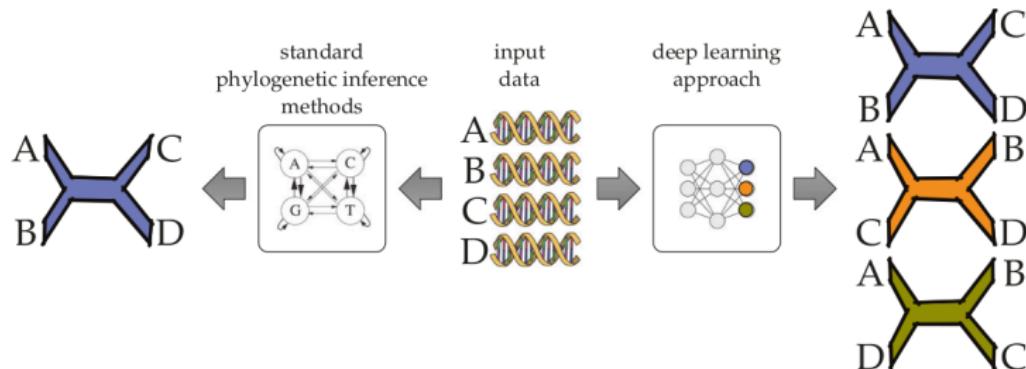


Phylogenetics

Phylogenetics

A problem is that the tree space of possible trees grows unfeasibly for output nodes. Hoenen et al. (2017): for Greg Trees representing 10 manuscripts already 102 515 201 984!

"Recently CNNs have been used to infer the unrooted phylogenetic tree on four taxa ... an analysis of the performance of the method ... shows that CNNs were not as accurate as other standard tree estimation methods, e.g., maximum likelihood, maximum parsimony, and neighbor joining"



And stemmatology? Philology?



but one small village... Main Experiment

Can we find a task, where DeepLearning can be applied successfully in stemmatology?



Introduction and related work

Phylogenetic placement...

- ...is the task of localizing a new node on a known tree
- ...may help corroborate the position of uncertain nodes/witnesses

Introduction and related work

Phylogenetic placement...

- ...is the task of localizing a new node on a known tree
- ...may help corroborate the position of uncertain nodes/witnesses

Related Work

① phylogenetics

- standard approach is to add the sequence data to the MSA and then generate a new tree analysing placement (the new tree can however be quite different from the previous one)
- Jiang et al. (2021) have introduced DEPP, a DL software which learns distance relations to place a new node correctly facing the problem of gene vs. species trees

② stemmatology

- tree generation approach (two sections) to investigate successive contamination

Algorithm

We use a **sequence to sequence** DL approach:

- from tree with known stemma, hold back any one leaf node

Algorithm

We use a **sequence to sequence** DL approach:

- from tree with known stemma, hold back any one leaf node
- for each held-back node, train a DL model with input = alignment for each pair of the other witnesses, output the pair's distance (number of edges on stemma)

Algorithm

We use a **sequence to sequence** DL approach:

- from tree with known stemma, hold back any one leaf node
- for each held-back node, train a DL model with input = alignment for each pair of the other witnesses, output the pair's distance (number of edges on stemma)
- prediction: let model estimate length between held back node and any other on the tree

Algorithm

We use a **sequence to sequence** DL approach:

- from tree with known stemma, hold back any one leaf node
- for each held-back node, train a DL model with input = alignment for each pair of the other witnesses, output the pair's distance (number of edges on stemma)
- prediction: let model estimate length between held back node and any other on the tree
- use a scoring to place node from estimates
 - only one node truly has edge dist 1: the parent; if also only one in the DL estimates, take that node
 - else: for each node + predicted distance localize and score each possible parent node; then retrieve score winner

Algorithm

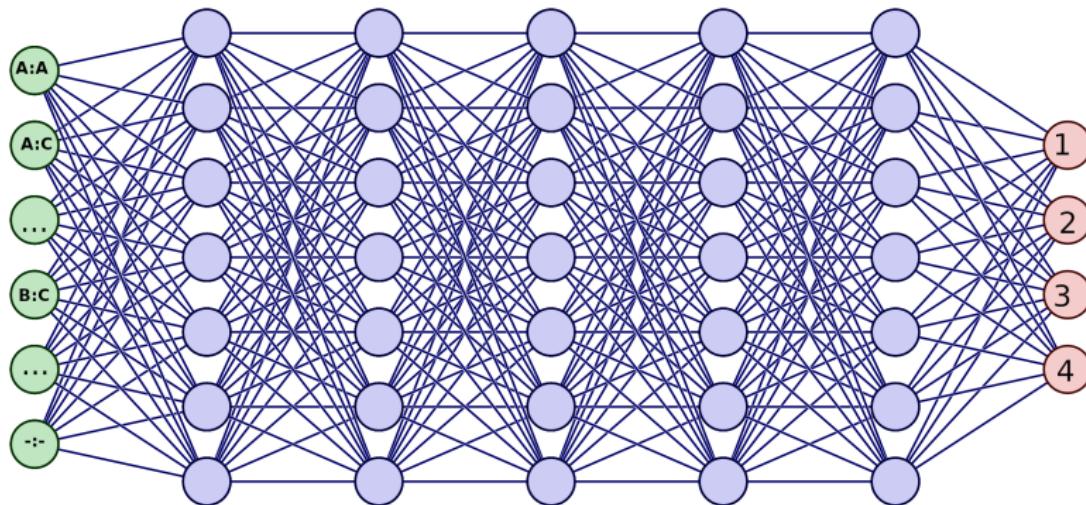
We use a **sequence to sequence** DL approach:

