Digitizing Georges Six's *Dictionnaire biographique*: A Data Pipeline for French Revolutionary Military Biography

Rob Carroll

2026-02-28

Table of contents

Introduction

Between 1789 and 1815, France transformed its army root and branch. The Old Regime had structured military careers by seniority, lineage, and the venal sale of offices; the Revolution dissolved those structures wholesale. Careers that had taken a generation to advance compressed into years. Whether a man broke into the officer corps — and how quickly he rose once inside — depended less on birth than on the timing of his entry into service and on what the Republic needed from him that week.

These dynamics survive in extraordinary detail in Georges Six's *Dictionnaire biographique des généraux et amiraux français de la Révolution et de l'Empire* (1934), a two-volume work containing biographical entries for 2,117 generals and admirals who served between roughly 1792 and 1814. Each entry is a compact chronological record — rank promotions, unit assignments, campaign participations, wounds, honors, and administrative acts — set down clause by clause in a consistent semicolon-delimited prose style. The dictionary is, in effect, a longitudinal panel dataset in prose form.

This document describes the pipeline built to convert that prose into structured, machine-readable data. The entries encode:

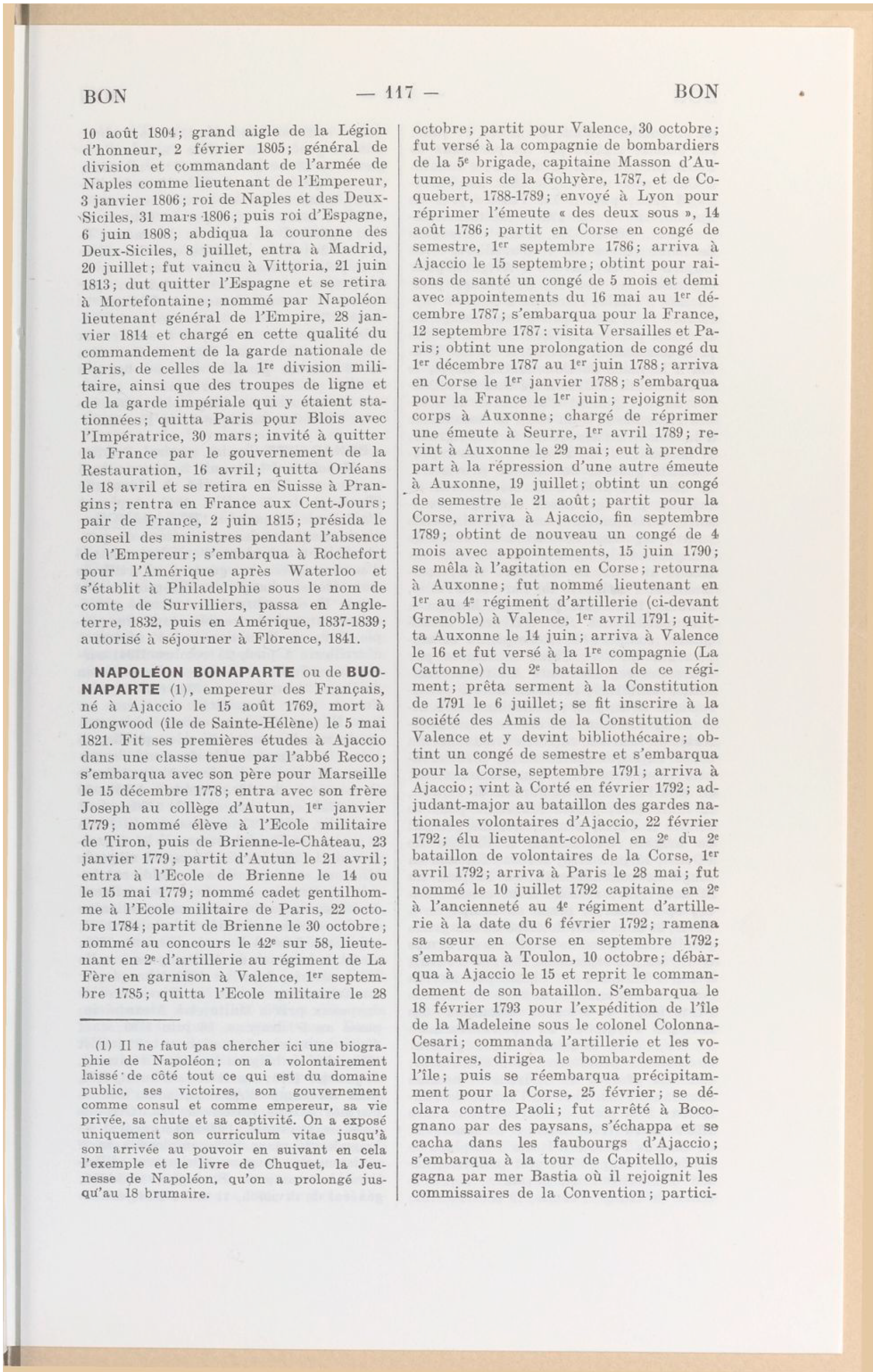
- personal details (name, birth and death dates, place of birth);
- every rank promotion, with date;
- military units served in and dates of service;
- campaigns and battles participated in;
- awards and honors received; and
- relationships with other military figures and political leaders.

Readers interested in what the structured data can reveal about French military careers, geography, and battlefield history should consult the companion document, *Six's Generals: A Quantitative Introduction.*

The original text

The original text of Six's *Dictionnaire* was published in print form in 1934. The text is organized alphabetically by the last name of the general or admiral, with each entry containing a narrative biography of the individual's life and military career. The entries vary in length and detail, with some providing extensive information about the individual's background, military service, and achievements, while others are more concise. The text is written in French and uses a formal, historical style.

A sample page from the original text is shown below:



10 août 1804; grand aigle de la Légion d'honneur, 2 février 1805; général de division et commandant de l'armée de Naples comme lieutenant de l'Empereur, 3 janvier 1806; roi de Naples et des Deux-Siciles, 31 mars 1806; puis roi d'Espagne, 6 juin 1808; abdiqua la couronne des Deux-Siciles, 8 juillet, entra à Madrid, 20 juillet; fut vaincu à Vittoria, 21 juin 1813; dut quitter l'Espagne et se retira à Mortefontaine; nommé par Napoléon lieutenant général de l'Empire, 28 janvier 1814 et chargé en cette qualité du commandement de la garde nationale de Paris, de celles de la 1re division militaire, ainsi que des troupes de ligne et de la garde impériale qui y étaient stationnées; quitta Paris pour Blois avec l'Impératrice, 30 mars; invité à quitter la France par le gouvernement de la Restauration, 16 avril; quitta Orléans le 18 avril et se retira en Suisse à Prangins; rentra en France aux Cent-Jours; pair de France, 2 juin 1815; présida le conseil des ministres pendant l'absence de l'Empereur; s'embarqua à Rochefort pour l'Amérique après Waterloo et s'établit à Philadelphie sous le nom de comte de Survilliers, passa en Angleterre, 1832, puis en Amérique, 1837-1839; autorisé à séjourner à Florence, 1841.

