HelixTranslate

HelixTranslate

ترجمهٔ کتاب با مدل‌های تأییدشده — طراحی‌شده برای صداقت، هرگز جایگزین خاموش نیست.

خلاصه

HelixTranslate پلتفرم ترجمه کتاب الکترونیکی با کارایی بالا و مبتنی بر Go است که کتاب‌ها تأییدشده ترجمه می‌کند. این سیستم دارای LLMرا بین بیش از ۱۰۰ زبان با استفاده از ارائه‌دهندگان بلادرنگ و سیاست مسیریابی «عدم جایگزینی خاموش» است که به جای WebSocket نظارت افت بی‌صدا، با شکست آشکار مواجه می‌شود.

توضیح کوتاه

ابزار جامع ترجمه کتاب الکترونیکی مبتنی بر Go. فرمت‌های FB2، EPUB، TXT، HTML، PDF و DOCX را در بیش از ۱۰۰ زبان با استفاده از قوی‌ترین مدل‌های تأییدشده (از طریق LLM) ترجمه می‌کند. این سیستم از REST/HTTP-3 و gRPC، APIهای پل LLMsVerifier بلادرنگ بهره می‌برد. پردازش توزیع‌شده و داشبورد نظارت WebSocket.

توضیح مفصل

HelixTranslate یک سیستم سازمانی و مبتنی بر Go برای ترجمهٔ کامل کتاب‌ها بین زبان‌ها با LLM استفاده از ارائه‌دهندگان است. — نه پاراگراف‌ها یا قطعات کوتاه، بلکه آثار بلند از ابتدا تا انتها. این سیستم فرمت‌های مختلف کتاب الکترونیکی (FB2، EPUB، TXT، HTML، PDF، DOCX) را تجزیه و بازسازی می‌کند، از بیش از ۱۰۰ زبان با تشخیص خودکار پشتیبانی کرده و هم و جریان رویداد HTTP/3، gRPC بر بستر API (REST و هم سرورهای CLI ابزارهای WebSocket) را ارائه می‌دهد تا هم در گردش کار ترمینال و هم در معماری سرویس‌مش به‌خوبی جای گیرد.

است: به جای کدنویسی ثابت یک ارائه‌دهنده و امید نحوهٔ انتخاب مدل ویژگی تعیین‌کنندهٔ این سیستم LLMsVerifier (`pkg/bridge`) تمام اختیارات مدل‌را به پل HelixTranslate، به سلامت آن را انتخاب کرده وزنجیرهٔ جایگزین قطعی و تأییدشده APIواگذار می‌کند که قوی‌ترین مدل مرتب‌شده بر اساس امتیازرا بازمی‌گرداند. صلاحیت مدل بر اساس امتیاز وزنی در معیارهایی همچون پاسخگویی، کد، غنای ویژگی‌ها و قابلیت اطمینان تعیین می‌شود — بنابراین مدلی که ترجمهٔ شمارا انجام می‌دهد، جایگاه خودرا با اثبات کارایی‌اش به دست آورده است، نه با حضور در یک فایل پیکربندی.

نکتهٔ حیاتی این است که سیستم در کد خود اصل «عدم جایگزینی خاموش»را به‌طور جدی اعمال می‌کند: اگر کلید API ارائه‌دهنده موجود نباشد یا اپراتور به‌طور صریح درخواست ارائه‌دهندهٔ غیرقابل‌دسترس کند، خط لوله به جای تعویض بی‌صدای ارائه‌دهنده یا افت به‌زمان اجرا محلی و وانمود کردن اینکه همه چیز خوب است، با خطای سخت و آشکار مواجه می‌شود — قانونی که با یک دروازهٔ پیش‌ساخت اختصاصی و آزمون جهش جفت‌شده تثبیت شده است. زمان‌های اجرای محلی (Ollama، llama.cpp) به‌طور عمد از مسیر پیش‌فرض حذف شده‌اند تا هیچ موتور ضعیف‌تری نتواند به‌طور پنهانی جایگزین یک مدل تأییدشده شود.