- from tree with known stemma, hold back any one leaf node
- for each held-back node, train a DL model with input = alignment for each pair of the other witnesses, output the pair's distance (number of edges on stemma)
- prediction: let model estimate length between held back node and any other on the tree
- use a scoring to place node from estimates
 - only one node truly has edge dist 1: the parent; if also only one in the DL estimates, take that node
 - else: for each node + predicted distance localize and score each possible parent node; then retrieve score winner
- evaluate placement and estimate accuracy

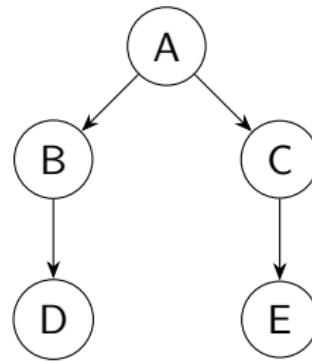
Training

Edges	pairwise position delta input	edge-dist
CC → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	2
DF → AW	A:A A:A A:A A:A A:A A:C -:- D A:A A:A A:A A:A	5
DL → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	1
GN → AW	A:A A:C A:A A:A A:A A:A -:- C A:A A:A A:A A:D	5
HS → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:C	5
HS2 → AW	A:A A:A A:A A:C A:A A:A -:- A:A A:A A:A A:A	5
IMBL → AW	A:A A:A A:A A:A A:A A:C -:- A:A A:A A:A A:A	4
JH → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	4
JS → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	3
JS2 → AW	A:A A:A A:A A:A A:A A:C -:- A:A A:A A:A A:A	5
JW → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	4
KH → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	4
LF → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	3
LM → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	2
ND → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	3
SD → AW	A:A A:C A:A A:A A:A A:A -:- C A:A A:A A:A A:D	4
SYL → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	2
WP → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	5
YT → AW	A:A A:C A:A A:A A:A A:A -:- C A:A A:A A:A A:D	5
ZM → AW	A:A A:A A:A A:A A:A A:A -:- A:A A:A A:A A:A	2

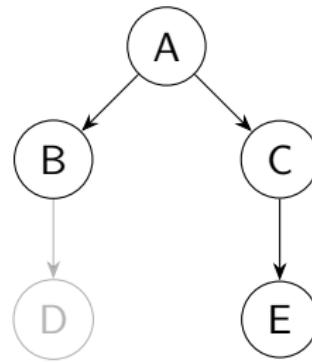
Trained



Initial DAG

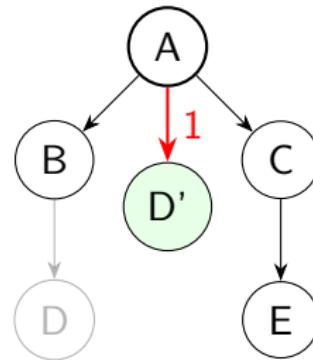


Held-back Node: D



Only one score is 1, attach leaf to A - mismatch

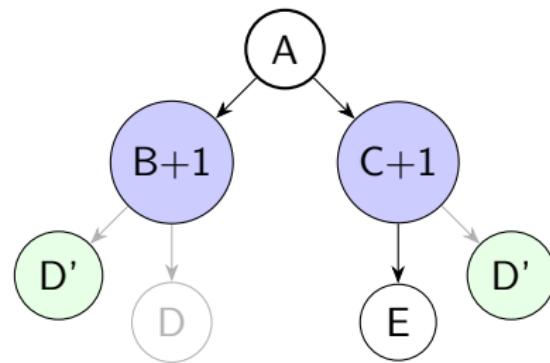
DL model learned from (A,B):1,(A,C):1,(A,E):1,(B,C):2,(B,E):3,(C:E):1
Predictions: (D',A):1, (D',B):3, (D',C):3, (D',E):2,



