Kerne.studio

— NORMALIZZAZIONE PER RAG

Documenti caotici trasformati in dati pronti per il retrieval.

08

Lo strato invisibile che fa funzionare ogni RAG: pulizia, deduplica, segmentazione semantica, metadati coerenti. Senza questo step, il tuo assistente AI risponde male — e tu pensi sia colpa del modello.

— IL PROBLEMA

RAG installato.
Risposte ancora pessime.

Tutti parlano di RAG. Pochi parlano del fatto che un RAG su dati sporchi è peggio di un assistente generico: cita fonti sbagliate, mescola contesti, allucina con sicurezza. La qualità del retrieval dipende al 70% dalla qualità del corpus.

70%

della qualità di un RAG è preparazione dei dati. Il modello e l'embedding contano per il 30%. Eppure quasi tutti partono dal 30%.

PDF disordinati

Header, footer, numeri di pagina, colonne — il testo estratto è ingestibile senza un livello di pulizia.

Duplicati ovunque

Lo stesso paragrafo in 5 documenti diversi. Il modello recupera tutti e 5 e ti dà 5 risposte uguali.

Chunk a caso

Spezzare un manuale ogni 1.000 caratteri è veloce. Ed è il motivo per cui le risposte tagliano a metà ragionamenti.

— LA SOLUZIONE

Il livello che gli altri saltano.

Costruiamo la pipeline di normalizzazione: estrazione strutturata da PDF/scansioni/email, pulizia, deduplica, segmentazione semantica (non a caratteri), metadati coerenti. Output: un corpus su cui il retrieval funziona davvero.

Pipeline · 5 step

01	**Estrazione**	*PDF, scansioni, .docx, email, XLS*
02	**Pulizia**	*header/footer/rumore OCR rimossi*
03	**Deduplica**	*contenuti ripetuti collassati*
04	**Chunking**	*su confini semantici, non lunghezza*
05	**Metadati**	*titolo, sezione, fonte, data, lingua*

Capisce la struttura

Distingue titoli, paragrafi, note, tabelle — anche su PDF disordinati con OCR.

Spezza con criterio

Chunking semantico: ogni segmento è auto-contenuto. Niente ragionamenti tagliati a metà.

Arricchisce di metadati

Titolo, sezione, fonte, data, lingua — abilita filtri e re-ranking nel retrieval.

— CAPABILITY

Sei modi di rendere il RAG utile.

01

Da caos a corpus

PDF, scansioni, email, fogli di calcolo, web — tutto diventa testo strutturato e indicizzabile.

02

Pulizia e deduplica

Header, footer, rumore OCR, paragrafi duplicati — rimossi prima dell'indicizzazione.

03

Segmentazione semantica

Chunk su confini di significato, non a 1.000 caratteri. Il retrieval recupera blocchi sensati.

04

Metadati coerenti

Titolo, sezione, fonte, data, autore, lingua — abilita filtri, re-ranking, citazioni puntuali.

05

Il livello che fa la differenza

Senza questo step le risposte degradano. Con questo step, il tuo RAG è davvero usabile.

06

Pipeline ripetibile

Integrabile nei tuoi processi di ingestion. Nuovi documenti entrano, vengono normalizzati, indicizzati.

— STESSA DOMANDA, DUE RAG DIVERSI

Prima e dopo la normalizzazione.

La stessa query, posta a due assistenti — stesso modello, stesso embedding, stesso vector DB. Solo i dati sono diversi.

QUERY *Qual è la procedura di reso prevista dal contratto di fornitura del 2022?*

RAG senza normalizzazione

Risposta: *"La procedura di reso prevede 30 giorni dalla consegna..."*

Problemi:

— Cita FAQ del sito, non il contratto

— Frammenti tagliati a metà frase

— Mescola condizioni di vendita B2C

RAG con normalizzazione

Risposta: *"Art. 12 del contratto 2022/Edil-Sirio: reso entro 15 gg con DDT..."*

Cosa cambia:

— Cita la fonte giusta (il contratto, non FAQ)

— Chunk semantico: paragrafo completo

— Metadati filtrano: solo 2022, no B2C

— COME FUNZIONA

Tre strati. Una pipeline ripetibile.

Niente magia, niente black box. La pipeline è documentata, ispezionabile, integrabile nei tuoi processi di ingestion.

01 · Ingestion

PDF · Email · Scansioni · KB

- Lettura nativa di formati eterogenei
- OCR su scansioni rumorose
- Crawler su intranet, SharePoint, sito

02 · Normalizzazione

Clean · Dedup · Chunk · Tag

- Pulizia struttura e rumore
- Deduplica e fingerprinting contenuti
- Chunking semantico + metadati ricchi

03 · Output

Corpus pronto · API

- Indicizzabile in qualsiasi vector DB
- Esportabile (JSON, Parquet, plug-in)
- Pipeline ripetibile via API o batch

— RISULTATI ATTESI

Cosa cambia, in numeri.

Stime su un corpus aziendale tipo (~10.000 documenti, PDF/email/KB), dopo applicazione della pipeline di normalizzazione.

+45%	−60%	3x	100%
Accuracy delle risposte	Allucinazioni e citazioni errate	Velocità di onboarding nuovi docs	Tracciabilità di ogni chunk
misurato su set di valutazione	*fonti sbagliate quasi eliminate*	*pipeline batch ripetibile*	*fonte, sezione, data, lingua*

— PERCHÉ KERNE.STUDIO

Il lavoro che nessuno vede. Quello che fa la differenza.

È più sexy parlare di modelli e benchmark. Il lavoro vero, però, è qui — nella pipeline che trasforma il caos aziendale in un corpus su cui un retrieval può funzionare. È meno glamour, ma è il livello che decide se il tuo RAG vale qualcosa.

Pipeline in 3–4 settimane

Analisi del corpus, definizione regole, prima ingestion normalizzata.

Sopra il tuo stack

Funziona con qualsiasi vector DB e modello — LangChain, LlamaIndex, custom.

Ripetibile, non one-shot

Pipeline automatizzata: nuovi documenti entrano, vengono normalizzati, indicizzati.

Deploy on-prem o EU

I dati aziendali non escono — la normalizzazione gira dove decide tu.

— COME PARTIAMO

Dal corpus alla pipeline in 4 settimane.

01

SETTIMANA 1

Audit

- Campionamento del corpus esistente
- Identifichiamo formati, rumori, duplicati
- Definizione del successo (KPI di retrieval)

02

SETTIMANA 2

Design

- Definizione regole di chunking semantico
- Schema metadati specifico per il tuo dominio
- Scelta vector DB e modello di embedding

03

SETTIMANA 3

Build

- Costruzione della pipeline end-to-end
- Prima ingestion completa del corpus
- Valutazione baseline vs pipeline

04

SETTIMANA 4

Hand-off

- Documentazione e training tecnico
- Integrazione nei tuoi processi di ingestion
- Supporto continuo per 4 settimane

Kerne.studio

— PARLIAMONE

Mandaci un campione del corpus. Ti mostriamo la differenza che fa.

PARLA CON NOI

hello@kerne.studio

kerne.studio

PROSSIMO PASSO

Benchmark di retrieval

100 docs · 20 query · prima vs dopo.