CS 336: Assignment 4

Brandon Snider

May 22, 2025

Contents

2 Filtering Common Crawl

Problem (`look_at_cc`): 4 points

a) URL: http://0371rykj.com/ipfhsb/34.html°

The URL is no longer accessible.

It looks like the page could be an e-commerce website or profile page for a company called "Shanghai Linpin Instrument Stock Co Ltd", but that's about as much as I can gather from the raw HTML, most of which is not in English.

b) The extracted text appears to contain URL paths, phone numbers, unique identifiers of some sort (e.g. product codes), product specifications, and possible a category selector or filtering mechanism of some sort (along with a lot of non-English text).

At a minimum, information like phone numbers and email addresses should be filtered out. Even if people or companies want this information published, they may not want it embedded in a model trained on this data. Additionally, having content that is structured like a web page could cause a model trained on this data to learn patterns in the layouts and content of web pages, which may not be desirable depending on what the model is being trained for (e.g. answering questions, as opposed to generating web-style content).

That said, a page like this could help a model learn about the structure of webpages, or about product specifications and their relationship to surrounding content, which could be useful depending on the use case.

c) If our use case is a copilot for writing optimized product pages, it could be beneficial to include content like product specifications and descriptions, structured like a web page, in the training data. If we're building an AI therapist, this data is unlikely to be useful.

d) Examples until first high-quality page: 3

The table below shows my findings in the first 25 contentful results from the Common Crawl sample. Almost all of them would ideally be filtered out. I happened to find a page containing some high-quality content at the third position (a page for a computational mechanics olympiad containing some fluent English), but this was the only such page in the sample.

Language	Page Type	Domain Name
Chinese	E-commerce/Industrial	0371rykj.com
Chinese, English	Fan Blog/Forum	10www.chinatikfans.com
English	Conference/Academic	13.usnccm.org
Chinese	Adult Chat	176.utchat888.com
Japanese, Chinese	Spam/Error Page	176766.cn
Chinese	Spam/Error Page	178mh.com
Chinese	Adult Live Stream	1796370.tgtg97.com
Chinese, English	Adult Video	18sex.v340.info

Language	Page Type	Domain Name
Dutch, English	Blog	1kb.klimtoren.be
Greek, English	Help Desk/Forum	1pekesat-exae.mysch.gr
Greek, English	Help Desk/Forum	1pekesat-exae.mysch.gr
Chinese	Content Farm/App	1s6605084.yhxzseo.com
Turkish, Danish, English	Software Doc	20com20.fr
English	Gambling	24ktcasino.net
English	Error Page	2kgames.eu
Chinese	Content Farm/App	2l6185919.yizhangting.com
Chinese	Spam	303323.com
Chinese	Pirate Streaming/Anime	30bad.com
Chinese	Unclear	312001.net
Chinese	Adult Chat/Dating	354577.mwe075.com
English	Empty Search Results	356.schoollibrary.edu.pe.ca
Chinese	Adult Chat/Video	366392.haaxz.com
Chinese, English	Adult Chat/Dating	387tel.com
Spanish	Blog/Forum	3diasdemarzo.blogspot.com

Table 1: First 25 WET records from the Common Crawl sample

Problem (`extract_text`): 3 points

a) See `cs336_data/extract_text.py`

b) In general, the extracted text in the WET files is much higher-quality and that produced by the custom function. The custom function's extracted text is mostly whitespace, and contains some formatting characters (like bullet points). Both the whitespace and bullet points have been stripped from the WET files' extracted text. There are also some substantive content differences; for example, the custom function's text contains image names like "bridge_USACM_trim.jpg", which are not present in the WET files.

Problem (`language_identification`): 6 points

a) See `cs336_data/language_identification.py`

b) If we're trying to train a model that performs well in a particular language, it would be highly problematic if, with high frequency, the language identifier (i) incorrecly classified other languages as the target language (ii) failed to recognize the target language. In the first case, the noisy data would prevent the model from learning the target language well. In the second case, the model might fail to be useful in some scenarios due to patchy understanding of the target language.