NAPOLÉON BONAPARTE ou de **BUONAPARTE** (1), empereur des Français, né à Ajaccio le 15 août 1769, mort à Longwood (île de Sainte-Hélène) le 5 mai 1821. Fit ses premières études à Ajaccio dans une classe tenue par l'abbé Recco; s'embarqua avec son père pour Marseille le 15 décembre 1778; entra avec son frère Joseph au collège d'Autun, 1er janvier 1779; nommé élève à l'Ecole militaire de Tiron, puis de Brienne-le-Château, 23 janvier 1779; partit d'Autun le 21 avril; entra à l'Ecole de Brienne le 14 ou le 15 mai 1779; nommé cadet gentilhomme à l'Ecole militaire de Paris, 22 octobre 1784; partit de Brienne le 30 octobre; nommé au concours le 42e sur 58, lieutenant en 2e d'artillerie au régiment de La Fère en garnison à Valence, 1er septembre 1785; quitta l'Ecole militaire le 28 octobre; partit pour Valence, 30 octobre; fut versé à la compagnie de bombardiers de la 5e brigade, capitaine Masson d'Autume, puis de la Gohyère, 1787, et de Coquebert, 1788-1789; envoyé à Lyon pour réprimer l'émeute « des deux sous », 14 août 1786; partit en Corse en congé de semestre, 1er septembre 1786; arriva à Ajaccio le 15 septembre; obtint pour raisons de santé un congé de 5 mois et demi avec appointements du 16 mai au 1er décembre 1787; s'embarqua pour la France, 12 septembre 1787: visita Versailles et Paris; obtint une prolongation de congé du 1er décembre 1787 au 1er juin 1788; arriva en Corse le 1er janvier 1788; s'embarqua pour la France le 1er juin; rejoignit son corps à Auxonne; chargé de réprimer une émeute à Seurre, 1er avril 1789; revint à Auxonne le 29 mai; eut à prendre part à la répression d'une autre émeute à Auxonne, 19 juillet; obtint un congé de semestre le 21 août; partit pour la Corse, arriva à Ajaccio, fin septembre 1789; obtint de nouveau un congé de 4 mois avec appointements, 15 juin 1790; se mêla à l'agitation en Corse; retourna à Auxonne; fut nommé lieutenant en 1er au 4e régiment d'artillerie (ci-devant Grenoble) à Valence, 1er avril 1791; quitta Auxonne le 14 juin; arriva à Valence le 16 et fut versé à la 1re compagnie (La Cattonne) du 2e bataillon de ce régiment; prêta serment à la Constitution de 1791 le 6 juillet; se fit inscrire à la société des Amis de la Constitution de Valence et y devint bibliothécaire; obtint un congé de semestre et s'embarqua pour la Corse, septembre 1791; arriva à Ajaccio; vint à Corté en février 1792; adjudant-major au bataillon des gardes nationales volontaires d'Ajaccio, 22 février 1792; élu lieutenant-colonel en 2e du 2e bataillon de volontaires de la Corse, 1er avril 1792; arriva à Paris le 28 mai; fut nommé le 10 juillet 1792 capitaine en 2e à l'ancienneté au 4e régiment d'artillerie à la date du 6 février 1792; ramena sa sœur en Corse en septembre 1792; s'embarqua à Toulon, 10 octobre; débarqua à Ajaccio le 15 et reprit le commandement de son bataillon. S'embarqua le 18 février 1793 pour l'expédition de l'île de la Madeleine sous le colonel Colonna-Cesari; commanda l'artillerie et les volontaires, dirigea le bombardement de l'île; puis se réembarqua précipitamment pour la Corse, 25 février; se déclara contre Paoli; fut arrêté à Bocognano par des paysans, s'échappa et se cacha dans les faubourgs d'Ajaccio; s'embarqua à la tour de Capitello, puis gagna par mer Bastia où il rejoignit les commissaires de la Convention; partici-

(1) Il ne faut pas chercher ici une biographie de Napoléon; on a volontairement laissé de côté tout ce qui est du domaine public, ses victoires, son gouvernement comme consul et comme empereur, sa vie privée, sa chute et sa captivité. On a exposé uniquement son curriculum vitae jusqu'à son arrivée au pouvoir en suivant en cela l'exemple et le livre de Chuquet, la Jeunesse de Napoléon, qu'on a prolongé jusqu'au 18 brumaire.

Figure 1: Sample page from Six's Dictionnaire; this page shows the beginning of Bonaparte's entry

A few basic observations about the text:

- Each page of the text contains two single-spaced columns of text, with a header at the top showing three-letter abbreviations of the last names of the generals whose entries appear on that page along with the page number.
- Each new entry is set apart from the previous one by a blank line, and the name of the general or admiral is given in bold at the beginning of the entry.
- After the first line of the entry, the data are presented in a series of semicolon-separated clauses, with each clause containing a specific piece of information about the individual's life or military career.
- The text contains occasional—very occasional—footnotes, which are indicated by numbers in parentheses and appear at the bottom of the respective column. These footnotes sometimes spill over across columns and, if need be, across pages. The footnotes contain additional information about the individual, but they are not essential to the main biographical narrative and are not part of the primary extraction process.

There are, across two volumes, approximately 1,200 pages of text so written, containing biographical entries for 2,117 generals and admirals.

Pipeline overview

The data pipeline for digitizing and analyzing Six's *Dictionnaire* consists of the following stages:

1. **Digitization**: The original text was scanned and converted into a machine-readable format using optical character recognition (OCR) software. The output is a set of Markdown files, one per page, containing the raw text along with metadata.
2. **Page-level cleaning**: The raw OCR output was cleaned to remove artifacts introduced by the scanning and recognition process. This involved stripping residual HTML markup, collapsing whitespace left by the two-column layout, correcting systematic OCR misreadings, and normalizing inconsistent typography (ordinals, degree symbols, and similar).
3. **Entry assembly**: The cleaned pages were processed to identify individual biographical entries and assemble them into complete records. Because entries sometimes span page boundaries, this stage involves both splitting pages at entry breaks and stitching partial entries across consecutive pages.
4. **Entry enrichment**: The assembled records were enriched with four layers of scholarly metadata: a provisional-status flag derived from Six's typographic conventions; errata tracking that links Six's own published corrections to the affected entries; cross-reference consolidation that records all known name variants for each individual in a single record; and errata application, in which Six's published corrections (substitutions, deletions, insertions, and reorganizations) are applied directly to each entry's text, with the original text preserved for provenance.
5. **Clause parsing and fact extraction**: Each biographical entry was segmented into its constituent clauses (Six's semicolons serve as the primary delimiter) and each clause classified by type — rank promotion, unit assignment, campaign participation, honor, injury, administrative event, and so on — with type-specific fields extracted into structured records. The classification

is rule-based, relying on pattern matching against Six's consistent vocabulary and syntax, ensuring full reproducibility.

6. **Gold enrichment**: Two post-parsing passes add derived fields: date propagation resolves year ellipses across consecutive clauses (e.g., when Six omits the year because it is the same as the preceding clause), and father-military detection flags entries where Six records the general's father as a military officer.
7. **Tabular export**: The enriched JSON is flattened into a single rectangular CSV with one row per clause (88,282 rows × 52 columns), for use in analysis notebooks.

The pipeline transforms Six's biographical prose — 88,282 clauses spanning three military eras — into structured data suited to quantitative analysis of careers, promotion dynamics, and institutional change.

We now discuss each stage of the pipeline in more detail, along with the challenges encountered and the solutions implemented to address them.

Digitization

The OCR was executed in Marker, a Python library especially well-suited for determining page layouts, dealing with multi-column text, ignoring headers, and generally producing clean output from scanned documents. Marker was used to process the scanned images of the two volumes of Six's *Dictionnaire*, resulting in a set of Markdown files, one per page, containing the raw text in machine-readable form.

A sample of the raw OCR output for a single page is shown below:

```python
from pathlib import Path

p = Path("../../data/bronze/pages/v1_page_0143.md")
print(p.read_text(encoding="utf-8"))
```

```
10 août 1804; grand aigle de la Légion d'honneur, 2 février 1805; général de
division et commandant de l'armée de Naples comme lieutenant de l'Empereur, 3
janvier 1806; roi de Naples et des Deux-Siciles, 31 mars 1806; puis roi
d'Espagne, 6 juin 1808; abdiqua la couronne des Deux-Siciles, 8 juillet, entra
à Madrid, 20 juillet; fut vaincu à Vittoria, 21 juin 1813; dut quitter
l'Espagne et se retira à Mortefontaine; nommé par Napoléon lieutenant général
de l'Empire, 28 janvier 1814 et chargé en cette qualité du commandement de la
garde nationale de Paris, de celles de la 1re division militaire, ainsi que
des troupes de ligne et de la garde impériale qui y étaient stationnées;
quitta Paris pour Blois avec l'Impératrice, 30 mars; invité à quitter la
France par le gouvernement de la Restauration, 16 avril; quitta Orléans le 18
avril et se retira en Suisse à Prangins; rentra en France aux Cent-Jours; pair
de France, 2 juin 1815; présida le conseil des ministres pendant l'absence de
l'Empereur; s'embarqua à Rochefort pour l'Amérique après Waterloo et s'établit
à Philadelphie sous le nom de comte de Survilliers, passa en Angleterre, 1832,
puis en Amérique, 1837-1839; autorisé à séjourner à Florence, 1841.
```