CLI بلامرنگ قرار دارد: ابزار ترجمهٔ WebSocket در اطراف هستهٔ ترجمه، زیرسیستم نظارت رویدادهای تایپ‌شده‌را به سرور نظارت ارسال می‌کند که داشبورد وب‌زنده‌را هدایت می‌کند، در حالی که از راه دور بار کاری‌را برای ترجمهٔ توزیع‌شده پخش می‌کنند. بر روی این لایه‌ها، پرداخت SSH کارگران چندمرحله‌ای برای یکنواختی، تحلیل کیفیت در مرحلهٔ آماده‌سازی، کش ترجمه برای مدیریت هزینه در ورودی‌های طولانی و تضمین کیفیت مبتنی بر بینایی قرار گرفته است. کل پلتفرم بر اساس قانون اساسی مهندسی ضد-لافزنی عمل می‌کند: آزمون‌ها باید نتایج واقعی و قابل‌مشاهده برای کاربر را اثبات کنند و پشتوانهٔ آن‌ها آزمون جهش اجباری است، نه تیک‌های سبزی که هیچ چیزی‌را اثبات نمی‌کنند.

محتوا

چرا آن‌را ساختیم

هرگز ترجمه‌ای «ناقص اما موجود» ارائه — صادقانه برای ترجمهٔ کتاب‌های بلند به‌صورت قابل‌اعتماد و نکنیم. اصل طراحی این است که ترجمهٔ گم‌شده یا تأییدنشده باید خطایی بلند و سخت باشد، و انتخاب مدل همیشه باید به یک ارائه‌دهندهٔ واقعی و تأییدشده ختم شود، نه حدس از پیش تعیین‌شده یا جایگزینی خاموش محلی.

چرا انقلابی است

به‌صورت خاموش شکست می‌خورند — بی‌صدا به HelixTranslate بیشتر خطوط لولهٔ ترجمهٔ مدل ضعیف‌تر بازمی‌گردند، به محیط محلی می‌لغزند، یا خروجی ناقص تولید می‌کنند در حالی که این HelixTranslate. مجموعهٔ آزمون همچنان سبز می‌ماند و کسی متوجه سقوط کیفیت نمی‌شود حالت شکست‌را به‌کلی غیرممکن می‌سازد: انتخاب مدل دروازه‌دارِ تأیید است، زنجیرهٔ جایگزین قطعی و کاملاً شفاف است، و «نبود کلید / نبود مدل تأییدشده» به خطای سخت صادقانه ختم می‌شود، نه به بی‌تفاوتی خاموش. همین تصمیم طراحی، پرسش «آیا این ترجمه واقعاً با مدل توانمند و تأییدشده اجرا شده؟»را از امیدی که نمی‌توان بررسی کرد به ضمانتی تبدیل می‌کند که سیستم به‌جای شما اجرا می‌کند.

نوآوری‌ها

- تأییدشده قوی‌ترین مدل — LLMsVerifier از طریق پل **مسیردهی مدل دروازه‌دارِ تأیید** به‌طور خودکار انتخاب می‌شود، بنابراین اپراتورها نیّت خودرا اعلام می‌کنند، نه نام فروشنده، و هرگز ارائه‌دهنده‌ای‌را به‌صورت دستی انتخاب نمی‌کنند که ممکن است از دسترس خارج باشد.
- که در کد اعمال می‌شود — چهار شاخهٔ مسیریابی صریح **ضمانت عدم جایگزینی خاموش** (ماک / تأییدکنندهٔ صریح / ارائه‌دهندهٔ صریح / پیش‌فرض پل)، که هر یک به‌جای تعویض خاموش، خطای سخت تولید می‌کنند، به‌علاوه حذف عمدی محیط‌های محلی از مسیر پیش‌فرض تا چیزی ضعیف‌تر برای جایگزینی وجود نداشته باشد.

