گزارش نهایی سیستم طبقه‌بندی متن و وب‌سایت (NLP Classifier)

۲۳ شهریور ۱۴۰۵

فهرست مطالب

۱ خلاصه مدیریتی ۱

۲ دامنه گزارش و زمینه پروژه ۱

۳ مدل زبانی و زیرساخت اجرا ۱

۴ معماری داده و شاخص برداری ۲

۵ منطق طبقه‌بندی و دتکتورها ۲

۶ امتیازدهی نهایی و میزان اطمینان ۳

۷ پیشنهادهای بهبود ۳

۸ منابع ۴

۱ خلاصه مدیریتی

این گزارش نهایی، مستندات فنی سامانه طبقه‌بندی متن و وب‌سایت (NLP Classifier) را بر پایه کدهای واقعی پروژه – یعنی فایل‌های «nlp_classifier_copy_v1.py»، «build_taxonomy_embeddings.py» و «taxonomies.py» – تدوین کرده است. سامانه با تلفیق دو رویکرد «شباهت معنایی برداری» و «قواعد مبتنی بر الگو و کلیدواژه»، اطلاعات ورودی شرکت‌ها و وب‌سایت‌ها را در پنج مدل کسب‌و، مدل کسب‌وکار، نوع مشتری، نقش شرکت و موقعیت جغرافیایی دسته‌بندی می‌کند.

هسته مدل، امبدینگ چندزبانه «BAAI/bge-m3» است که به‌صورت کاملاً محلی و روی پردازنده اجرا می‌شود. بردارهای طبقه‌بندی‌ها پیش از اجرا به‌صورت آفلاین ساخته و در دو فایل «taxonomy_embeddings.npz» و «taxonomy_metadata.json» ذخیره می‌شوند؛ این پیش‌محاسبه باعث شده در زمان طبقه‌بندی، هر درخواست تنها با یک ضرب برداری و بدون بارگذاری مجدد مدل پاسخ داده شود. خروجی هر طبقه‌بندی شامل برچسب نهایی، درصد اطمینان و فهرست گزینه‌های برتر (top_k) است.

۲ دامنه گزارش و زمینه پروژه

هدف این سامانه، طبقه‌بندی خودکار شرکت‌ها و وب‌سایت‌ها برای فهرست‌های تجاری و سامانه‌های اطلاعاتی است؛ به‌گونه‌ای که برای هر ورودی، ویژگی‌های ساختاری آن بدون دخالت دستی مشخص شود. دامنه گزارش به سه مؤلفه اصلی پروژه محدود است:

- «taxonomies.py»: مرجع تعریف طبقه‌بندی‌ها؛ شامل برچسب‌ها، کلیدواژه‌ها، مترادف‌ها، سیگنال‌های متنی و دیکشنری نرمال‌سازی کشورها (COUNTRY_CANONICAL و COUNTRY_NORMALIZATION).
- «build_taxonomy_embeddings.py»: اسکریپت شاخص‌ساز که عبارات وزن‌دار هر دسته را با مدل امبدینگ کدگذاری کرده و شاخص برداری نهایی را می‌سازد.
- «nlp_classifier_copy_v1.py»: موتور طبقه‌بندی که در زمان اجرا از شاخص ساخته‌شده استفاده می‌کند و دتکتورهای تخصصی هر فیلد را در اختیار می‌گذارد.

این تفکیک، به‌روزرسانی طبقه‌بندی‌ها را بدون تغییر در موتور طبقه‌بندی ممکن می‌سازد؛ کافی است مرجع طبقه‌بندی ویرایش و شاخص دوباره ساخته شود.

۳ مدل زبانی و زیرساخت اجرا

مدل امبدینگ استفاده‌شده در هر دو فایل اصلی، «BAAI/bge-m3» است؛ مدلی چندزبانه با پشتیبانی قوی از فارسی و انگلیسی که برای تطبیق معنایی متن‌های کوتاه تا متوسط انتخاب شده است (nlp_classifier_copy_v1.py، خطوط 15–21).

نکات کلیدی زیرساخت اجرا:

- **بارگذاری محلی و آفلاین:** مدل همیشه از مسیر لوکال اسنپ‌شات هugsینگ‌فیس بارگذاری می‌شود و بارگذاری بر اساس نام فعال نیست؛ در اسکریپت شاخص‌ساز حالت آفلاین (HF_HUB_OFFLINE و TRANSFORMERS_OFFLINE) نیز به‌صورت پیش‌فرض برقرار است تا از تماس با هاب جلوگیری شود (build_taxonomy_embeddings.py، خطوط 23–29 و 230–269).
- **اجرای صرفاً روی CPU:** در ابتدای هر دو فایل، متغیر محیطی «CUDA_VISIBLE_DEVICES» خالی تنظیم می‌شود. این تصمیم عمداً برای سازگاری با سیستم‌های فاقد کارت گرافیک و جلوگیری از خطای توکنایزر ویندوز گرفته شده است.
- **نمونه تک‌نسخه از مدل:** تابع «get_model» در طبقه‌بند، مدل را به‌صورت Singleton و با کش‌کردن بارگذاری می‌کند تا بار مکرر حافظه نداشته باشد.
- **نرمال‌سازی بردارها:** با NORMALIZE_EMBEDDINGS=True، همه بردارها واحدسازی می‌شوند تا محاسبه شباهت کسینوسی با یک ضرب نقطه‌ای ساده ممکن باشد.