NAPOLÉON BONAPARTE ou de BUO-NAPARTE (1), empereur des Français, né à Ajaccio le 15 août 1769, mort à Longwood (île de Sainte-Hélène) le 5 mai 1821. Fit ses premières études à Ajaccio dans une classe tenue par l'abbé Recco; s'embarqua avec son père pour Marseille le 15 décembre 1778; entra avec son frère Joseph au collège d'Autun, 1er janvier 1779; nommé élève à l'Ecole militaire de Tiron, puis de Brienne-le-Château, 23 janvier 1779; partit d'Autun le 21 avril; entra à l'Ecole de Brienne le 14 ou le 15 mai 1779; nommé cadet gentilhomme à l'Ecole militaire de Paris, 22 octobre 1784; partit de Brienne le 30 octobre; nommé au concours le 42e sur 58, lieutenant en 2e d'artillerie au régiment de La Fère en garnison à Valence, 1er septembre 1785; quitta l'Ecole militaire le 28

octobre; partit pour Valence, 30 octobre; fut versé à la compagnie de bombardiers de la 5º brigade, capitaine Masson d'Autume, puis de la Gohyère, 1787, et de Coquebert, 1788-1789; envoyé à Lyon pour réprimer l'émeute « des deux sous », 14 août 1786; partit en Corse en congé de semestre, 1er septembre 1786; arriva à Ajaccio le 15 septembre; obtint pour raisons de santé un congé de 5 mois et demi avec appointements du 16 mai au 1er décembre 1787; s'embarqua pour la France, 12 septembre 1787; visita Versailles et Paris; obtint une prolongation de congé du 1er décembre 1787 au 1er juin 1788; arriva en Corse le 1er janvier 1788; s'embarqua pour la France le 1er juin; rejoignit son corps à Auxonne; chargé de réprimer une émeute à Seurre, 1er avril 1789; revint à Auxonne le 29 mai; eut à prendre part à la répression d'une autre émeute à Auxonne, 19 juillet; obtint un congé de semestre le 21 août; partit pour la Corse, arriva à Ajaccio, fin septembre 1789; obtint de nouveau un congé de 4 mois avec appointements, 15 juin 1790; se mêla à l'agitation en Corse; retourna à Auxonne; fut nommé lieutenant en 1er au 4º régiment d'artillerie (ci-devant Grenoble) à Valence, 1er avril 1791; quitta Auxonne le 14 juin; arriva à Valence le 16 et fut versé à la 1<sup>re</sup> compagnie (La Cattonne) du 2e bataillon de ce régiment; prêta serment à la Constitution de 1791 le 6 juillet; se fit inscrire à la société des Amis de la Constitution de Valence et y devint bibliothécaire; obtint un congé de semestre et s'embarqua pour la Corse, septembre 1791; arriva à Ajaccio; vint à Corté en février 1792; adjudant-major au bataillon des gardes nationales volontaires d'Ajaccio, 22 février 1792; élu lieutenant-colonel en 2e du 2e bataillon de volontaires de la Corse, 1er avril 1792; arriva à Paris le 28 mai; fut nommé le 10 juillet 1792 capitaine en 2e à l'ancienneté au 4e régiment d'artillerie à la date du 6 février 1792; ramena sa sœur en Corse en septembre 1792; s'embarqua à Toulon, 10 octobre; débarqua à Ajaccio le 15 et reprit le commandement de son bataillon. S'embarqua le 18 février 1793 pour l'expédition de l'île de la Madeleine sous le colonel Colonna-Cesari; commanda l'artillerie et les volontaires, dirigea le bombardement de l'île; puis se réembarqua précipitamment pour la Corse, 25 février; se déclara contre Paoli; fut arrêté à Bocognano par des paysans, s'échappa et se cacha dans les faubourgs d'Ajaccio; s'embarqua à la tour de Capitello, puis gagna par mer Bastia où il rejoignit les commissaires de la Convention; partici-

```
<sup>(1)</sup> Il ne faut pas chercher ici une biographie de Napoléon; on a
volontairement laissé de côté tout ce qui est du domaine public, ses
victoires, son gouvernement comme consul et comme empereur, sa vie privée, sa
chute et sa captivité. On a exposé uniquement son curriculum vitae jusqu'à son
arrivée au pouvoir en suivant en cela l'exemple et le livre de Chuquet, la
Jeunesse de Napoléon, qu'on a prolongé jusqu'au 18 brumaire.
```

This output contains the raw text of the page, along with some residual HTML markup and artifacts from the OCR process, such as misread characters and inconsistent whitespace. However, cursory inspection reveals that the OCR output is generally quite clean and faithful to the original text, with only a few systematic issues that can be addressed in the next stage of the pipeline.

We refer to this raw OCR output as the "Bronze Pages" stage of the pipeline, as it represents the initial digitized form of the text before any cleaning or structuring has been applied.

Page-level cleaning

Now that we have the raw OCR output in the form of the Bronze Pages, the next step is to clean this text to remove artifacts and inconsistencies that would interfere with downstream processing.

The problem

Six's *Dictionnaire* was printed in a dense two-column format with small type, and the surviving copies are of poor scan quality. The OCR engine (Marker, using the Surya recognition model) does a remarkably good job with the French text— diacritics come through largely intact, names and dates are accurate, and critically, Six's semicolon-delimited clause structure is well preserved. But the output contains systematic artifacts that would interfere with downstream parsing if left uncorrected.

The artifacts fall into three broad categories.

Markup residue. Marker produces Markdown output and occasionally emits raw HTML or Markdown formatting that reflects the original typography but does not belong in plain text. The most common example is `<sup>e</sup>` for superscripted ordinal suffixes (as in *15e de bataille*), which appears dozens of times per volume. Less frequently, bold markers (`**DEBRUN**`), Markdown heading syntax (`## BERNARD`), and image reference tags (`![](_page_0_Picture_7.jpeg)`) survive into the output. Running headers—short alphabetical guide strings like `BEX` or `CAU` that appear at the top of dictionary pages—also persist on some pages.

Layout artifacts from the two-column format. Marker processes the two columns of each page sequentially, but does not always handle the transition between them cleanly. The most pervasive artifact is a spurious blank line inserted at the column boundary, mid-sentence, which would cause downstream parsers to mistake the middle of an entry for an entry break. In some cases a word is also hyphenated across the column break, producing fragments like `com-` at the end of one column and `mandant` at the start of the next.

Inconsistent rendering of ordinal numbers. French ordinal abbreviations (*1er*, *2e*, *32e*, etc.) are rendered in superscript in the original text, and Marker handles them inconsistently. Sometimes they come through correctly as plain text (`1er`); sometimes they appear as HTML superscripts (`15<sup>e</sup>`); sometimes the superscript is replaced by a degree symbol (`32°`) or masculine ordinal indicator (`45º`); and sometimes, particularly for `1er` and `1re`, the superscript is misread as a straight double quote (`1"`). All four renderings represent the same typographical feature in the source and need to be normalized to a single consistent form.

A smaller number of artifacts arise from **failed dehyphenation**—compound words like *état-major* and *lieutenant-colonel* that are hyphenated across line breaks within a column and then rejoined without the hyphen, producing `étatmajor` or `lieutenantcolonel`. These are infrequent (roughly 40 instances across a 100-page sample) but would cause problems for any rule that relies on matching military vocabulary.

Methodology

We followed a sample-diagnose-build-validate cycle. First, we manually inspected a small handful of pages selected from both volumes to develop a qualitative catalog of error types. We then drew a random sample of 100 pages (using a fixed seed for reproducibility) and ran regex-based diagnostics to measure the prevalence of each pattern. This quantitative pass confirmed the patterns identified in manual inspection, surfaced a few additional low-frequency issues, and gave us confidence that no major error class had been missed.

Based on the diagnostic results, we wrote a deterministic cleaning script consisting of ten rules applied in a fixed order. Each rule addresses a single artifact type, is implemented as a standalone function, and is independently testable. The rules are applied to every page; no page-specific logic is used.

After running the script across the full corpus, we drew a second 100-page sample from the cleaned output (using the same seed, yielding the same pages for direct comparison) and ran a validation pass confirming that all targeted artifacts had been eliminated and all legitimate data had been preserved.

Cleaning rules

The ten rules are applied in the following order. The order matters: for example, footnote markers must be extracted while `<sup>` tags are still present, and column breaks must be collapsed before ordinal normalization to avoid operating on text that is split across a spurious line break.

1. Extract footnotes. Six includes occasional scholarly footnotes—nine across both volumes—marked with superscripted numbers in the text and corresponding note blocks at the bottom of the page. Because the in-text markers use `<sup>(1)</sup>` syntax that would be destroyed by later tag stripping, footnote extraction runs first. The markers are removed from the biographical text and the note bodies are extracted to a separate JSON file for manual review. The footnotes contain editorial commentary (source credibility assessments, terminological clarifications, corrections) that may be useful as metadata but do not belong in the structured biographical records.

2. Fix misread superscript ordinals. Marker sometimes misreads the content inside superscript tags, producing `<sup>st</sup>`, `<sup>et</sup>`, or `<sup>c</sup>` instead of the correct `<sup>er</sup>` or `<sup>e</sup>`. If these were simply stripped (as in Rule 4), the result would be nonsensical forms like 1st or 3et. This rule detects any `<sup>` tag after a digit whose content is not already correct (`e`, `er`, or `re`) and normalizes it using the same context-sensitive `1er`/`1re` logic described in Rule 9. The rule is deliberately permissive—it catches any non-standard inner text rather than listing known misreadings—so that variants not present in the sample are handled automatically.

3. Fix long s. The archaic long s character (`ſ`, U+017F) occasionally appears where Marker misreads a `1` in an ordinal context, producing `ſer` instead of `1er`. This rule replaces `ſer` and `ſre` at word boundaries with the correct ordinal forms, then converts any remaining long s characters to a regular s as a safe fallback.