No score is 1, iterate and count

No prediction has value 1

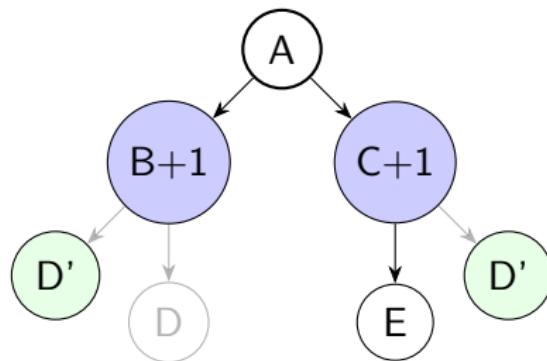
Predictions: **(D',A):2, (D',B):3, (D',C):3, (D',E):2**



No score is 1, iterate and count

No prediction has value 1

Predictions: **(D',A):2, (D',B):3, (D',C):3, (D',E):2**

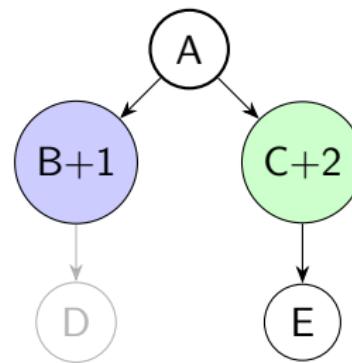


If $(D',A):2$, then either B or C could be the parent to D' .

iterate and count

No prediction has value 1

Predictions: $(D', A):2$, **$(D', B):3$** , $(D', C):3$, $(D', E):2$

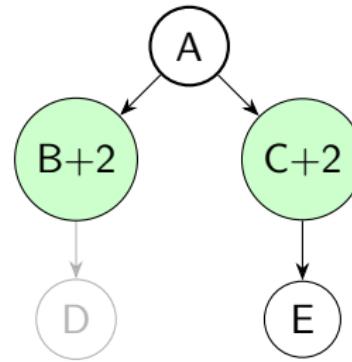


If $(D', B):3$, then only C could be the parent to D' .

iterate and count

No prediction has value 1

Predictions: $(D', A):2$, $(D', B):3$, $(D', C):3$, $(D', E):2$

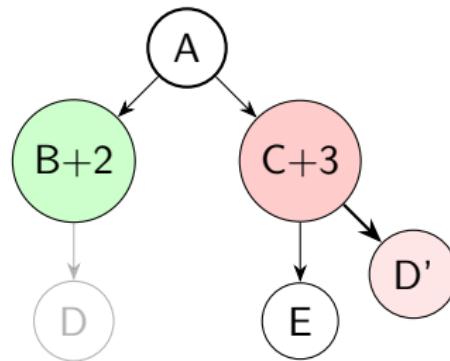


If $(D', C):3$, then only B could be the parent to D' .

still iterate and count (not branch and bound)

No prediction has value 1

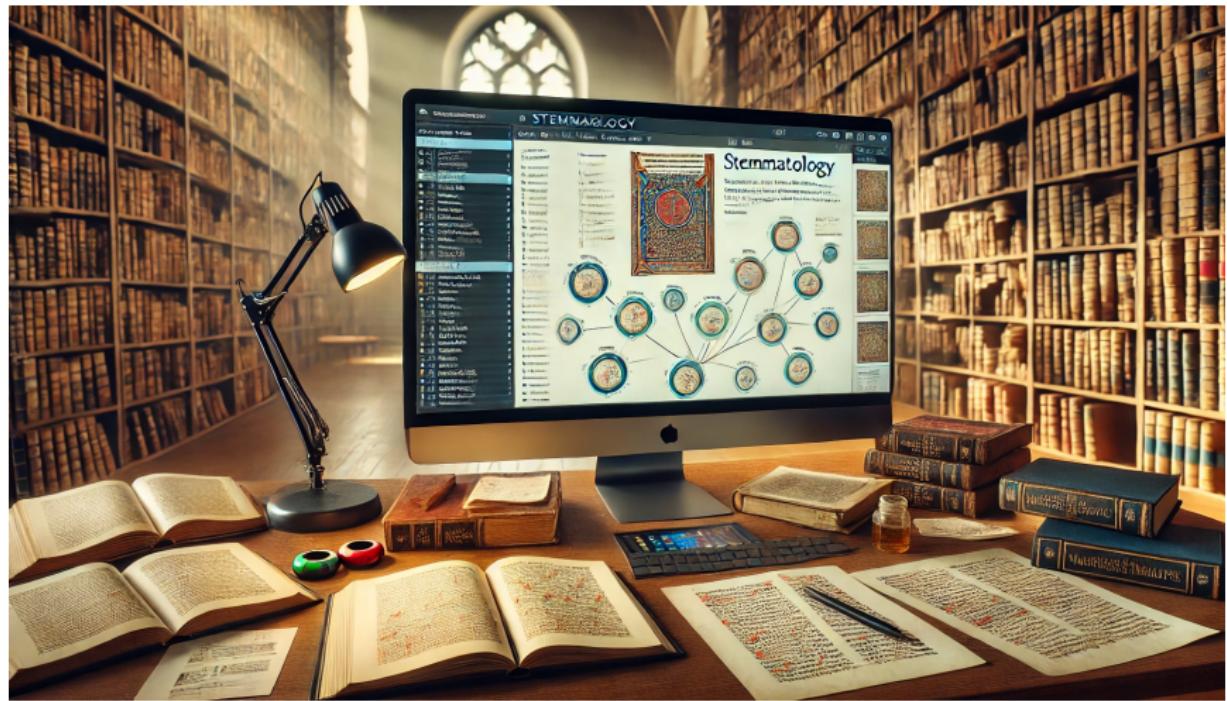
Predictions: $(D', A):2$, $(D', B):3$, $(D', C):3$, **$(D', E):2$**



