

بازتولید نتایج مقاله InstructABSA و بهبود آن

۱. باز تولید نتایج

به عنوان نمونه، برای بازتولید نتایج اجرای تسک ATSC با مدل InstructGPT2 بر روی دیتاست Rest15، کافیت اسکریپت طراحی شده با عنوان simple_eval.py را اجرا نماییم که خروجی مورد انتظار (Accuracy: 84.50%) را مطابق تصویر ذیل تولید می‌نماید.

```
(venv) PS F:\NLP\Project\InstructABSA> python simple_eval.py
Device: cpu
Loading model...
Model loaded!
Loaded 685 samples
100%|██████████████████████████████████████| 685/685 [02:42<00:00, 4.23it/s]

=====
Results on SemEval15 Restaurants Test
=====
Correct: 458/542
Accuracy: 84.50%

Paper reports: 84.50%
```

لازم به ذکر است:

- ✓ جرئیات بخش‌های مختلف این اسکریپت به صورت کامنت درون آن درج شده است.
- ✓ برای بازتولید کل نتایج جدول شماره ۲ مقاله (ATSC subtask results-Accuracy) میتوان از اسکریپت `Reproduce_ATSC_results.py` استفاده نمود که نتایج ذیل را تولید می‌نماید:

ATSC Results (Accuracy):

Model	Lapt14	Rest14	Rest15	Rest16
InstructABSA-2	81.56	85.17	84.50	89.43

- ✓ به منظور تست عملکرد مدل در تسک ATSC، میتوان هر فایل ورودی CSV دلخواه با فرمت text,aspect ساخت (مانند فایل sample_input.csv) و با اجرای دستور زیر، نتایج پیش بینی مدل را مشاهده کرد:

➤ **python inference.py --input sample_input.csv --output predictions.csv**

```
Results saved to results.csv

Sample predictions:
text aspect predicted_sentiment
The pizza was absolutely delicious and fresh pizza positive
I had to wait 45 minutes for my order wait negative
The restaurant has a cozy atmosphere atmosphere positive
Prices are way too high for this quality prices negative
Staff was friendly and helpful staff positive
The pasta was cold when it arrived pasta negative
Great location in the city center location positive
Portion sizes are generous portion sizes positive
The music was too loud music negative
Best coffee I've ever had coffee positive

(venv) PS F:\NLP\Project\InstructABSA>
```

۲. بهبود خروجی مدل با تغییرات کوچک و معنادار

برای بهبود خروجی مدل، برخی تغییرات به شرح بندهای ذیل بررسی شده‌اند.

۲-۱. تغییر تعداد مثال‌های ورودی – Prompt Engineering

مدل در دستورالعمل پایه از ۲ مثال مثبت، ۲ منفی و ۲ خنثی در Prompt/Instruction استفاده می‌نماید و ما این عدد را به مقادیر ۴ و ۸ تغییر می‌دهیم و تسک ATSC را روی دیتاست Rest15 اجرا می‌نماییم (Prompt_engineering.py) تا بررسی کنیم با افزایش/کاهش مثال‌ها، دقت بالاتر مدل می‌رود یا خیر؟

- تعداد مثال‌ها ۴ (2 positive + 1 negative + 1 neutral): **بهبود ۱.۴۸٪ در Accuracy**
- تعداد مثال‌ها ۸ (3 positive + 3 negative + 2 neutral): **بهبود ۱.۲۹٪ در Accuracy**

```
=====
RESULTS - InstructABSA on Rest15 (542 samples)
=====
Prompt          Accuracy      vs Paper
-----
Paper (6 examples)  84.50%        -
4 Examples       85.98%        +1.48%
8 Examples       85.79%        +1.29%
=====

🏆 Best: 4 Examples (85.98%)
```

اما قاعدتاً Prompt طولانی‌تر معادل با inference کندتر و گران‌تر خواهد بود و بایستی Trade off داشته باشیم بین دقت و هزینه و سرعت.

۲-۲. استفاده از مدل Flan-T5 به جای Tk-Instruct

در این مرحله به جای استفاده از مدل Tk-Instruct که بر روی ۱۶۰۰ تسک آموزش دیده است، از مدل Flan-T5 که روی ۱۸۰۰ تسک آموزش دیده استفاده کرده و عملکرد آنها را در تسک ATSC روی دیتاست Rest15 بررسی میکنیم.