۴ معماری داده و شاخص برداری

سامانه رویکرد «جست‌وجوی برداری مبتنی بر سنتروئید» با پیش‌محاسبه آفلاین دارد. به این معنا که به‌جای مقایسه متن با تک‌تک عبارات در زمان اجرا، یک بردار نماینده (Centroid) برای هر دسته ساخته و ذخیره می‌شود.

طبقه‌بندی‌های هشت‌گانه: اسکریپت شاخص‌ساز از «taxonomies.py» هشت طبقه‌بندی را می‌خواند: نوع سایت (site_type)، مدل کسب‌وکار (business_model)، نوع مشتری (customer_type)، نقش شرکت (company_role)، دامنه بازار (market_scope)، منطقه جغرافیایی (market_region)، صنعت (industry) و گروه محصول (product_groups).

استخراج عبارت وزن‌دار: برای هر دسته، اجزای مختلف با وزن‌های مشخص به شاخص اضافه می‌شوند: برچسب اصلی با وزن 3، کلیدواژه‌ها و مترادف‌ها با وزن 2، سیگنال‌های متنی با وزن 1.5 و توضیحات با وزن 1. عبارات تکراری قبل از امبدینگ حذف می‌شوند (build_taxonomy_embeddings.py، خطوط 62–81).

محاسبه سنتروئید: تابع «compute_weighted_centroid» میانگین وزن‌دار بردارهای عبارات هر دسته را با «np.average» محاسبه و نتیجه را نرمال‌سازی می‌کند تا به‌عنوان بردار نماینده دسته استفاده شود (خطوط 296–322).

خروجی نهایی: بردارها در «taxonomy_embeddings.npz» و متادیتا (شامل نام مدل، مسیر، وزن‌های فیلدها، اثر انگشت sha256 هر طبقه‌بندی و ساختار سلسله‌مراتبی صنعت در سه سطح (l1/l2/l3 در

«taxonomy_metadata.json» ذخیره می‌شود. موتور طبقه‌بندی این دو فایل را در زمان ایمپورت به‌صورت خودکار در حافظه بارگذاری می‌کند.

۵ منطق طبقه‌بندی و دتکتورها

لایه تشخیص در «nlp_classifier_copy_v1.py» کاملاً ماژولار است و هر فیلد دتکتور اختصاصی خود را دارد. پیش از هر ارزیابی، متن ورودی از طریق توابع نرمال‌سازی و پاک‌سازی یکدست می‌شود و در صورت نیاز به قطعه‌های کوچک‌تر تفکیک می‌گردد.

- **detect_site_type** (خط 705): رویکرد «قاعده‌محور در اولویت» دارد. ابتدا الگوهای URL فهرست‌های تجاری (مانند «-importers»، «-exporters»، «-suppliers»، «-manufacturers»، «-directory» و «/company-list») بررسی می‌شوند؛ اگر الگوی URL مطابقت داشته باشد یا دست‌کم دو سیگنال متنی از سیگنال‌های فهرست تجاری در متن پیدا شود، نتیجه «directory_listing» با اطمینان 0.95 (تطبیق هم‌زمان URL و متن) یا 0.88 (فقط قاعده) برگردانده می‌شود. در غیر این صورت، به‌عنوان پشتیبان، از امبدینگ با آستانه 0.30 استفاده می‌کند.
- **detect_business_model** (خط 745): تشخیص مدل B2B، B2C یا سایر مدل‌های تجاری صرفاً با شاخص برداری «business_model» و آستانه 0.30.
- **detect_customer_type** (خط 760): تفکیک بازار هدف و مخاطب نهایی با شاخص «customer_type».
- **detect_company_role** (خط 775): تعیین ماهیت سازمان و نقش آن در زنجیره عرضه با شاخص «company_role».
- **detect_country** (خط 790): استخراج کشور بر مبنای دیکشنری مرجع و عبارات منظم (regex)؛ در صورت یافته شدن، کشور نرمال‌شده به‌همراه فهرست کشورها و اطمینان 0.90 برگردانده می‌شود و در غیر این صورت اطمینان صفر است.

در کنار این پنج دتکتور اصلی، تابع «detect_market_scope» نیز دامنه جغرافیایی بازار را در چهار سطح (چندکشوری، ملی، منطقه‌ای، محلی) با نردبان اطمینان 0.85 تا 0.60 تخمین می‌زند.