In a high stakes scenario, we could (1) tune two confidence thresholds, discarding very low-confidence predictions ($c < C_{\text{min}}$), flagging moderate-confidence predictions ($C_{\text{min}} \leq c < C_{\text{max}}$) for human-in-the-loop review, and accepting high-confidence predictions ($C_{\text{max}} \leq$

c). We could periodically audit and recalibrate the classifier by sampling predictions on held-out data. We could also ensemble multiple language-ID models or supplement fastText with rule-based checks (e.g., keyword lists or script detection) to further reduce systematic errors.

c) In a sample of ~27k documents, the classifier reports that 43% are in English.

The classifier seems to classify almost all documents in the same way that I would. However, when the classifier's score is low, the document often either (i) contains a mixture of languages (e.g. a restaurant menu with item names listed in English and Spanish), or (ii) is low-quality, containing a lot of random character strings, phone numbers, and the like.

Examples of both (i) and (ii) are shown below.

From my observations, a threshold of ~0.8 would exclude the vast majority of ambiguous cases (no dominant language) and classification errors. The overwhelming majority of documents for which there is a clear correct classification have a score > 0.8, so we'd keep almost all high-quality documents, while discarding almost all others.

Example of (i) (multiple languages causing low score):


```
{
   "url": "https://grapevine.ca/listing/208-asper-trail-circle-ottawa-ontario-k2m-0k7-27802601/",
   "lang": "en",
   "score": 0.7182,
   "snippet": "| 613.829.1000  • Home  • Sell  • For Sale By Owner  • Fully Brokered  • Compare Services
• Buy  • Get Cash Back  • Grapevine Listings  • Ottawa Listings  • About Us  • Our Company  • Our
Realtors  • Contact Us  • Sold & Saved  • Recent Sales  • Testimonials  LOADING  • « Go back  208 Asper
Trail Circle Ottawa, Ontario K2M 0K7  view favourites  • 208 Asper Trail Circle, Ottawa, Ontario K2M 0K7 - Photo
1 - X11923597  • 208 Asper Trail Circle, Ottawa, Ontario K2M 0K7 - Photo 2 - X11923597  • 208 Asper Trail Circle,
Ottawa, Ontario K2M 0K7 - Photo 3 - X11923597  • 208 Asper Trail Circle, Ottawa, Ontario K2M 0K7 - Photo 4 -
X11923597  • 208 Asper Trail Circle, Ottawa, Ontario K2M 0K7 - Photo 5 - X11923597  • 208 Asper Trail Circle,
Ottawa, Ontario K2M 0K7 - Photo 6 - X11923597  • 208 Asper Trail Circle, Ottawa, Ontario K2M 0K7 - Photo 7 -
X11923597  • 208 Asper Trail Circle, Ottawa, Ontario K2M 0K7 - Photo 8 - X11923597  • 208 Asper Trail "
},
```


Example of (ii) (low-quality content causing low score):


```
{
   "url": "https://eventor.orientering.se/Ranking/ol/Event/Class/410195",
   "lang": "sv",
   "score": 0.7128,
   "snippet": "• In English In English  Produkter och tjänster  • Förbundssida  • Livesida  •
Omaps  • Livelox  • hitta orientering Svenska Orienteringsförbundet Eventor - Svensk orienterings
centrala IT-system  • Tävlingskalender  • Sverigelistan  • Pressresultat  • Forum  • Skapa
konto  • Logga in  • Orientering  • Skidorientering  • Tävlingar  • Gallringsfilter  •
Utlandstävlingar  • Betala rankingavgift  • Subjektiv ansökan  • Köp klubblicens  • Support
och kontakt  • Frågor och svar  Sverigelistan, skog  Hela landetDistriktvisKlubbvis (Uppdaterad 2025-04-17)
Alla damer  Alla herrar  D21D20D18 D16 D35D40D45D50D55D60D65D70D75D80D85D90D95  H21H20H18
H16 H35H40H45H50H55H60H65H70H75H80H85H90H95  Rankingfilter, skog  Hela landet (Uppdaterat fredag
2025-04-11)  D21D20D18  H21H20H18  Sverigelistan, sprint  Hela landet (Uppdaterad 2025-04-17)  Alla
damer  Alla herrar  D21D20D18 D16 D35D40D45D50D55D60D65D70D75D80D85D90D95  H21H20H18 H16
H35H40H45H50H55H60H65H70H75H80H85H90H9"
}
```

Problem (`mask_pii`): 3 points

a) See `cs336_data/mask_pii.py`

b) See `cs336_data/mask_pii.py`

c) See `cs336_data/mask_pii.py`

d) Issues with false positives: the language model may learn erroneous patterns in where phone numbers, emails, or IPs naturally occur in text, and may produce generations with these placholders in locations that appear nonsensical to a user.