4. Strip HTML superscript tags. Removes `<sup>...</sup>` tags while preserving their inner text, so that `15<sup>e</sup>` becomes `15e` and `1<sup>re</sup>` becomes `1re`.

5. Strip Markdown bold markers. Removes `**...**` while preserving the inner text. Affects a small number of entry headers where Marker rendered the name in bold.

6. Strip other Markdown artifacts. Removes `##` heading markers (preserving the text) and `![]()` image reference tags (discarding them entirely).

7. Strip running headers. Removes short (≤6 character) all-caps alphabetical guide strings that appear alone on the first line of some pages.

8. Collapse column breaks. This is the most consequential structural fix. Blank lines followed by lowercase text or digits are identified as column-break artifacts and collapsed to a single space. Words hyphenated across column breaks (`com-` / `mandant`) are rejoined by removing the hyphen and joining the fragments. Blank lines followed by uppercase text are preserved, since these are legitimate entry boundaries between biographical records.

9. Normalize ordinal numbers. All variant renderings of French ordinal suffixes are normalized to a consistent plain-text form: `32°` and `32º` both become `32e`; `1"` becomes `1er` or `1re` depending on context. The `1er`/`1re` distinction is handled by a context-sensitive rule that checks whether the following word is a known feminine noun (*division*, *brigade*, *légion*, *compagnie*, *subdivision*, *demi-brigade*). This preserves the grammatically correct French form rather than defaulting to the masculine.

10. Restore dehyphenation failures. A dictionary of eleven known compound words (`état-major`, `lieutenant-colonel`, `sous-lieutenant`, `arrière-garde`, etc.) that are systematically broken by OCR line-break handling is used to restore the correct hyphenated form.

What was intentionally preserved

Not all unusual features in the text are errors.

- **Title-case entry names** (e.g., `Argod (François)` rather than `ARGOD (François)`) indicate provisional generals whose rank was never formally confirmed. This is a meaningful distinction in Six's typographic system and is preserved as-is.

- **Cross-reference entries** (`COUSIN DE DOMMARTIN (Elzéar-Auguste). Cf.  Dommartin.`) are short "see also" pointers, not biographical records. They are left intact for identification at the entry assembly stage.
- **German place names** containing umlauts (Königssegg, Möckern, Göspich) are legitimate and are not modified by the ordinal normalization rules, which operate only on digit+symbol sequences.

Known limitations

- **Footnote spillover.** Two or three of the nine footnotes span page boundaries. The extraction rule captures footnote bodies that begin on the same page as their marker, but continuation text on the following page is not automatically detected. These were corrected manually.
- **Bare footnote markers.** Where the original text uses `(1)` without superscript tags, the in-text marker cannot be reliably distinguished from parenthetical content (such as years in parentheses). A small number of bare `(1)` references remain in the text and are handled at the entry assembly stage.
- **Isolated OCR errors.** A single instance of a Greek alpha character (α) appearing in place of the letters `si` (producing `diviα sionnaire` instead of `divisionnaire`) was identified in the sample. One-off character errors at this frequency are not worth building rules for; they are corrected if encountered during downstream processing.

Validation results

The cleaning script was validated against a 100-page random sample (seed=42) drawn from the cleaned output. Diagnostic checks confirmed:

- Zero remaining `<sup>` tags, bold markers, Markdown artifacts, or image references
- Zero remaining column-break artifacts (blank lines followed by lowercase text)
- Zero remaining degree symbols or masculine ordinal indicators after digits
- Zero remaining known dehyphenation failures
- All cross-reference entries preserved (31 in sample)
- All entry headers preserved (167 in sample)
- All German place-name umlauts preserved (5 in sample)
- Semicolon clause delimiters untouched (7,105 in sample)

Across the full corpus of 1,193 Bronze Pages, 1,188 were impacted by at least one cleaning rule.

We consider the automated cleaning stage complete and its output stable. The cleaning rules are designed to be conservative and reversible, so that any issues that arise in downstream stages can be traced back and addressed without loss of original data.

Diagnostic-driven silver page corrections

Running the entry assembler against the cleaned Silver Pages exposed a small number of structural defects that the automated cleaning rules could not address because they require page-specific knowledge. These defects were identified through a diagnostic script that flagged assembled entries with anomalous endings (e.g., entries truncated mid-sentence or ending with a stray letter fragment rather than a period), and then traced back to the underlying Silver Pages.

Two-column OCR interleaving. In a small number of pages, Marker reads across the two columns line-by-line rather than completing one column before starting the other, producing paragraphs where alternating lines belong to different columns. The automated column-break rule (Rule 8) correctly handles the common case — a blank line mid-sentence where the two columns are read sequentially — but cannot detect interleaving, which produces syntactically plausible but semantically garbled text. A dedicated script (`silver_page_fixes.py`) identified and corrected 22 pages with garbled all-caps entry headers caused by this artifact.

Missing surnames at page breaks. For long entries that span page boundaries, the surname appears at the bottom of one page and the parenthetical first-name block appears at the top of the next. In a small number of cases the OCR for the continuation page begins with the parenthetical alone — no surname — because the surname landed on the final line of the previous page's right column, which was not captured. One such case was identified: `v2_page_0556` began the VEDEL entry with `(Dominique-Honoré-Antoine-Marie, comte de), général, né à Monaco...` without its surname, causing the assembler to absorb the entire VEDEL record as a fragment of the preceding VEAUX entry. The surname `VEDEL` was prepended manually, and a missing terminal period was added to the concluding page (`v2_page_0557`).

Structural defects in entry headers. A second diagnostic pass against the assembled Bronze Generals identified further silver-page defects where entries were not being detected at all — their text was silently absorbed into the preceding record. These fell into three sub-types:

- *Missing surname before a parenthetical.* Several entries began with a bare parenthetical `(firstnames...)` because the surname had been placed on the previous column's final line and was absent from the continuation page. Each required prepending the correct surname. Cases: NAPOLÉON BONAPARTE (`v1_page_0143`), SIMIEN `(François-Martin-Valentin, baron)` (`v2_page_0477`), CAMBACÉRÈS (`v1_page_0207`), SENARMONT père (`v2_page_0465`).
- *Surname and parenthetical split across lines.* In one case, AVRANGE D'HAUGERANVILLE (`v1_page_0061`), the surname appeared on one line and the cross-reference parenthetical on the next, which the detection regex could not span. The two lines were merged into one.
- *Running-header artifact mid-entry.* On `v2_page_0279`, a three-letter running header (`NOG`) was embedded in the body of the NOGUÈS entry, splitting it in two. The duplicate text block was removed.
- *Entries with no parenthetical firstnames.* Two entries lacked the parenthetical structure the detector requires. TELT (`v2_page_0507`) was a short redirect entry; TELT's parenthetical was added. NAPOLÉON BONAPARTE (see above) similarly required a parenthetical.

Column-break continuation misordering. On `v2_page_0180`, the continuation of MARTEL (Philippe-André)'s entry — text beginning `pitaine, 17 août 1796;` — appeared at the top of the silver page, before MARTEL (Philippe-André)'s own entry header. This occurred because the OCR placed the right-column continuation fragment ahead of the entry header in the page output. The stitcher consequently appended the continuation to the preceding MARTEL aîné entry, leaving

MARTEL (Philippe-André) truncated at the column-break hyphen fragment `ca-`. The page was restructured to present a single, complete MARTEL (Philippe-André) entry with the hyphen resolved to `capitaine`.

In total, 31 Silver Pages received targeted manual corrections as a result of entry-assembly diagnostics. All corrections are recorded in `silver_page_fixes.py` (for the 22 header cases) and as inline comments in the affected page files (for the structural cases). The corrected pages were re-ingested through the assembler before generating the Bronze Generals output.

Entry assembly

The 1,193 resulting Silver Pages contain approximately 2,200 biographical entries, but entries do not respect page boundaries. Most pages (93%) begin mid-entry, and roughly half of all entries span two or more pages. This stage identifies entry boundaries within each page and stitches text fragments across consecutive pages to produce one complete record per general or admiral.

Entry boundary detection

Six's entries follow a consistent formula: a surname, a parenthetical with first names and titles, a rank keyword, and then the biographical text. For example:

```
ABBÉ (Louis-Jean-Nicolas, baron), général, né à Trépail (Marne) le 28 août
1764...
```

Entry boundaries are detected by a two-pass regex strategy:

1. **ALL-CAPS pass (high confidence):** Matches surnames rendered in uppercase followed by a parenthetical — e.g., `ABBÉ (Louis-Jean-Nicolas, baron)`. The pattern allows up to 60 characters of intervening title text between the surname and the parenthetical, handling entries like `AIGUILLON, duc d'AGENOIS (Armand-Désiré...)` where a noble title separates the two. It also handles apostrophe-prefixed surnames (`O'SHEE`, `D'AUVARRE`, `L'HERMITE`). This pass captures the vast majority of entries.