مدل	توضیح	Instruction-tuned?
Flan-T5	آموزش روی ۱۸۰۰ تسک + T5	✓
Tk-Instruct	آموزش روی ۱۶۰۰ تسک + T5	✓

تحلیل نتیجه:

Flan-T5 (zero-shot): **84.69%** Tk-Instruct + ABSA: 84.50%

Flan-T5 با آموزش روی تسک‌های بیشتر و متنوع‌تر، بدون fine-tuning اختصاصی، عملکرد مشابهی با InstructABSA دارد.

Model	Accuracy
Random Baseline	33.03%
Majority Baseline	60.15%
Flan-T5-base (zero-shot)	84.69%
Flan-T5-base (6-shot)	83.03%
InstructABSA (6-shot, fine-tuned)	84.50%

۲-۳. مقایسه با Baseline

در این بخش عملکرد مدل InstructABSA2 در تسک ATSC بر روی دیتاست Rest15 با دو Baseline ساده مندرج در جدول ذیل مقایسه شده است تا عملکرد مدل در مقایسه با آنها بررسی شود (baseline_comparison.py):

Method	Accuracy
Random Baseline (positive/negative/neutral))	33.03%
Majority Baseline(Only Positive)	60.15%
InstructABSA2	84.50% ✓

مقایسه خروجی مدل InstructABSA با Baseline‌های فوق نشان می‌دهد که مدل InstructABSA واقعا یاد گرفته و خروجی تصادفی تولید نخواهد کرد. این خروجی در تصویر ذیل نیز قابل مشاهده است.

```
(venv) PS F:\NLP\Project\InstructABSA> python baseline_comparison.py
• Device: cpu
Total samples: 542

Class distribution:
Positive: 326 (60.1%)
Negative: 182 (33.6%)
Neutral: 34 (6.3%)
Majority class: positive

=====
Baseline 1: Random
=====
→ Accuracy: 33.03%

=====
Baseline 2: Majority (always 'positive')
=====
→ Accuracy: 60.15%

=====
Model: InstructABSA
=====
Loading model...
100%|
→ Accuracy: 84.50%
```

۲-۴. تست عملکرد مدل برای زبان فارسی

آیا مدلی که انگلیسی آموزش دیده (Tk-Instruct) می‌تواند بر روی جملات فارسی هم عمل کند؟ و آیا ارائه مثال در Prompt کمک می‌کند تا فارسی را بهتر بفهمد؟ در این بررسی، ۱۵ داده تست فارسی در دو حالت Zero shot و Few shot به مدل داده می‌شوند و عملکرد تسک ATSC در مورد آنها بررسی می‌شود (persian_comparison.py).

```
=====
Final Results - Persian ATSC
=====
Prompt          Accuracy
-----
Zero-shot       40.00%
6-shot Persian  20.00%

Improvement: -20.00%
⚠ Few-shot Was Not Effective

Saved to Output/persian_zero_vs_6shot.csv
```

تحلیل نتیجه:

✓ به نظر می‌رسد مدل InstructGPT که فقط روی انگلیسی آموزش دیده، در خصوص داده فارسی خروجی را صرفاً حدس زده، توانایی Cross-lingual Transfer به فارسی را نداشته و نیاز به fine-tuning روی داده فارسی دارد.

✓ اضافه کردن مثال فارسی به مدل انگلیسی نه تنها کمک نمی‌کند، بلکه عملکرد را بدتر می‌کند و باعث سردرگمی مدل می‌شود (از ۴۰٪ در حالت zero shot به ۲۰٪ در حالت fewshot).

۳. جمع‌بندی:

اسکرپت‌هایی که در این پروژه تهیه شده و در توضیحات فوق به آنها اشاره شد به شرح ذیل می‌باشند:

ردیف	توضیح	نام اسکرپت/فایل
۱	راهنمای بارگذاری و اجرای پروژه	README.md
۲	ارزیابی ساده مدل و بازتولید نتایج اجرای تسک ATSC بر روی دیتاست Rest15	simple_eval.py
۳	بازتولید نتایج اجرای تسک ATSC بر روی همه دیتاست‌ها	Reproduce_ATSC_results.py
۴	پیش‌بینی روی فایل ورودی کاربر (تسک ATSC با ورودی text + aspect)	inference.py sample_input.csv
۵	تغییر مثال‌ها به مقادیر ۴ و ۸ در تسک ATSC و تست خروجی برای دیتاست Rest15	Prompt_engineering.py
۶	مقایسه مدل Flan-T5 با Tk-Instruct	test_flan_t5.py
۷	مقایسه خروجی مدل با Baseline	baseline_comparison.py
۸	بررسی عملکرد مدل بر روی زبان فارسی	persian_comparison.py