Issues with false negatives: failure to mask some PII in the training set could cause the language model to output real PII in its generations.

Other issues:

- The model may refuse to produce generations that contain PII-like patterns. For example, a user might provide their email signature (containing PII) to a model and ask it to compose emails on the user's behalf. The model might then output emails ending with that signature, but with the PII masked, which would be frustrating for the user.
- Not every email, phone number, or IP is PII that should be masked. For example, various helpline numbers or emails should probably not be masked.
- Distribution shift: replacing diverse PII with uniform placeholders may cause the model to learn unnatural patterns and overgenerate these tokens.

Mitigations:

- Manually maintain a whitelist of PII-like data that should not be masked.
- Use more structured or randomized placeholders (e.g. "|||EMAIL_ADDRESS_1|||" instead of "|||EMAIL_ADDRESS|||") to reduce the likelihood of overgeneration.

e) Some examples of false positives and false negatives are shown below.

In general, email masking seems to be much more reliable than phone number masking, due to the more consistent and easily detectable structure of email addresses. Given the diversity of phone number formats, there are cases in which it's almost impossible to tell whether or not a string is a phone number without deep contextual understanding.

For example, in the first false positive listed below, a timestamp is masked as a phone number, simply because it's a string of digits of appropriate length. Only with some understanding of the surrounding context is this clear.

False positive (emails masked, phone numbers missed):


```
{
   "url": "http://indexrecruitment.com.np/testimonials",
   "text": "  • Enquiry\n  • Apply Now\n+977-1-5911443 |||EMAIL_ADDRESS|||\n+977-1-5911443 |||
EMAIL_ADDRESS|||\n  • Home\n  • About Us\n    • Introduction\n    • Mission\n    • Vision\n    • Our
Team\n    • Testimonials\n  • Services\n  • Gallery\n  • Blogs\n  • Contact\n  • Facebook\n  • Instagram\n  •
Twitter\n  • Youtube\n\nWhat People Say About Us\n\nPrakash Aryal\n\nSteel Bender\n\nI haven't worked here
long but I can say this is a great place to work the management is very helpful and understanding and helped
me with getting my friend on board same schedule\n\nABOUT OUR CONSULTING\n\nIndex Recruitment Pvt. Ltd.
\n\n+977-1-5911443 Link\n\nQuick Links\n\n  • Home\n  • Team\n  • Our Gallery\n  • Contact Us\n  • Book An
Appointment\n\nNEWSLETTER\n\n\n\n\nDesign & Developed By Web House Nepal",
   "emails_masked": 2,
```

```
    "phone_numbers_masked": 0,
    "ips_masked": 0,
    "all_masked": 2
}
```


False negative (phone number missed):


```
{
    "url": "http://adobsicimahi.org/undangan-rapat-pengurus-2016/",
    "text": "... Jenderal Achmad Yani (UNJANI)\nJl. Terusan Jenderal Gatot Subroto\nTelp. / Fax. |||PHONE_NUMBER|||
\nHunting 0811 249 7890\n\n© 2016 ADOBSI.",
    "emails_masked": 0,
    "phone_numbers_masked": 1,
    "ips_masked": 0,
    "all_masked": 1
}
```


False positive (timestamp masked as phone number):


```
{
    "url": "http://cdn.limetta.se/",
    "original_text": "imgix Configuration Details\n\nLast Deployed\nTue Mar 25, 2025 10:45:18 PM UTC
(1742942718)\nHash\n\"1563\"\n\nDashboard Website",
    "text": "imgix Configuration Details\n\nLast Deployed\nTue Mar 25, 2025 10:45:18 PM UTC |||
PHONE_NUMBER|||)\nHash\n\"1563\"\n\nDashboard Website",
    "emails_masked": 0,
    "phone_numbers_masked": 1,
    "ips_masked": 0,
    "all_masked": 1
}
```