2. **Mixed-case pass (medium confidence):** Matches names in mixed case followed by a parenthetical *and* a rank keyword (`général`, `maréchal`, `amiral`, etc.). This catches entries where OCR failed to preserve the original small-caps formatting — e.g., `Agé (Pierre), général`. This second pass recovers 101 entries (4.5% of full entries) that the first pass misses entirely.

Both passes include filters for known false positives: Roman numerals (`III`, `VIII`), French stopwords (`LE`, `DE`), and body-text parentheticals containing dates or regiment descriptions. When both passes detect the same entry within 50 characters *and* no paragraph break (`\n\n`) separates them, the detections are deduplicated, keeping the earlier and more complete match. The paragraph-break guard prevents legitimate adjacent entries — for example a one-line cross-reference immediately followed by the next general — from being incorrectly merged.

Cross-references — short redirect entries like `BACHARETIE DE BEAUPUY (Michel-Armand de). Cf. Beaupuy.` — are detected at entry-start time by the presence of `. Cf.` immediately after the parenthetical. A second, post-assembly check (`CROSS_REF_FULL`) catches the subset of cross-references whose surnames contain commas (e.g., `COUSIN DE DOMMARTIN, dit X`) and therefore do not match the inline detection pattern: any assembled entry shorter than 200 characters that ends with a `. Cf. <target>.` pattern is reclassified as a cross-reference.

Diagnostics

Before writing the stitcher, a diagnostic script (`entry_boundary_diagnostics.py`) was run against all 1,193 Silver Pages to validate the detection patterns and characterize the corpus structure. Key findings:

Metric	Value
Total pages scanned	1,193
Pages starting mid-entry (spillover)	93.4%
Pages with zero detected entries	65 (5.4%)
Entries per page: mode	2 (38.3% of pages)
Entries per page: range	0–6
Mixed-case detections (not duplicating an ALL-CAPS match)	121
False positives identified (body-text matches)	~3
Consecutive zero-entry pages (entries spanning 3+ pages)	3 runs

The 65 zero-entry pages fall into three categories: 41 isolated pages (a single entry spanning two full pages), 20 blank end-matter pages in volume 1, and two small clusters in volume 2 where a single entry spans three consecutive pages.

Stitching algorithm

The stitcher (`silver_pages_to_bronze_generals.py`) walks all Silver Pages in volume-then-page order and applies a simple sequential accumulation:

1. For each page, find all entry-start positions using the two-pass detection.
2. Any text *before* the first entry start on the page is appended to the previously accumulating entry (this is the spillover join).
3. Each entry start begins a new record. Text runs from that start to the next entry start on the same page, or to the end of the page if no further starts exist.
4. When the next page is processed, the cycle repeats: leading spillover text is appended to the current entry, and new starts trigger new records.
5. Before writing each completed entry, a `clean_entry_text()` pass removes residual artifacts that survive into the assembled text: page-number bleeds (e.g., `- 312 -` absorbed into a trailing sentence); tail-leak fragments (any content after a `\n\n` that contains no period — partial headers of the following entry that slipped past detection); cross-reference tail fragments (any `\n\n`-separated line ending in a `Cf. <target>.` pattern); and missing terminal periods on Arc de Triomphe inscriptions. The tail-stripping passes run in a loop until stable, since removing

one fragment can expose another. After all fragment removal, any remaining mid-text `\n\n` or `\n` breaks — artifacts of page-join stitching — are collapsed to a single space.

End-matter pages are excluded from processing: volume 1 pages 639–660 (errata, addenda, colophon, and blanks) and volume 2 pages 600–608 (foreign generals, errata, and addenda). The errata content is preserved in the Silver Pages archive for application at a later stage.

Output

The stitcher produces a single JSON file (`bronze_generals.json`) containing 2,471 records:

Category	Count
Full biographical entries	2,208
Cross-references (Cf. redirects)	263
Total	**2,471**

Each record carries provenance metadata: the list of source pages (volume and page number) from which it was stitched, the detection method used, and whether it is a cross-reference.

Validation

Quality checks on the 2,208 full entries:

Check	Result
Rank keyword in first 200 characters	~96%
"né à" (birthplace) in first 500 characters	~87%
Contains semicolons (parseable into clauses)	~100%
Ends with period (clean ending)	100%
Ends with trailing name/letter fragment (strippable)	0%
Truly truncated (no final period, mid-sentence)	0%
Entries that swallowed another entry	0
Broken words at page joins	0

The `clean_entry_text()` post-assembly pass eliminates the trailing name and letter fragments that previously accounted for ~6% of entries. Combined with the targeted silver-page corrections described above, the period-ending rate reached 100% across all 2,208 full biographical entries.

Page-span distribution confirms the expected structure of the source text:

Pages spanned	Entries	Examples
1	1,128 (51.1%)	Typical short entries
2	1,047 (47.4%)	Most longer entries

Pages spanned	Entries	Examples
3	31 (1.4%)	Senior officers (Kléber, Murat)
4	2 (0.1%)	La Fayette (15,269 chars), Soult (10,669 chars)

The longest entries correspond to the most prominent figures in the dictionary: La Fayette, Marmont, Kléber, Soult, Oudinot, Ney, Murat, Clapérède, Masséna, and Gérard — all marshals or senior generals with extensive service records.

Entry enrichment

The Bronze Generals JSON is a faithful structural extraction from the source text, but it carries no interpretation. The enrichment stage adds three layers of scholarly metadata that make the Silver Generals a more useful scholarly object: a provisional-status flag, errata tracking, and cross-reference consolidation. All three are derived deterministically from signals already present in Six's text.

Provisional status

Six uses a deliberate typographic convention to signal that a general's rank was never formally confirmed: entries for provisional generals are set in small capitals or mixed case in the original text, while confirmed appointments appear in full capitals. The Bronze Generals assembler records a `detection_method` for each entry, but the enrichment stage adds a cleaner authoritative flag: `is_provisional` is `True` if and only if the entry's surname contains at least one alphabetic character that is not uppercase.

One edge case is worth noting: RIBOISIÈRE (as in BASTON DE LA RIBOISIÈRE) was detected by the assembler via the mixed-case pass, but the surname itself is entirely uppercase. The surname-based rule correctly returns `is_provisional=False` regardless of detection method.

Result: **99 provisional entries** out of 2,471 total.

Errata and addenda

Six published corrections in the back matter of both volumes: two pages at the end of volume 1 (pages 639–640) and three pages at the end of volume 2 (pages 606–608). The enrichment stage flags affected entries with `has_errata=True` and records the errata source identifiers and raw note text. The actual textual corrections — substitutions, additions, deletions indicated by *lire:*, *ajouter:*, *supprimer:* — are deferred to a subsequent corrections script, following the same separation used for silver-page fixes: the enrichment stage marks and preserves; a dedicated corrections script edits.

Parsing. The two formats differ. Volume 1 errata are plain-text lines of the form `SURNAME, p. N : instruction`; volume 2 errata are Markdown list items of the form `- Page N. SURNAME (FirstName), instruction`. Each line is parsed to extract a normalized surname key used for matching.

Matching. Each entry is matched against the errata index by normalized surname (accents stripped, uppercased). Three conditions handle the variant spellings that Six himself uses in his corrections:

1. Exact normalized match.
2. Word-prefix: the entry surname starts with the errata key at a word boundary, and the errata key is at least 6 characters (preventing short fragments like `BAR` from matching `BARBANÈGRE`).
3. Near-match: for surnames of 14 or more characters, entries differing from the errata key by at most 2 positions are considered a match. This handles OCR variants in the errata themselves, such as `CHLOPICKI DE NECZUIA` (the errata's spelling of what the main entry records as `CHLOPICKI DE NECZNIA`).

Firstname disambiguation. The volume 2 format includes the general's firstname in parentheses immediately after the surname: `ABBATUCCI (Jacques)`. When present, this firstname acts as an additional gate — the match is accepted only if the entry's `firstnames` field begins with (or is a prefix of) the errata firstname after normalization. The prefix rule rather than exact equality handles the common case where Six abbreviates a hyphenated name: `Jacques` correctly matches `Jacques-Pierre` but not `Jean-Charles`. Without this filter, both ABBATUCCI entries would be flagged; with it, only Jacques-Pierre receives `has_errata=True`.

Result	Count
Entries with `has_errata=True`	79
False positives eliminated by firstname filter	6
Distinct errata sources	5

Cross-reference consolidation

Six's dictionary contains 263 cross-reference entries — short redirects of the form `BACHARETIE DE BEAUPUY (Michel-Armand de). Cf. Beaupuy.` — that point readers from an alternate name or spelling to the main biographical entry. These are assembled as Bronze Generals records in their own right but carry no independent biographical content. The enrichment stage links each cross-reference to its target and records the alternate name in that target's `additional_names` list, so that all known name variants for a single individual are consolidated in one place.

