Meulemans

A read-only query layer over a multi-layer lexical corpus of 225 macrosystems

Alejandro Toledo Martínez · ORCID 0009-0000-1277-9697 · 2026

Abstract

A corpus of 3.79 million forms is not browsed: it is asked questions. And those a linguist asks of a word — where it comes from, what other words go with it, how it sounds in its relatives — cross layers that in the database are separate, so answering them requires writing SQL against eighty-seven tables and knowing their traps.

Meulemans is the layer that makes the Integrative Corpus queryable: a read-only API and a web interface, 3,006 lines between them, with seven views answering seven different questions about the same data. It adds no data and computes none: it displays them, with the source of each in plain sight.

We describe three decisions that shaped the project and are defended as transferable: it is read-only, because the corpus's verification apparatus only works if there is a single write path; it does not translate the data although the interface is bilingual, because a gloss is not the English version of a concept but its identifier; and it shows provenance next to the datum, which beyond satisfying the CC licences makes the corpus's unevenness visible rather than glossing over it.

We also say what the tool is not. It runs locally and is not deployed; it displays a corpus whose historical layer covers one macrosystem out of 225, so it is an Indo-European etymological browser with a very broad lexical catalogue around it; and it does not validate what it shows, because validation is the corpus's responsibility and a query layer that "fixed" what it displayed would introduce a second truth.

1 · The problem

A lexical corpus of 3.79 million forms is not browsed: it is asked questions. And the questions a linguist asks of a word — where does it come from? what other words go with it? how does it sound in its relatives? — do not fit in a table, because they cross layers that in the database are separate: the form, its sounds, its senses, its etymology, its cognates, its consonant skeleton.

Answering them today requires writing SQL against eighty-seven tables and knowing their traps — beginning with the two `source_id` columns that mean different things. That puts the corpus out of reach of anyone who did not build it, which is most of the people it could serve.

Meulemans is the layer that makes the Integrative Corpus queryable. It adds no data and computes none: it displays them.

1.1 · The name

It honours Dr Christiane S. Meulemans and Dr José Ángel Elías Fernández y Cuevas, of the Fundación Dr. J. Meulemans, who created endolinguistics. The tool generates no primary data: it integrates

existing sources, each cited and with its licence, in order to query them.

1.2 · What it is and is not

It is a browser for a word's history: you look up a form and see its sound, its sense, its cognates, its colexification and its lineage as far as it reaches, with its sources.

It is not an analysis tool. It computes no codes, builds no catalogues, proposes no reconstructions and does not write to the corpus. It is read-only by design, and §4.1 explains why that restriction is the project's most important architectural decision.

Nor is it a substitute for the database. Anyone doing research on the corpus works against the database directly; Meulemans is for looking, checking and showing.

2 · The seven views

Each answers a different question about the same data. The list is not a feature menu: it is the set of questions the corpus can answer today.

view	the question it answers
Search	what do I know about this word? It is the entry point: a form, and from there its full record
Compare	how do other languages say this? One concept across the languages attesting it
Coderivatives	what other words share a skeleton with this one, and for which concepts?
Genealogy	where does it come from? The form's lineage, edge by edge, with the source of each
Clusters	what word family does it form? Grouping by root and stem
Classes	what sounds does this language use, and how are they distributed? Inventory and class matrix
Map	where is it spoken? Geographic distribution of what is on screen

And two secondary ones that are not incidental: About, which says what this is, and Legal, which lists the eighteen sources with their licence and citation. The second is not a courtesy: it is compliance with the CC licences under which the corpus is distributed, and that is why it is a view and not a footnote.

2.1 · A form's record

It is the central view and deserves describing. For a single word it gathers:

- the orthography and transcription as the source wrote them, unaltered;
- its segments, one by one, with their articulatory features;
- its consonant skeleton and its class layer;
- its senses, with the concepts they are mapped to;
- its lineage, as far as the etymological layer reaches;
- its cognates, where they exist;
- and the source of every one of those things.

The last line is the point. Every datum the record displays carries where it came from, because the corpus carries row-level provenance and it would be a waste not to show it.