Arguably a false positive (placeholder data masked as PII):


```
{
    "url": "http://3rte.com.br/product/lixeira-com-tampa-35-litros-preta-plasvale/",
    "original_text": "About\n  • Services\n  • Contact\n  • Shop\n  • Cart\n  • Checkout\n  • My account\nContact
Us\n1, My Address, My Street, New York City, NY, USA\n+1234567890\ncontact@domain.com\n1234567890\n© 2022
3RTE | PopularFX Theme",
    "text": "About\n  • Services\n  • Contact\n  • Shop\n  • Cart\n  • Checkout\n  • My account\nContact
Us\n1, My Address, My Street, New York City, NY, USA\n+|||PHONE_NUMBER|||\n|||EMAIL_ADDRESS|||\n|||
PHONE_NUMBER|||\n© 2022 3RTE | PopularFX Theme",
    "emails_masked": 1,
    "phone_numbers_masked": 2,
    "ips_masked": 0,
    "all_masked": 3
}
```


Problem (`harmful_content`): 6 points

a) See `cs336_data/harmful_content.py`

b) See `cs336_data/harmful_content.py`

c) Issues:

Naive application of these filters could skew the training distribution. For example, though we may no longer use this language, it may be useful for a model to understand what a master-slave relationship refers to in a CS context. We may also lose culturally significant language that includes borderline profanity, but may be important for a model to under-

stand. The classifiers may also simply produce false positives (removing benign content) and false negatives (retaining harmful content).

Mitigations:

- Calibrate confidence thresholds on manually vetted samples.
- Use soft filtering or example reweighting rather than outright removal.
- Incorporate periodic human-in-the-loop audits to refine classifier boundaries.

d) Examples of false positives and false negatives are shown below.

In a collection of 26,820 documents, 72 were classified as NSFW (0.27%) and 224 were classified as toxic (0.84%).

In general, finding an example of a false positive is much more difficult than finding a false negative, particularly for the NSFW classifier, for which there were many false negatives and almost no false positives.

No choice of thresholds will be perfect, because the classifiers' scores are very much imperfect (see the second example below, about which the NSFW classifier reports extremely high confidence for "non-NSFW").

It seems high thresholds (e.g. requiring >0.9 for a "non-NSFW" or "non-toxic" classification) would be reasonable, given the rarity of false positives and frequency of false negatives, even when the classifier reports high confidence for the "non-NSFW" or "non-toxic" classification. Though this is subjective, there is probably some asymmetry here in that a small number of false negatives is likely much more harmful than a small number of false positives.

When the classifiers are used together, it appears that a 0.9 threshold appears would catch a high proportion of all harmful content. For example, although the first example below should be classified as NSFW but isn't, it *is* classified as toxic, and therefore would be removed.

False positive (toxic):


```
{
   "url": "http://www.w3schools.com/python/trypython.asp?filename=demo_default",
   "nsfw": "non-nsfw",
   "nsfw_score": 0.9975,
   "toxic": "toxic",
   "toxic_score": 0.7668,
   "snippet": "Get your own Python server ❯Run Code Ctrl+Alt+R Change Orientation Ctrl+Alt+O Change Theme
Ctrl+Alt+D Go to Spaces Ctrl+Alt+P  Privacy policy and Copyright 1999-2025  Hello, World!"
}
```


False negative (should be NSFW; many such cases):


```
{
   "url": "https://amatureporнmaster.com/homemade-porn-videos/cheerleader/",
   "nsfw": "non-nsfw",
   "nsfw_score": 0.9845,
   "toxic": "toxic",
   "toxic_score": 0.8715,
   "snippet": "Free Sextapes Porn Videos And HQ Home XXX Films Only On Amatureporнmaster.Com  • Top
```