Algorithmic matching. The target name in the `Cf.` clause is matched against non-cross-reference entries by normalized surname, in priority order: exact match, starts-with, and substring (for targets of 4 or more characters).

Manual overrides. Six's `Cf.` targets are informal references — abbreviated names, reversed compound names, particle variants (`d'/de/du`), phonetic spellings, and OCR errors — that often differ substantially from the actual headword. After the algorithmic pass, 26 cross-references remained unresolved. Each was researched individually. A representative sample of the resolutions:

Cross-reference	`Cf.` target	Main entry	Failure mode
ZAIONCZEK	*Zayon-check*	ZAYONCHEK	phonetic spelling
DELBHECQ	*Elbhccq*	ELBHECQ	OCR error in Six's errata
LE PELLEY - DUMANOIR	*Dumanoir-Le Pelley*	DUMANOIR LE PELLEY	reversed compound
BROC	*de Broc*	DEBROC	particle elision
AULTANNE	*Daultanne*	DAULTANE	particle/spelling variant
MONTHYON	*Bailly de Monthyon*	BAILLY DE MONTHION	reversed compound + spelling

Two entries — `DU CHILLEAU` and `VAN MARISY` — have no matching main entry in the dataset and remain unresolved; the underlying biographical records appear to be absent from the digitized text.

Result	Count
Cross-references resolved algorithmically	237
Cross-references resolved via manual override	24
Cross-references unresolvable (no main entry)	2
Entries with at least one `additional_names` record	238

Errata application

With affected entries flagged and the raw errata text recorded, a second script (`silver_generals_fixes.py`) applies the textual corrections directly to `silver_generals.json`. The enrichment stage (above) is intentionally read-only — it marks and preserves; this script edits.

Scope. Six's back matter contains approximately 100 individual correction operations, distributed across two formats:

- Volume 1 errata (pp. 639–640): 40 entries with corrections ranging from a single date substitution to a multi-step sequence of nine operations (SÉRAS).
- Volume 2 errata (pp. 606–608): 39 additional corrections, plus one unnamed note concerning the date of the Battle of Ocaña.

Not all corrections were matched by the enrichment stage's errata-tracking logic (short or particle-prefixed surnames can confound the matcher). An additional 11 entries that were missed by matching — AOUST, ARBONNEAU, SONGIS DES COURBONS, SOULT, TOUR-MAUBOURG, VALÉE, VARENNES, VAULTIER, VINCENT, WATRIN, and WEDEL — are included directly in the corrections table.

Operation types. The script encodes each correction as one of seven operation types:

Type	Description	Example
`subst`	Replace old text with new text	`26 mars 1790` ⟶ `26 mars 1799` (ABBATUCCI)
`delete`	Remove a phrase entirely	Delete `quitta le service de Naples en 1814` (AYMÉ)
`insert_after`	Insert text immediately after an anchor	After `Kreutznach, 1er décembre`: insert new division command (BERNADOTTE)
`insert_before`	Insert text immediately before an anchor	Before `commandant la 3e demi-brigade de vétéran` (MOURET)
`append`	Append a sentence to the end of the entry	`Fils d'un trésorier de France.` (D'ARBONNEAU)
`move_after`	Cut a phrase from one location and paste it after an anchor	Move `vainqueur à Peyrestortes` to follow `vainqueur au combat du Vernet` (D'AOUST)
`firstnames_subst`	Replace text in the `firstnames` field rather than `entry_text`	Remove the forename `Ange` from both the header and firstnames (HAUTPOUL)
`noop`	Typographic directive — no text change	`le nom de famille doit être imprimé en minuscules grasses` (NICOLAS)

Anchor-based matching. All operations locate their target by exact string match against `entry_text`. An `--inspect` mode (no writes) validates that every anchor is found before the `--apply` mode executes changes. This two-pass design catches mismatches — for example, one entry (LAURENT, 1424) was found to have no "juin 1796" in its text, indicating the enrichment stage had matched its errata note to the wrong LAURENT homonym; that entry was removed from the corrections table.

Disambiguation. Several surnames have two or more entries in the dictionary. For these, the errata-tracking stage flags all matches; the corrections table targets only the entry whose `entry_text` actually contains the anchor for the relevant correction.

Provenance preservation. For every entry that is modified, the original `entry_text` is saved as `pre_errata_entry_text` before any change is made. This field appears immediately before `entry_text` in the schema, providing a complete audit trail: the pre-correction text, the post-correction text, and the errata source and note that motivated the change.

Unnamed Ocaña note. One errata entry has no associated surname: *La bataille d'Ocana eut lieu le 19 novembre et non le 18 novembre 1809.* A separate pass identifies all 11 entries whose `entry_text` contains the string `18 novembre 1809` and corrects it to `19 novembre 1809`.

Results.

Metric	Value
Total operations encoded	115
Entries modified	90
Operations applied on initial run	115
Entries with `pre_errata_entry_text`	90
Entries with `has_errata=True` but no correction needed	1 (NICOLAS — typographic only)
Entries corrected despite no `has_errata` flag	11

Clause parsing and fact extraction

The Silver Generals JSON is a clean, errata-corrected, source-faithful transcription of Six's text. The Gold Generals JSON transforms it into structured biographical data: each entry's narrative is split into discrete clauses, each clause is classified by type, and type-specific fields are extracted into a predictable schema ready for database population.

Stages 5 and 6 of the pipeline (clause parsing and fact extraction) are implemented in a single script (`silver_generals_to_gold_generals.py`), because the two operations are tightly coupled: splitting a clause from a header and classifying it happen in the same pass. An intermediate "raw clauses" representation would add a file to maintain with no analytical payoff.

The script supports `--inspect` (print what would be extracted, no writes) and `--apply` (write the output JSON). The `--inspect --entry N` form prints the full gold record for a single entry, which was the primary tool for iterative development.

The structure of a Six biography

Six writes in a highly consistent style. A typical entry looks like this:

```
ABBÉ (Louis-Jean-Nicolas, baron), général, né à Trépail (Marne) le 28 août
1764; soldat au régiment de Hainaut, 1er juillet 1781; caporal, 20 août 1781;
[...]; mis à la retraite, 1er juin 1832; mourut à Châlons-sur-Marne, 9 avril
1834. Le nom du général Abbé est inscrit au côté Ouest de l'Arc de Triomphe.
```

The biographical content falls into two clearly separated zones:

1. **Header**: the preamble — surname, firstnames, optional rank title, birth information, and sometimes death information. This occupies part or all of the first semicolon-delimited chunk.
2. **Career clauses**: the sequence of semicolon-delimited biographical facts that constitute the career record proper.

The Arc de Triomphe inscription, when present, always appears as the final clause and is treated separately (see below).

Header extraction

The header is the biographical preamble containing the individual's name, optional rank title, and birth and death dates. Two structural patterns appear in the corpus:

Pattern A — transitional (~80% of entries). The first semicolon-delimited chunk contains both the biographical header *and* the first career clause, separated by a sentence-ending period followed by a capital letter. For example:

```
ABBATUCCI (Jacques-Pierre), né à Zicavo (Corse) le 7 septembre 1723; devint
capitaine de dragons dans la légion corse...
```

The comma after the parenthetical, the absence of a rank title between the `)` and `né à`, and the sentence-ending period before `devint` are all reliable signals. The split point is found by scanning for `. [A-Z]` (period + space + uppercase letter) while guarding against known abbreviations (`M.`, `St.`, `Dr.`, ordinal suffixes) and text inside parentheses.

Pattern B — clean header (~20% of entries). The entire first semicolon-delimited chunk is the header. No career clause follows the biographical sentence within that chunk:

```
ABBÉ (Louis-Jean-Nicolas, baron), général, né à Trépail (Marne) le 28 août
1764;
mourut à Châlons-sur-Marne, 9 avril 1834; soldat au régiment de Hainaut...
```

When no sentence-ending period followed by a capital is found in the first chunk, the entire chunk is treated as the header and the career record begins at the second semicolon-delimited chunk.

From the header, four fields are extracted:

- **Birth**: matched by `né à` / `naquit à` followed by a place name, optional department in parentheses, and a date phrase. Approximately 94% of non-cross-reference entries have parseable birth information in the header; the remaining ~6% lack the `né à` phrase entirely (common for foreign-born individuals or entries where the birth paragraph was not in Six's possession).
- **Death**: matched by `mourut` / `mort` / `y mourut` (same place as birth). Present in the header for approximately 43% of entries; a further portion have death information in body clauses classified as the `death` type rather than promoted to entry level.
- **Rank title**: the noun phrase appearing between the closing parenthesis of the firstnames block and the `né à` phrase — e.g., `général`, `général d'artillerie`, `Maréchal de France`, `marin`. Set to `null` when absent (roughly 15% of entries go straight from name to birth).