3 · Architecture

```
browser ──HTTP──► FastAPI (read-only API + statics) ──SQL──► Postgres
   vanilla JS SPA        api/  ·  1,225 lines            corpus_integrativo
   web/  ·  1,781 lines                                  (separate project)
```

piece	size	what it is
`api/queries.py`	783 lines	the queries, one per endpoint
`api/main.py`	253	routing and statics
`api/auth.py`	162	sessions, against a separate tiny database
`api/skeleton.py · db.py`	74	the display skeleton, and the connection pool
`web/js/`	1,781	the SPA: router, render, i18n and one page per view
`web/style.css`	399	all the styling

28 endpoints, all `GET` except the session one.

3.1 · No build step

The SPA is native-module JavaScript served as is. No bundler, no transpilation, no `node_modules`. You edit a file and reload.

It is a decision with a cost — no type checking, no dependency tree — and a gain that weighs more in a research project: the code you read in the repository is exactly the code that runs in the browser. There is no build layer between what one writes and what happens, and therefore no entire class of failures where the deployed artefact differs from the source.

3.2 · Hash routing, with all state in the URL

Every state — the search, the filters, the selection — lives in the hash. Two things come free from that: deep links work (you can share exactly what you are looking at) and the browser's back button does what it should.

That is what makes the tool citable: a claim about a word can travel with the link that shows it.

3.3 · Two databases with opposite requirements

```
corpus_integrativo   READ-ONLY · large · replaceable
meulemans_ops        read/write · 7.9 MB · IRREPLACEABLE
```

The distinction matters because it induces the expensive mistake: paying for backup and high availability on gigabytes of data that are already duplicated on the origin machine and can be regenerated, and not backing up the eight megabytes of accounts and usage log, which are the only thing that cannot be recovered.

4 · Three rules that shaped the project

4.1 · Read-only, and it is not a limitation

Meulemans does not write to the corpus. Not a table, not a column, not a counter.

The reason is not caution but separation of responsibilities. The corpus has its own verification apparatus — invariants that fail, a schema contract, a single door — and that apparatus only works if there is a single write path. A query layer that wrote, even an access log, would open a second path the invariants do not watch.

Hence accounts and usage live in a separate database. It is not duplication: it is keeping the corpus as an immutable artefact that can be regenerated whole from its pipeline without losing anything.

4.2 · The data is not translated

The interface is bilingual, English by default. And there is a rule that is not negotiable: the chrome is translated, never the data.

Labels, explanations, states and column headings are translated. Not the words, the IPA transcription, the skeletons, the codes, the glosses, the language, family or branch names, the source identifiers or the Dolgopolsky and SCA classes.

They come from the sources or are neutral, and translating them would falsify them. An English Concepticon gloss is not ”the English version” of a concept: it is that concept’s identifier, and translating it into Spanish would produce a datum no source supports.

4.3 · Every datum carries its source in plain sight

The corpus carries row-level provenance because CC licences require attribution (§9 of the corpus paper). Meulemans displays it, and not as hidden metadata: the source appears next to the datum.

It has a side effect more useful than legal compliance: it makes the corpus’s unevenness visible. Anyone looking at a record for an Indo-European language sees etymology and cognates; anyone looking at one from another macrosystem sees those sections empty, and sees why. The gap is shown rather than glossed over.

5 · State, and what deploying it would cost

Better said plainly: Meulemans runs locally today and is not deployed. A paper describing a public tool that was not public would be lying, so this section says where it stands and what is missing.

5.1 · What exists

It works: the seven views, the 28 endpoints, the bilingual layer, deep links and sessions. It comes up with one command against the corpus’s local database.

What is missing is not functionality but the polish separating a working tool from a public one: error handling at the edges, rate limits, and a review of what is shown to a visitor who does not know what a consonant skeleton is.

5.2 · What was measured about deployment

The figures are measured on the local cluster, not estimated:

the database served	8.8 GB after compaction (17 GB uncompacted)
accounts and usage	7.9 MB — the only irreplaceable thing
tables Meulemans queries	`form`, `segment`, `skeleton`, `sense`, `racimo`, `form_etymology`, `concept`, `cognate_member`, `morph_rule`, `sound_class`

The 8 GB difference between those two figures is not content: it is dead space. The `form` table occupies 2,242 MB against 1,311 MB with exactly the same 3,788,296 rows — bloat accumulated across rebuilds. It is the kind of figure that only surfaces while preparing a deployment and is worth measuring before paying for it.