videos • New videos • All Models • All Niches Real Life Porn Tubes Trinity Does What It Takes - Trinity May Trinity Does What It Takes - Trinity May Candice Delaware In Exotic Xxx Movie Upskirt Watch , Its Amazing Candice Delaware In Exotic Xxx Movie Upskirt Watch , Its Amazing From To Cum Dumpster - Scarlett Mae From To Cum Dumpster - Scarlett Mae Raising Your Spirit With Nia Nacci Raising Your Spirit With Nia Nacci Big Butt Cheerleader Does Splits On The Dick - Belle Sparkles Big Butt Cheerleader Does Splits On The Dick - Belle Sparkles Beach Blonde Cheerleader Paisley Porter Takes That Huge Prick Deep Into Her Tight Vagina Beach Blonde Cheerleader Paisley Porter Takes That Huge Prick Deep Into Her Tight Vagina Crazy Porn Movie Big Tits Watch , It's Amazing Crazy Porn Movie Big Tits Watch , It's Amazing Jazzi Lai - Black Cheerleader Gang 26 Jazzi Lai - Black Cheerleader Gang 26 A Petite With"
}

Problem (`gopher_quality_filters`): 3 points

a) See `cs336_data/gopher_quality_filters.py`

b) On a sample of 26,820 documents, 7,362 passed the quality filters (27.45%).

I generally found high agreement between the quality filters and my own judgement.

Cases of mild disagreement:

There were instances in which the filters rejected documents that I might have kept, like the following product page: https://www.footballgiftsonline.co.uk/products/everton-pulse-double-duvet-set°. There's certainly high-quality training data on this page, but perhaps if we're only applying filters at the page level (rather than trying to extract the prose from the page), perhaps discarding it is more reasonable.

It's not obvious though, why that product page was filtered out, but this one was not: https://www.dshirt14.it/en/t-shirt/6988-T-shirt-Kids-Pop-Origami-Scottish-Terrier.html°. If anything the former appears more useful as training data, though neither is particularly dense with worthwhile text, so perhaps with page-level filtering both should be discarded.

There were may such cases of ecommerce-related pages that seem to be close to the boundary, as it's not clear why some are kept and other aren't.

Case of strong disagreement:

Though rare, there were a few cases, such as the one below, where I certainly would have kept a page that the filters rejected. Below is one example:

https://www.hji.edu/being-a-peacemaker-leadership-series-event/°

Problem (`quality_classifier`): 15 points

a) See `cs336_data/quality_classifier.py`

b) See `cs336_data/quality_classifier.py`

3 Deduplication

Problem (`exact_deduplication`): 3 points

a) See `cs336_data/exact_deduplication.py`

Problem (`minhash_deduplication`): 8 points

a) See `cs336_data/minhash_deduplication.py`

4 Leaderboard: filter data for language modeling

Problem (`filter_data`): 6 points

a) See `cs336_data/leaderboard/*`

Proportion of examples removed at each step:

Step	Removed (% of total)	Removed (% of remaining)
p(English) > 0.85	85.97	85.97
Gopher filters	3.26	23.25
Exact line deduplication	2.03	18.85
Validation set classifier	6.89	78.84

Table 2: Proportion of examples removed at each step

b) Time to filter 5,000 CC WET files:

Step	Time (minutes)
p(English) > 0.85 + Gopher filters	475
Exact line deduplication	30
Validation set classifier	22
Total	527

Table 3: Time to filter 5,000 CC WET files

Estimated time to filter 100k CC WET files: 10,540 minutes (~176 hours or ~7.3 days)

Problem (`inspect_filtered_data`): 4 points

a) Positive Example 1:

- Quote:

  > ...His research laboratories have trained medical students, residents, and fellows in these MRI techniques and trained MS and PhD students from biomedical, electrical, and computer science engineering to become the next generation of MR physicists. Dr. Faro has an international reputation as an expert neuroradiologist and pioneer in clinical functional neuroradiology...

- Comment:

  This is a high-quality example that is certainly worthwhile to use for language modeling in this context. It is well-written English that is both structurally and semantically sound.

Positive Example 2:

- Quote:

> ...With Sanias serve broken after a long sequence of deuce and advantage points, the Americans began hooping around the court energetically.Rajeev kept his end of the bargain up by locking out the Indians from any opportunities to break his serve. His fluid single hander backhand was in fine working fettle on the night and provided some sparkling moments with passes past desperate racquets.Venus, too, had transformed by this time into the player who owns seven singles and sixteen Grand Slam doubles titles....