- **اجرای مکانیکی** — دروازهٔ پیش‌ساخت `CM-NO-LOCAL-RUNTIME` به‌همراه آزمون جهش‌یافتهٔ جفت‌شده، در زمان ساخت تأیید می‌کند که هیچ کلاینت محیط محلی هرگز در مسیر پیش‌فرض ساخته نمی‌شود: این ضمانت نمی‌پوسد زیرا اگر چنین شود، ساخت با شکست مواجه می‌شود.
- **زنجیرهٔ جایگزین قطعی و مرتب‌شده بر اساس امتیاز** — جایگزینی ارائه‌دهنده به ارائه‌دهنده در مجاز و کاملاً شفاف است، تمایزی اصولی با جایگزینی خاموش ممنوع تأییدشده میان مدل‌های همیشه می‌دانید کدام مدل توانمند کار را به‌دست گرفته است.
- **پایش بلادرنگ WebSocket** — رویدادهای ترجمهٔ تایپ‌شده به‌صورت زنده به داشبورد SSH ارسال می‌شوند، با کارگران توزیع‌شدهٔ تا کار ترجمهٔ کتاب به‌صورت موازی و قابل‌مشاهده انجام شود، نه در جعبه‌ای سیاه.
- **رژیم آزمون ضدفریب** — آزمون جهش، ادعاهای منفی، اجرای سیستم واقعی، و تضمین کیفیت مبتنی بر بینایی با هم تضمین می‌کنند که «آزمون‌ها قبول می‌شوند» هرگز نتواند «قابلیت واقعاً کار نمی‌کند» را به‌صورت خاموش پنهان کند.

بزرگ‌ترین چالش‌های فنی و راه‌حل‌های ما

- **ضمانت خط لولهٔ ترجمهٔ صادقانه (بدون تنزل کیفیت خاموش).** با متمرکزسازی همهٔ اختیارات حل شد تا نقطهٔ تصمیم‌گیری واحدی برای نظارت وجود داشته باشد، کدگذاری چهار شاخهٔ مسیریابی صریح که هر یک به‌جای حدس زدن، با خطای بلند شکست می‌خورند، حذف کامل جایگزین‌های محیط محلی از مسیر پیش‌فرض، و جوشکاری این قانون با دروازهٔ ساخت به‌علاوه آزمون جهش که اگر ضمانت هرگز حذف شود، ساخت با شکست مواجه می‌شود.
- **«آزمون‌ها سبز، قابلیت‌ها شکسته».** این حالت شکست در قانون اساسی به‌صراحت نام برده شده و با رژیم آزمون ضدفریب شکست داده می‌شود: ادعاهای ملموس و قابل‌مشاهده برای کاربر به‌جای جزئیات پیاده‌سازی، سیستم‌های واقعی در حلقه (ماک‌ها محدود به آزمون‌های واحد)، آزمون جهش اجباری (شکست عمدی قابلیت باید آزمون را قرمز کند)، و تضمین کیفیت مبتنی بر بینایی که واقعاً به خروجی نگاه می‌کند.
- **کیفیت چندقالبی و بلندمدت.** ورودی‌های به‌اندازهٔ کتاب هم ثبات و هم بودجه را تحت فشار قرار می‌دهند؛ با پرداخت چندمرحله‌ای که متن را دوباره مرور می‌کند، تحلیل فاز آماده‌سازی که حجم کار را از پیش تخمین می‌زند، و حافظهٔ نهان ترجمه که از پرداخت دوبارهٔ یک بخش جلوگیری می‌کند حل شد.