5.3 · A correction: the public export is no longer needed for content

The deployment document describes two databases with different contents: the research one and a public one produced by filtering out non-redistributable sources. That has stopped being true.

Since 2026-09-05 the corpus's main database is the distributable one: the non-redistributable sources were withdrawn from it, with the cost measured and published. The etymology difference that document recorded — 1,266,185 against 1,262,354 — is exactly that withdrawal, and it is already applied.

So the public export retains one reason to exist, and it is not the stated one: compaction. It filters nothing because there is nothing left to filter.

It is noted here because it is the kind of document that ages silently: it correctly described a state of the world that changed in another repository.

5.4 · Where the code is, and how to request it

The repository is private and access is granted on request, by writing to alex@pixelspace.com with who you are and what for. Whoever receives it has full read access and can fork it to their own account.

It is the same criterion by which accounts to the application itself are granted (§4.1), and for the same reason: while the tool is maturing, each access is granted knowing who asked, so the API can keep changing without breaking anyone's work without warning.

The corpus it reads from is distributed under CC-BY-SA-4.0; access to the repository and the licence of the content are different things.

6 · Limits

6.1 · It shows what the corpus has, unevenness included

Meulemans cannot show what does not exist. The record for an Indo-European word brings etymology, cognates and lineage; the record for a word from another macrosystem brings form, sound and sense, with the other sections empty — because the corpus's historical layer covers one macrosystem out of 225.

That is not a flaw in the tool and it is not hidden: it is displayed, with the reason. But anyone arriving expecting a universal etymological browser will find an Indo-European one with a very broad lexical catalogue around it.

6.2 · It does not validate what it shows

The tool shows what the database contains, heuristics included. Two corpus columns look like data and are conjectures — the proper-noun detector by initial capital and the borrowing marker inherited from each source — and Meulemans does not correct them: at most it can flag them.

Validation is the corpus's responsibility, and its invariant apparatus's, not the query layer's. A visualisation tool that "fixed" what it displayed would be worse: it would introduce a second truth.

6.3 · It is not reproducible on its own

It depends on the corpus, and the corpus has a declared reproducibility limit today: none of its eighteen sources carries a pinned version. Two Meulemans installations against two rebuilds made at different times may show different things, and nothing warns of it.

When the corpus pins versions, Meulemans should display which one it is serving. It does not yet.

6.4 · No build, no types

The decision of §3.1 has its bill: there is no static checking of anything. At 1,781 lines of JavaScript this is manageable; beyond a certain size it would stop being so, and the decision would have to be revised rather than defended.

7 · References

Meulemans generates no data: it displays it. That gives its reference apparatus two parts worth keeping apart — where what it shows comes from, and what what it shows is built on.

7.1 · The provenance of what is displayed

None of the words, glosses, etymologies or cognates that appear in the interface is ours. They all come from the eighteen sources the Integrative Corpus integrates, each with its citation and its licence, and the full list is deliberately not reproduced here: duplicating it would let the two versions drift apart the day a source enters or leaves. It lives in the corpus's `source` table, which is what one consults to compose the attribution for a particular piece of work, and it is enumerated in §12 of the corpus paper.

What is worth saying here is the practical consequence: whoever cites a Meulemans screen is citing third-party data, and the attribution owed is to those sources, not to this layer.

7.2 · The other two works in this series

- Toledo Martínez, A. (2026). *Integrative Corpus: a multi-layer lexical database of 225 macrosystems, with row-level provenance and its verification apparatus.* — The database Meulemans queries. Meulemans is read-only: it neither writes to nor modifies anything there.

- Toledo Martínez, A. (2026). *The catalogue of consonantal codes: a comparative method without genealogy, cognacy or reconstruction.* — The other consumer of the same corpus, and the contrast that explains why there is no single figure for ”the usable corpus”: that work forbids itself to look at lineage, and this one displays it, because displaying it is what it does.

7.3 · What Meulemans uses as a tool

- PostgreSQL Global Development Group. *PostgreSQL*. <https://www.postgresql.org>
- Ramírez, S. *FastAPI*. <https://fastapi.tiangolo.com>