- Comment:

  This example is not ideal, though is probably still useful. There are some misspellings (e.g. “hooping” instead of “hopping”), some formatting errors, and it is not quite a semantically coherent as the previous one.

Positive Example 3:

- Quote:

> ...Once children are in school, they spend one year in sheltered classes focused on learning German. Students who do not have basic literacy spend an additional year in those classes. After these 1-2 years of sheltered instruction, students are integrated into mainstream classes. This sheltered approach is relatively uniform across Germany. With the large numbers of students and the focus on sheltered instruction in Germany, we wondered what Canadian educators could learn from the German situation. ...

- Comment:

  This is another high-quality example. It seems as structurally and semantically sound as the first, but covers different topics, and I would certainly want to train on it in this context.

Positive Example 4:

- Quote:

> ...Welcome to Zivame! You will receive a one-time SMS to download the app Phone number: Send me the app By providing your phone number, you agree to receive a one-time automated text message with a link to get the app. Standard messaging rates may apply. ...

- Comment:

  This is a fairly low-quality example. It is essentially spam and is not entirely coherent. Training on this is not ideal.

Positive Example 5:

- Quote:

> ...The finalists will then submit their full dissertations for further review and selection. Obligations for Finalists Being selected as a finalist is an honor. Finalists are required to present their dissertations at a specially designed session during the 2018 AOM Annual Meeting in Chicago, Illinois....

- Comment:

  This is another mostly well-formed, coherent example covering a different subject area, and I would be happy to train on this.

b) Negative Example 1:

- Quote:

> ...Herzlich willkommen - Zügelkönig - Umzug in Zürich Willkommen Umzug Privatumzug Firmenumzug Umzugscheckliste Entsorgungen Reinigung Wohnungsreinigung Endreinigung Firmen Fassadenreinigung Bewertungen Preise Über uns Team Kontakt AGB's Herzlich willkommen Home Ihre Offerte in 2 Minuten !....

- Comment:

  This example (in German) was appropriately removed by the language classifier.

Negative Example 2:

- Quote:

> ...Registration fees for Transfer Up to $85,000 $210.30 $85,001 - $120,000 $220.30 $120,001 - $200,000 $240.30 $200,001 - $300,000 $260.30 $300,001 - $400,000 $280.30 $400,001 - $500,000 $300.30 $500,001 - $600,000 $320.30 $600,001 - $700,000 $340.30 $700,001 - $800,000 $360.30 $800,001 - $900,000 $380.30 $900,001 - $1,000,000 $400.30 $1,000,001 - $1,100,000 $420.30 $1,100,001 - $1,200,000 $440.30 $1,200,001 - $1,300,000 $460.30 $1,300,001 - $1,400,000 $480.30 $1,400,001 - $1,500,000 $500.30 $1,500,001 - $1,600,000 $520.30 $1,600,001 - $1,700,000 $540.30 $1,700,001 - $1,800,000 $560.30 $1,800,001 - $1,900,000 $580.30 $1,900,001 - $2,000,000 $600.30 Over $2,000,000 - $600.30 plus $20 for every $100,000 or part thereof.....

- Comment:

  This examples was removed by the Gopher filters, for having too few tokens with at least one alphabetic character. In this context (seeking to minimize validation loss on c4_100_domains from Paloma), this seems reasonable, as there's little semantic or structural learning to be done from this example.

Negative Example 3:

- Quote:

> Sign in to your account

- Comment:

This is a line that was removed during exact line deduplication. This makes sense, as the line is common boilerplate, has little semantic or structural information content, and is not representative of the target domains.

Negative Example 4:

- Quote:

  > This site requires JavaScript to run. Please make sure you are using a modern browser and JavaScript is activated. Horned Frogs Connect

- Comment:

  This example was removed by the Gopher filters for having too few tokens. This seems appropriate, particularly in the context of minimizing validation loss on c4_100_domains from Paloma. The example has low informational content and is not representative of the target domains.