- **Family relation**: phrases like `frère du général X` or `fils du général Y` appearing in the descriptor region, extracted as a structured object with `relation` and `person_ref` fields.

Career clause splitting

After extracting the header, the remainder of the entry text is split on semicolons. The first career clause is either the fragment following the sentence boundary in Pattern A, or the first post-header semicolon chunk in Pattern B. Clauses are trimmed of whitespace; empty strings are discarded.

Compound clauses joined by *et*, *puis*, or *mais* within a single semicolon-delimited chunk are treated as atomic — they are not sub-split. Six's semicolons are the primary structural delimiter, and sub-splitting on conjunctions would introduce ambiguity about which date applies to which fact.

The Arc de Triomphe inscription is detected by a regular expression covering the standard formula (`Le nom du général X est inscrit au côté [direction] de l'Arc de Triomphe`) and its variants (entries using `maréchal`, `contre-amiral`, `Son nom est inscrit sur le côté`, multi-word surnames, and OCR artifacts like `estinscrit` without a space). Matching clauses are consumed and their direction recorded at the entry level rather than emitted as body clauses.

Date extraction

Every clause is scanned for a date, in the following priority order:

1. **Revolutionary calendar**: `N [month] An [roman]`, where the Revolutionary month name is one of the twelve (*vendémiaire* through *fructidor*). Converted to approximate Gregorian by computing the day offset from 22 September of year 1791+N. When Six provides a Gregorian equivalent in parentheses immediately after, that is used instead (and used to cross-check the conversion).
2. **Full Gregorian**: `D [month] YYYY`, capturing the last occurrence in the clause to prefer end-of-clause dates over incidentally mentioned earlier ones.
3. **Year range**: `YYYY–YYYY`, stored as `year` + `year_to` (see schema below).
4. **Month + year**: `[month] YYYY`.
5. **Year only**: last four-digit year in the clause matching `1[6–8]\d{2}`.

All date structures use a single canonical shape:

```
{"day": 7, "month": 9, "year": 1723, "year_to": null}
```

For year ranges, `year` holds the start year and `year_to` holds the end year; `day` and `month` are `null`. For partial dates, unknown components are `null`. The shape is uniform across all date types, which eliminates schema variation during database import.

`date_raw` always contains the date string exactly as it appears in the text. `date` is `null` when no date is found.

Clause classification

Each clause is classified into one of ten types using a first-match-wins rule cascade. The priority order constitutes the implicit ontology — when two rules could plausibly fire, the earlier one wins.

This order was established empirically and is stable for the current corpus; changing it requires a `parser_version` bump.

Priority	Type	Signal patterns
1	`arc_de_triomphe`	Arc de Triomphe inscription formula (consumed at entry level)
2	`death`	`mourut`, `décéda`, `mort à`
3	`injury`	`blessé`, `reçut [N] coups`, `tué dans`, `contusionné`
4	`honor`	`chevalier de/du`, `légion d'honneur`, `saint-louis`, `médaille`, `ordre de`, `baron/comte de l'Empire`, `sabre d'honneur`
5	`administrative`	`mis à la retraite`, `réformé`, `suspendu`, `arrêté`, `prisonnier de guerre`, `démissionna`, `rayé des contrôles`, `demi-solde`, and ~25 further patterns
6	`personal`	`fils de`, `était`, `fit ses études`, family/profession phrases
7	`rank_promotion`	Promotion verbs (`devint`, `fut nommé`, `promu`, `élu`) + rank term; direct rank-at-start; `avec rang de`; `confirmé par lettres`
8	`appointment`	`commandant de`, `aide de camp`, `chef d'état-major`, `prit le commandement`, `ministre de`, `président du/de la`
9	`battle`	`combat de`, `siège de`, `batalla du/des`, `s'empara de`, `combattit`, `couvrit la retraite`
10	`service`	`à l'armée de`, `servit à/dans`, `au corps d'`, `sous [name]`, `passé par amalgame`
11	heuristic	`à [Capitalized], [date]` ⟶ `battle`; bare `[Capitalized], [date]` (no verb, no earlier match) ⟶ `battle`
12	`unknown`	Fallback

Rank recognition (step 7) uses a list of ~50 rank terms ordered longest-first to prevent substring matches. Terms include the full Revolutionary and Napoleonic rank ladder from `maréchal de france` down through `adjudant-commandant`, `mestre de camp`, `maréchal des logis chef`, `capitaine-adjudant-major`, `quartier-maître trésorier`, and naval equivalents (`contre-amiral`, `vice-amiral`).

Type-specific field extraction

Each classified clause has a `details` object whose structure depends on the clause type. All clauses also carry `clause_text` (verbatim), `date_raw`, `date`, `clause_id`, and `transform_notes` (a list, empty for most clauses).

Type	`details` fields
`rank_promotion`	`to_rank` (str|null), `provisional` (bool), `by_whom` (str|null), `unit` (str|null)
`appointment`	`role` (str), `person_ref` (str|null), `unit_ref` (str|null)
`service`	`army` (str|null), `commander` (str|null)
`battle`	`location` (str|null), `outcome` (`"victory"`|`"defeat"`|null)
`honor`	`honor` (str), `honor_class` (str|null: `chevalier`|`officier`|`commandeur`|`grand-croix`)
`administrative`	`action` (`retraite`|`réforme`|`suspension`|`arrestation`|`liberté`|`disponibilité`)
`injury`	`location_geo` (str|null), `fatal` (bool)
`death`	`place` (str|null), `cause` (str|null)
`personal`, `unknown`	`{}` (empty)

Gold JSON schema

Each entry in `gold_generals.json` has the following top-level structure:

```
{
  "entry_index": 1,
  "parser_version": "gold_v1",
  "surname": "ABBATUCCI",
  "firstnames": "Jean-Charles",
  "is_crossref": false,
  "is_provisional": false,
  "has_errata": false,
  "additional_names": [],
  "birth": {
    "date_raw": "15 novembre 1770",
    "date": {"day": 15, "month": 11, "year": 1770, "year_to": null},
    "place": "Zicavo",
    "dept": null,
    "uncertain": false,
    "source": "header"
  },
  "death": null,
  "rank_title": "général",
  "family_relation": {"relation": "fils", "person_ref": "général Jacques-
Pierre"},
  "arc_de_triomphe": true,
  "arc_de_triomphe_side": "est",
  "clauses": [...]
}
```

`birth` and `death` are `null` when not found. Both carry `"source": "header"` to distinguish header-derived facts from any future promotion of body-clause data to entry level. Cross-reference entries carry `null` for all biographical fields and no `clauses` list.

Each clause record:

```
{
  "clause_index": 2,
  "clause_id": "1_2",
  "clause_text": "devint capitaine de dragons dans la légion corse avec rang
de lieutenant-colonel, 1er septembre 1769",
  "clause_type": "rank_promotion",
  "transform_notes": [],
  "date_raw": "1er septembre 1769",
  "date": {"day": 1, "month": 9, "year": 1769, "year_to": null},
  "details": {
    "to_rank": "capitaine de dragons",
    "provisional": false,
    "by_whom": null,
    "unit": null
  }
}
```

`clause_id` is `"{entry_index}_{clause_index}"` — a stable positional identifier that survives reruns as long as the splitting logic is unchanged. `parser_version` on the entry records which version of the classification rules produced the output; both fields together make it straightforward to diff outputs across parser iterations.

Results

Metric	Value
Total entries	2,471
Cross-references (no clause parsing)	263
Full entries parsed	2,208
Birth found in header	2,075 / 2,208 (94.0%)
Death found in header	950 / 2,208 (43.0%)
Arc de Triomphe inscriptions	618

Metric	Value
Total career clauses	88,465
Clauses with no date	22,655 (25.6%)
Clauses classified as `unknown`	11,836 (13.4%)

The clause type distribution across all 88,465 career clauses:

Type	Count	Share
`appointment`	18,591	21.0%
`service`	17,299	19.6%
`rank_promotion`	17,180	19.4%
`unknown`	11,836	13.4%
`administrative`	8,171	9.2%
`battle`	6,962	7.9%
`honor`	5,146	5.8%
`injury`	2,855	3.2%
`death`	221	0.2%
`personal`	204	0.2%

Known limitations

Birth parse rate. The 94% birth-found rate reflects the 6% of entries where Six does not include the standard `né à` formula — most commonly foreign-born individuals (Poles, Germans, Italians) for whom Six lacked reliable birth records, and a small number of entries where the biographical preamble is atypically structured.