If $(D', E):2$, then only C could be the parent to D' .

C is the score winner - mismatch.

Materials and Results



Materials - Data

Artificial datasets - mainly Parzival.

Datatype	Value	Note
<i>Language</i>	English (from German, archaic)	
<i>Copy-Mode</i>	manual, written	no contamination
<i>n. of mss</i>	21	
<i>n. of leafs</i>	12	
<i>n. of rows</i>	958	
<i>max dist leaf-leaf</i>	6	edges
<i>Publication</i>	Spencer et al. (2004)	

Results - Main Result

On BRNN with 512 rnn-size, 1 layer, no dropout, 5:185 validationset proportion, batch-size 16, python ONMT:

Feature	Value	Note
<i>correct predictions</i>	111/240 (0.46)	
<i>average deviation</i>	0.6 (SD:0.6)	
<i>max dist</i>	3	
<i>hitrate localizations</i>	9.5/12 (0.79)	distance of 3 misclassified from parent: 1
Random baseline	100.000 iterations	
<i>correct predictions</i>	40/240 (0.17)	max 67
<i>average deviation</i>	1.85 (SD:1.4)	
<i>max dist</i>	5	10 times not 5 (0.0001)

Results - data configuration

Various input variations were tested and compared:

- binary input, variant letters (sorted and unsorted), actual words,
variants sorted (A:C and C:A both encode A:C)

Results - data configuration

Various input variations were tested and compared:

- binary input, variant letters (sorted and unsorted), actual words,
variants sorted (A:C and C:A both encode A:C)
- only places of variation vs. **all places**

Results - data configuration

Various input variations were tested and compared:

- binary input, variant letters (sorted and unsorted), actual words,
variants sorted (A:C and C:A both encode A:C)
- only places of variation vs. **all places**
- validation set size: 5-185 and 10-180: no significant difference

Discussion



Discussion I

Small dataset, few nodes, better architectures probably around, ...

- DL method already clearly better than chance and even quite accurate in estimating placement
 - **In the collation there is edge-distance info**, not just text distance info: DL is a way to look at collation context

Discussion I

Small dataset, few nodes, better architectures probably around, ...

- DL method already clearly better than chance and even quite accurate in estimating placement
 - **In the collation there is edge-distance info**, not just text distance info: DL is a way to look at collation context
- all-places better than only-places-of-variation
 - hint to information also within what and how much is shared (which a computer might exploit better or more objectively than humans reading for difference)

Discussion II

Variants encoded, sorted performed best: [Why?](#)

- (encoded) Is the 'concentration profile' of scribes@copytime informative?
- (encoded) Does the exact error that is being made matter less than WHERE or at copy-time WHEN it is made (with plenty of possible errors once concentration slips)?
- (sorted) Is sorted better because it generalizes polarity between corrections and errors?

Discussion II

Variants encoded, sorted performed best: [Why?](#)

- (encoded) Is the 'concentration profile' of scribes@copytime informative?
- (encoded) Does the exact error that is being made matter less than WHERE or at copy-time WHEN it is made (with plenty of possible errors once concentration slips)?
- (sorted) Is sorted better because it generalizes polarity between corrections and errors?

Could this mean some generalizability across traditions?

FOLLOW-UP: Given letters for variants are tradition-agnostic, one can use models trained on one tradition for predictions on another.

- so far no conclusive results on side-experiment to this end

Conclusion



Conclusion

We have...

- ...demonstrated how LLMs could be used for stemma-from-text-extraction in stemmatology
- ...demonstrated a DL approach for placement in stemmatology

Q & A