Negative Example 5:

- Quote:

  > ...You are unable to access this email address duroujian.shop The website from which you got to this page is protected by Cloudflare. Email addresses on that page have been hidden in order to keep them from being accessed by malicious bots. You must enable Javascript in your browser in order to decode the e-mail address. If you have a website and are interested in protecting it in a similar way, you can sign up for Cloudflare. How does Cloudflare protect email addresses on website from spammers? Can I sign up for Cloudflare? Cloudflare Ray ID: 936c92ebbe6e0809 • Your IP: Click to reveal 18.97.14.83 • Performance & security by Cloudflare

- Comment:

  This example was removed by the fastText classifier that I trained using my filtered subset as negative examples, and the validation set as positive examples. This is interesting because the example looks structurally acceptable, but semantically is not that useful for modeling the target distribution. Though no conclusions can be drawn from one example, more such examples would indicate that the classifier has actually learned to distinguish usefully between the target distribution and random examples from my filtered subset.

c) Some observations and subsequent changes:

- The overwhelming majority (>95%) of docs in the validation set are classified as English with $p > 0.85$. Because the language classifier is quite efficient, I took advantage of this fact to fairly quickly filter out CC docs that don't meet this criterion, which turns out to be >85% of them.

- Almost all docs in the validation set pass the Gopher quality filters, but many raw CC docs don't (including 25% of those classified as English with p > 0.85). I therefore used the Gopher filters in my initial round of filtering.
- Using the NSFW and toxicity classifiers on all docs with p(English) > 0.85 slows down filtering by 30-40%. As a proportion, there are almost no documents that are English, pass the Gopher filters, and are still NSFW or toxic. In a setting where we really cared about harmfulness in downstream interactions, using these filters might be worth it. In this setting, I dropped them.
- My final classifier (filtered subset as negatives; validation set as positives) seems to work fairly well. After inspecting randomly sampled examples in different score ranges (e.g. >0.9, >0.8, etc.), the subjective quality does appear to degrade roughly as the score decreases. As a result, I chose to bucket examples by their scores, and repeat examples some number of times, based on the score (higher score -> more repeats; details in the leaderboard section).

Problem (`tokenize_data`): 2 points

See `cs336_data/leaderboard/05-tokenize.py`

Number of tokens in produced dataset: **6,377,480,404**

Problem (`train_model`): 2 points

Best validation loss: **3.288** (cluster crashed after **68k steps**)

Associated learning curve:

WandB Report°

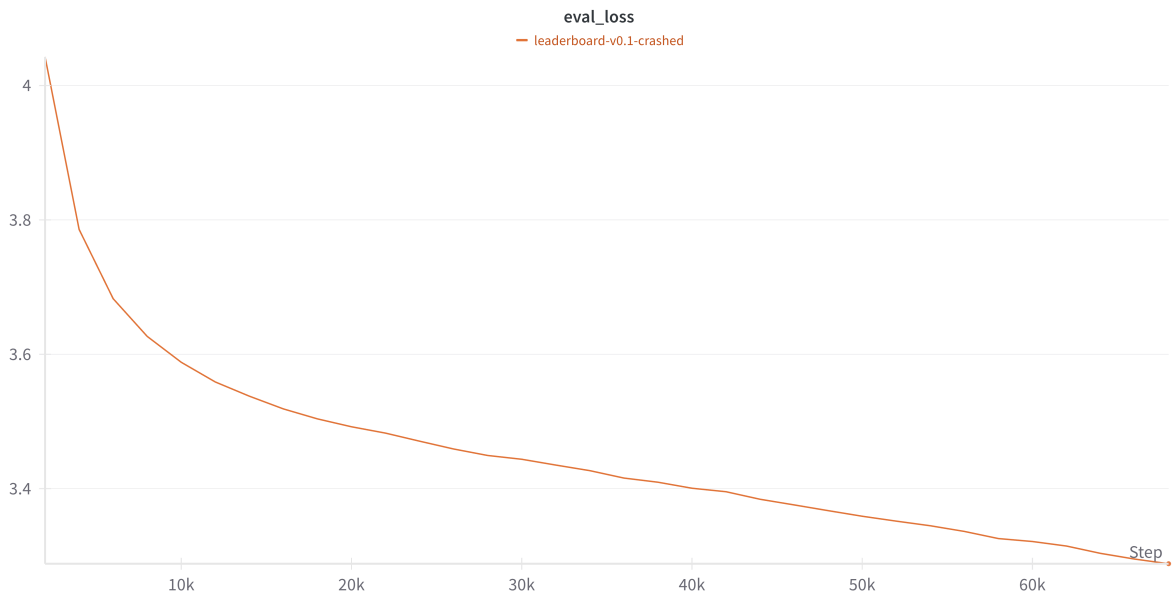

Figure 1: Leaderboard Run — Eval Loss Curve

What I did:

- Pass 1: p(English) > 0.85

    - Used the `lid.176.bin` fastText classifier
    - Discarded ~86% of docs (100% → 14.03%)
- Pass 2: heuristics (from the C4 and Gohper papers)
    - C4 heuristics:
        - Discard entire documents containing "lorem ipsum" or "{"
            - There's a lot of JavaScript on the web, and "{" is rare in non-code pages
        - Remove lines with fewer than 5 words
        - Remove lines ending with non-punctuation terminators (valid: `.`, `!`, `?`, `"`, `'`)
        - Remove lines containing blacklisted boilerplate (e.g. "javascript", "privacy policy")
            - See appendix for the full list
        - Discard documents from which all lines were removed
    - Gopher heuristics:
        - Discard documents with fewer than 50 tokens
        - Discard documents with more than 100,000 tokens
        - Discard documents with a mean token length less than 3 or greater than 10
        - Discard documents with more than 30% of lines ending with "..."
        - Discard documents with fewer than 80% of tokens containing an alphabetic character
    - Result: Discarded ~56% of remaining data (14.03% → 6.16%)
- Pass 3: exact line deduplication
    - Discard all occurrences (including the first) of any duplicated line, corpus-wide
    - Discarding documents that now contain <50 words
    - Discarded ~35% of remaining docs (6.16% → 3.99%)

Training a fastText classifier:

I then trained a fastText classifier using examples from this filtered subset (~4% of the raw CC data) as the negative examples, and documents from the validation set as positive examples. I used 13,500 documents from each class (i.e. all the documents from the validation set, holding out 500 for the classifier's own validation set).

I tweaked fastText parameters, tried using more examples from the (abundant) negative class (3:1 and 5:1 negative:positive ratio), and separately tried duplicating the positive examples to balance the classes (1:1 ratio, but with positives repeated). Nothing worked better than simply using balanced classes and the default fastText parameters, aside from an increased learning rate (lr=0.2).

The best accurary I could achieve with the classifier was 0.81 on the validation set. I surmise that, while more would have helped, the distributions of the two classes were already not massively different after the above filtering, so learning to distinguish well in the constrained-data regime was tough.

Applying the classifier:

I estimated that we'd require ~6.5B training tokens based on the training configuration, and sought the highest-quality subset of that size. I tested two approaches:

- Single-threshold: I picked a threshold for the score associated with the positive class that would keep ~6.5B tokens (which turned out to be 0.09), and retained examples that passed the threshold.
- Multi-threshold: I repeated higher-scoring examples in the training set, as follows:
  - >0.84: 4x
  - >0.72: 3x
  - >0.58: 2x
  - >0.36: 1x

  The thresholds were chosen to (i) roughly balance the number of examples passing each threshold (counting repeats), (ii) retain ~6.5B total tokens. Truthfully, this is just for lack of a more principled way of doing it.

  I found it interesting that, in the final run, the training loss and validation loss were within noise distance of each other from start to finish. I think this suggests that (i) the distributions are similar, (ii) the repetition of examples with high scores did not cause problematic overfitting.

Appendix

C4 line-level blacklist

All lines containing any of the following were removed:

- “javascript”,
- “privacy policy”,
- “terms of use”,
- “cookie policy”,
- “uses cookies”,
- “use of cookies”,
- “use cookies”,
- “all rights reserved”,
- “terms and conditions”,
- “copyright ©”,
- “© copyright”,

Short lines (less than 15 words) containing any of the following were also removed:

- “powered by”,
- “designed by”,
- “theme by”,
- “template by”,
- “website by”,

Lines that did not end with a valid terminator (., !, ?, “, ‘) were also removed.

Bibliography

Index of Figures

Index of Tables