محتوا

پشتهٔ فناوری

- **Go** — به‌خاطر ساختارهای همزمانی‌اش انتخاب شده که به‌طور طبیعی با پردازش، ترجمه و جریان‌دهی همزمان چندین فصل سازگار است؛ بک‌اند با همزمانی بالا، ماژول `digital.vasic.translator`.
- **Gin** — REST سریع و کم‌حجم برای سرویس‌دهی به سطح HTTP به‌عنوان یک مسیریاب انتخاب شده است API.
- **QUIC / HTTP/3 (quic-go)** — برای فراهم‌کردن یک انتقال کم‌تأخیر و مدرن برای REST API انتخاب شده که در شبکه‌های ناپایدار نیز کارایی خود را حفظ می‌کند.
- **gRPC + Protocol Buffers** — برای ایجاد یک رابط سرویس قویاً تایپ‌شده و با REST کارایی بالا در کنار برای فراخوانی‌های برنامه‌نویسی انتخاب شده است.
- **Gorilla WebSocket** — برای انتقال جریان رویدادهای ترجمه به‌صورت بلادرنگ و تایپ‌شده انتخاب شده که داشبورد نظارتی را به‌صورت زنده تغذیه می‌کند.
- **PostgreSQL, SQLite, Redis** — تقسیم عمدی سه‌لایه: PostgreSQL برای داده‌های رابطه‌ای پایدار، SQLite برای وضعیت محلی/توکار (که همچنین پشتیبان فروشگاه

به‌عنوان حافظهٔ Redis و (`data/verified_models.db`، مدل‌های تأییدشده پل است پنهان داغ.
- **unidoc/unioffice + unipdf** — برای مدیریت فرمت‌های دشوار انتخاب شده‌اند: به‌گونه‌ای که کتاب‌های الکترونیکی چندفرمته به‌طور دقیق PDF و DOCX پردازش و بازسازی دورگردانی شوند.
- **Cobra** — و `unified-translator` انتخاب شده که ابزار CLI به‌عنوان چارچوب ابزارهای همراه آن‌را قدرت می‌بخشد.
- **golang-jwt (JWT HS256)** — انتخاب شده که API برای احراز هویت بدون‌حالت برای سخت‌کردن TLS/QUIC و امنیت انتقال IP همراه با محدودیت نرخ توکن بر اساس سطح حمله استفاده می‌شود.
- **پل LLMsVerifier (`pkg/bridge`)** — محور اصلی: قوی‌ترین مدل تأییدشده به‌همراه زنجیرهٔ بازگشتی قطعی‌اش‌را تأمین می‌کند و به‌عنوان تنها نقطهٔ اعمال تضمین «عدم بازگشت خاموش» عمل می‌کند.
- **Testify** — انتخاب شده است، از جمله تست اختصاصی Go برای مجموعه تست‌های `provider_routing_test.go` و دروازه‌های تغییر که قوانین صداقت‌را صادق نگه می‌دارند.
- **Docker / Podman (بدون نیاز به روت) + Compose** — برای استقرار کانتینری Podman انتخاب شده‌اند، با (`docker-compose.distributed.yml`) و توزیع‌شده بدون نیاز به روت برای وضعیت امنیتی سخت‌گیرانه‌تر.

وضعیت و یادداشت‌های صداقت

- **وضعیت: بتا.** `AGENTS.md` و `Makefile`، `VERSION` پلتفرم کارکردی است؛ نسخه در فایل‌های به‌طور ناهماهنگ ذکر شده و بنابراین غیرقطعی تلقی می‌شود.
- **مجوز: نامشخص.** فایل README ادعای مجوز MIT دارد اما این موضوع در فایل LICENSE تأیید نشده است — پیش از اعلام، تأیید شود.
- نقاط پایانی داشبورد/نظارت فقط محلی هستند و عمومی نیستند. اعداد عملکرد WebSocket همچنان به `ARCHITECTURE.md` در مستندات اهداف اعلام‌شده هستند و تأیید نشده‌اند. فایل که حذف شده‌اند اشاره دارد (اطلاعات قدیمی) Ollama موتورهای محلی.

اولویت لایه: Helix-اصلی (خوشهٔ LLM-زیرساخت). پس از Helix، در خانوادهٔ پلتفرم‌های HelixTrack قرار دارد.