۶ امتیازدهی نهایی و میزان اطمینان

قلب ارزیابی، تابع «classify_with_embeddings» است: متن ورودی امبدینگ شده و با بردار سنتروئید هر دسته مقایسه می‌شود. به‌جای اتکا به تک عدد شباهت، امتیاز هر دسته از ترکیبی وزن‌دار از چند سیگنال ساخته می‌شود (خطوط 440–521):

- **۵۰٪** بهترین شباهت کسینوسی میان عبارات دسته و متن؛
- **۳۰٪** میانگین سه شباهت برتر (top_mean)؛
- **۱۲٪** نسبت عبارات پشتیبان (support_ratio) با شباهت دست‌کم 0.34؛
- **۸٪** سقف تعداد پشتیبان‌ها (تا سه مورد).

همچنین اگر دسته فقط یک عبارت پشتیبان داشته باشد و فاصله‌اش با رقیب نزدیک باشد، جریمه 0.08 و اگر نسبت عبارات ضعیف (شباهت کمتر از 0.22 بالا باشد، جریمه تا 0.06 اعمال می‌شود. وزن‌های عبارات هنگام امتیازدهی در بازه $[0.25, 4]$ محدود می‌شوند تا یک کلیدواژه سنگین، کل نتیجه را تسخیر نکند.

آستانه‌ها و خروجی: همه فیلدها آستانه پذیرش 0.30 دارند (جدول CLASSIFICATION_THRESHOLDS در خطوط 30–42). اگر هیچ دسته‌ای از آستانه عبور نکند، برچسب «unknown» به‌همراه بالاترین امتیاز یافته‌شده برگردانده می‌شود و در صورت بروز خطا، اطمینان صفر و پیام خطا ثبت می‌گردد. خروجی استاندارد هر دتکتور شامل برچسب نهایی، عدد اطمینان، فهرست گزینه‌های برتر (top_k) و فهرست نامزدهاست؛ توابع کمکی «_build_candidate_list» و «_top3_labels_from_candidates» نیز سه گزینه برتر را برای مصرف در لایه‌های بالاتر مرتب می‌کنند.

۷ پیشنهادهای بهبود

معماری فعلی از سه جهت قابل ارتقاست و پیشنهادهای زیر بر مبنای ساختار موجود تنظیم شده‌اند:

- **اعتبارسنجی کمّی:** مجموعه‌ای از نمونه‌های برچسب‌دار بسازید و دقت (precision/recall) هر دتکتور را گزارش کنید؛ این کار پایه‌ای برای مقایسه نسخه‌های بعدی مدل است.
- **به‌روزرسانی مدل امبدینگ:** نسخه‌های جدیدتر خانواده bge یا مدل‌های چندزبانه رقیب را روی همان مجموعه اعتبارسنجی بسنجید؛ مسیر مدل در هر دو فایل از طریق متغیر محیطی قابل تعویض است و نیازی به تغییر کد ندارد.
- **حذف جریمه‌های دستی به‌مرور:** ضرایب جریمه 0.08) و 0.06) به‌صورت تجربی تنظیم شده‌اند؛ با رشد داده برچسب‌دار می‌توان این ضرایب را از طریق بهینه‌سازی روی داده واقعی کالیبره کرد.
- **تقویت دتکتور کشور:** افزودن پشتیبانی از کتابخانه استاندارد نرمال‌سازی جغرافیا (مانند pycountry) و پوشش زبان‌های بیشتر در الگوهای regex، دقت استخراج کشور را بهبود می‌دهد.
- **گزارش‌گیری و پایش:** ثبت امتیاز و اطمینان هر طبقه‌بندی در پایگاه داده موجود پروژه، امکان شناسایی الگوهای «unknown» و اصلاح هدفمند طبقه‌بندی‌ها را فراهم می‌کند.

۸ منابع

منابع مورد استناد در این گزارش، فایل‌های واقعی پروژه‌اند که مستقیماً بررسی و استخراج شده‌اند:

- **S1** — «nlp_classifier_copy_v1.py»، کد طبقه‌بند متن پروژه؛ خطوط 12–42 (تنظیمات مدل و آستانه‌ها)، 146–204 (بارگذاری شاخص)، 440–610 (امتیازدهی و طبقه‌بندی برداری) و 705–799 (دتکتورهای تخصصی).
- **S2** — «build_taxonomy_embeddings.py»، اسکریپت ساخت بردارهای طبقه‌بندی؛ خطوط 24–81 (تنظیمات آفلاین و وزن‌های فیلدها)، 230–322 (بارگذاری مدل و محاسبه سنتروئید) و 405–519 (متادیتا و جریان اصلی ساخت شاخص).
- **S3** — «taxonomies.py»، مرجع تعریف طبقه‌بندی‌ها، کلیدواژه‌ها، مترادف‌ها و نرمال‌سازی کشورها.