Death parse rate. The 43% header-death rate is expected: a large proportion of deaths are recorded in body clauses (`mourut à X, le Y`) rather than in the header, because Six only includes death in the header when the individual had already died by the time of publication (1934). Living generals at publication time have no death entry at all. The 221 `death`-type body clauses capture a portion of these, but many deaths appear in clauses that also contain other information (final retirement followed by death in the same sentence) and are classified by the earlier-priority rule.

Unknown clauses (13.4%). The classification rules were developed iteratively against the full corpus, improving from an initial 41.5% unknown rate. The remaining 13.4% is genuine long-tail variation: no single unclassified pattern accounts for more than ~10 clauses. All `unknown` clauses retain their full `clause_text`, so no information is lost; further classification improvements can be applied in a future parser version without invalidating existing `clause_id` references.

Military school attendance

Military school clauses are consumed at the same point in the pipeline as Arc de Triomphe clauses — detected before `build_clause` is called and stored as an entry-level field rather than emitted into the clause sequence.

Detection. A clause is identified as a student-at-school clause when it matches any of the following patterns (case-insensitive, with tolerance for unaccented OCR variants such as `Elève` for `Élève`):

- *élève à / de l'école*, *élève au corps*, *élève du génie*, *élève d'artillerie*, *élève de la marine*
- *entré à l'école / entré comme élève*
- *admis à l'école / admis comme élève*
- *cadet gentilhomme à l'école* (note: *cadet gentilhomme au régiment* is deliberately excluded — regiment entry is a career event, not a school)
- *lieutenant en second à l'école / sous-lieutenant élève / élève sous-lieutenant*
- *reçu élève / nommé élève à l'école*

Clauses matching *commandant l'école* or *inspecteur général des écoles* are explicitly excluded: these are administrative appointments held later in a career, not student attendances.

Extraction. For each matched clause, the school name is extracted with a regex anchored at the keyword `école`, `collège`, or `corps royal`, stopping at the first comma, semicolon, or stop-word (`puis`, `et`, `en sortit`). When a single clause mentions multiple schools — for example, *Élève à l'École de Brienne, puis cadet gentilhomme à l'École militaire de Paris* — all names are extracted and stored as separate records. The entry year is taken from the clause's parsed date.

Output field. Each non-cross-reference entry gains:

```
"military_schools": [
  {"name": "Ecole de Brienne",          "year": 1782},
  {"name": "Ecole militaire de Paris",  "year": 1782}
]
```

Cross-reference entries carry `"military_schools": []`.

Results.

Metric	Value
Generals with ≥1 school record	157
Total school records	167
Clauses consumed (not emitted)	183
Career clauses after school extraction	88,282

The most common schools in the corpus (after light normalization):

School	Records
Écoles d’artillerie (La Fère, Châlons, Metz, Verdun, Bapaume…)	66
École royale du génie de Mézières	21
Collèges militaires (various)	10
École du génie (other locations)	9
École Militaire de Paris	7
École de Metz (combined artillery/génie)	7
École des Ponts et Chaussées	4
École militaire de Brienne	4
École de Mars (Revolutionary, 1794)	3
École Polytechnique / École Centrale des Travaux Publics	2

The distribution is historically coherent. The artillery and engineering schools dominate because Six’s cohort of generals was disproportionately technical — the pre-revolutionary *corps savants* (génie and artillerie) required formal schooling and produced a high fraction of the eventual generalate. The École Militaire de Paris and Brienne entries represent the noble-cadet track; the École de Mars and Polytechnique represent the Revolutionary meritocratic track.

Analytical significance. Military school attendance is a candidate variable for studying structured differences in pre-revolutionary career trajectories. The schools in Six’s corpus are heterogeneous — artillery schools at La Fère and Châlons, the engineering school at Mézières, the École Militaire de Paris, the Revolutionary École de Mars — with different admissions criteria and social compositions. Whether school attendance functioned as a proxy for privileged access to faster promotion tracks, or simply reflected technical specialization, is a question the data can gesture at but not resolve: the *en second* position records that would be the clearest signal of any fast-track mechanism are nearly absent from the corpus (≤1 occurrence in 17,180 rank-promotion clauses).

Gold enrichment

The enrichment stage (`gold_generals_enrich.py`) applies two additional enrichments to `gold_generals.json` in a single pass, producing `gold_generals_enriched.json`. The input JSON is never modified; the output is a parallel file with `"parser_version": "gold_v1_enriched"`.

Date propagation

Six’s writing frequently omits the year — or the year and month — when they are the same as the preceding clause. The gold parser processes each clause in isolation and cannot resolve these ellipses. The propagation pass walks each entry’s clauses in index order, maintaining a running context of the most recently seen year and month, and fills in missing date components.

Patterns handled:

Text pattern	Inferred action
Clause with month but no year	Fill year from context
même année / *l'an même*	Copy year from previous clause
l'année suivante / *l'an suivant*	Previous year + 1
l'année précédente / *l'an précédent*	Previous year − 1
au mois de [month] (no year in text)	Detect month, fill year from context
le [day] [month] (no year)	Detect day + month, fill year from context

Inferred components are recorded in `date["date_inferred"]` as a list of field names (e.g. `["year"]` or `["year", "month"]`). `date_raw` is never altered — all original parsed text is preserved.

Metric	Value
Total clauses	88,282
Null dates before propagation	22,638 (25.6%)
Null dates after propagation	21,990 (24.9%)
Dates with inferred components	648

The recovery rate (2.9% of null dates) reflects the conservative approach: a date is only filled when the clause text contains positive evidence of a date expression, even a partial one. Clauses with no date language whatsoever remain null.

Father-military flag

Whether a general's father was a professional military officer is a key variable for the pre-revolutionary track analysis. Six does not record this systematically, but it appears in four identifiable patterns:

Evidence type	Detection method	Confidence
`family_relation`	Entry-level field carries `relation: "fils"` + military title in `person_ref` (général, maréchal, colonel, capitaine, lieutenant, major, commandant, amiral…)	High
`clause_explicit`	Clause text contains *fils d'un officier de [marine/infanterie/…]*	High

Evidence type	Detection method	Confidence
`clause_survivance`	Clause text contains *en survivance de son père* (inherited a venal military office)	High
`clause_adc`	Clause text contains *aide de camp de son père* (implies father held general-officer rank)	High
`clause_commanded`	Clause text contains *commandé par son père* in a naval warship context (excluding commercial vessels)	Medium

The `family_relation = "fils"` path also produces `father_military: False` when the father's title is recognizably civilian (constituant, conventionnel, sénateur, marquis, duc without military service, etc.), providing a confirmed-non-military signal.

Three new entry-level fields are added:

```
"father_military":          true,
"father_military_evidence": "family_relation",
"father_military_note":     "général François-Marie d'Aboville"
```

Metric	Value
`father_military = True`	38
`father_military = False`	8
`father_military = None` (not detectable)	2,425

The 38 confirmed cases are a floor. Six's dictionary is about the generals, not their fathers; paternal military service is mentioned only when directly relevant to the career narrative (service as aide de camp to a father, inheritance of a military post, or explicit biographical note). The 2,425 `None` entries are genuinely unknown, not confirmed non-military.

Tabular export

`gold_generals_to_events_csv.py` flattens `gold_generals.json` into a single rectangular CSV (`gold_generals_events.csv`) with one row per clause. Entry-level fields are repeated on every row for that entry; clause-level and detail fields occupy their own columns.

Cross-reference entries (no clauses) produce no rows.

Output dimensions: 88,282 rows × 52 columns.

Column group	Columns	Notes
Entry identity	entry_index, surname, firstnames, rank_title	Repeated on every clause row
Entry flags	is_provisional, has_errata, arc_de_triomphe, arc_de_triomphe_side	
Family	family_relation, additional_names	additional_names pipe-joined if multiple
Birth	birth_date_raw, birth_year, birth_month, birth_day, birth_place, birth_dept, birth_uncertain	
Death	death_date_raw, death_year, death_month, death_day, death_place, death_cause, death_uncertain	
Clause identity	clause_id, clause_index, clause_type, transform_notes	
Clause date	date_raw, date_year, date_month, date_day, date_year_to	
Clause text	clause_text	Verbatim
Details	d_to_rank, d_provisional, d_by_whom, d_unit, d_role, d_person_ref, d_unit_ref, d_army, d_commander, d_location, d_outcome, d_honor, d_honor_class, d_action, d_location_geo, d_fatal, d_place, d_cause	null for inapplicable clause types

Multi-valued fields (additional_names, transform_notes) are pipe-joined (|) rather than expanded into multiple rows, to keep the row-per-clause invariant intact.

The military_schools field is not included in the events CSV because it is an entry-level list, not a per-clause fact. It is available in gold_generals.json and gold_generals_enriched.json.