Creating Corpus for Georgian Language Modelling

Anonymous ACL submission

Abstract

The effectiveness of modern NLP methods remain contingent upon the availability of extensive and diverse high-quality training datasets. This poses a significant challenge for low-resource languages, among which Georgian stands out as not only low-resource but also remarkably under-researched. In this paper, we address one of the essential elements of this problem - the absence of the well-organized and openly accessible resources for Georgian language modeling. In particular, we introduce a software framework for collecting, cleaning, and organizing data for Georgian LLM training. We also publish an initial version of 37GB of dataset, laying the groundwork for subsequent research in this domain.

1 Introduction

Language modeling has been one of the most fundamental subfields of NLP, especially during the Transformer era (Vaswani et al., 2017), starting with a family of BERT (Devlin et al., 2019) models up to present-day's sophisticated LLMs (Brown et al., 2020; Touvron et al., 2023; Jiang et al., 2023). Arguably, one of the pressing issues we currently confront is the inadequate performance of these models when dealing with low-resource languages (Yong et al., 2024). This challenge stems not only from a scarcity of benchmark datasets, which are pivotal for assessing the capabilities of LMs across diverse downstream tasks but also from the disproportionately low representation of these languages within the training datasets of these models. For instance, existing GPT-3/3.5/4 tokenizers all encode any single Georgian alphabet character using multiple BPE tokens (on the contrary, in English, a single word usually comprises one or few tokens), which would imply certain limitations in performance with regard to Georgian language. See Figure 3 for details.

ა	ბ	გ	დ	ე	ვ	ზ	თ	ი	კ	ლ
a	b	g	d	e	v	z	t	i	k'	l
[a]	[b]	[ɡ]	[d]	[e]	[v]	[z]	[tʰ]	[i]	[kʼ]	[l]
მ	ნ	ო	პ	ჟ	რ	ს	ტ	უ	ფ	ქ
m	n	o	p'	zh	r	s	t'	u	p	k
[m]	[n]	[ɔ]	[pʼ]	[ʒ]	[r]	[s]	[tʼ]	[u]	[pʰ]	[kʰ]
ღ	ყ	შ	ჩ	ც	ძ	წ	ჭ	ხ	ჯ	ჰ
gh	q'	sh	ch	ts	dz	ts'	ch'	kh	j	h
[ɣ]	[qʼ]	[ʃ]	[t͡ʃʰ]	[t͡sʰ]	[d͡z]	[t͡sʼ]	[t͡ʃʼ]	[x]	[d͡ʒ]	[h]

Figure 1: 33 Letters of Modern Georgian Alphabet (Mkhedruli) along with Pronunciation[1]

The Georgian language has its unique alphabet consisting of 33 letters (Figure 1). It's an agglutinative language, featuring numerous inflected nouns and a complex verb conjugation system. The language's unique characteristics mean that benchmarks and methods tailored for English and other well-studied languages may not be directly applicable. While some progress has been made, there is currently no well-organized training/benchmark data or model for Georgian language in this subfield, as far as our knowledge extends. In essence, the majority of the work undertaken in this area is groundbreaking and unprecedented.

We summarize our contributions as follows:

1. We propose a Python framework (based on Datatrove - HuggingFace's data processing framework[2]) for collecting, cleaning, organizing and evaluating unstructured textual data from various publicly available sources. The pipeline consists of source-specific crawlers, metric-based filtering/cleaning steps, deduplication, language detection as well as transliteration logic to normalize Georgian non-unicode encoded texts (e.g. Latin) back to unicode.

[1] https://www.advantour.com/georgia/population/georgian-language.htm

[2] https://github.com/huggingface/datatrove

2. We publish an initial version of 37GB data on HuggingFace platform, ready for other researchers to use in their work, particularly, in language modeling. We found out that on average 90% of our data (see figure 1 in Appendix) is unique from CulturaX's (Nguyen et al., 2023) Georgian subset - which is an extensively cleaned multilingual corpus derived from Common Crawl and used as baseline in this work.

2 Related Work

In this section, we explore related research that has influenced our work and/or from which we drew inspiration. We begin with The Pile (Gao et al., 2020), a large English corpus pivotal in catalyzing the development of open-source LLMs subsequent to the groundbreaking advancements made by OpenAI's GPT series. They introduce various qualitative analysis techniques, such as perplexity-based measures, and considerations of bias and pejorative content, which we have incorporated into our own methodology. Additionally, CulturaX (Nguyen et al., 2023) emphasizes the significance of rigorous data cleaning procedures to ensure the quality of LLMs, a principle we have adopted by implementing some of their metric-based filtering mechanisms in our data pipeline. Furthermore, JAIS (Sengupta et al., 2023) presents an Arabic dataset alongside a comprehensive data collection and cleaning pipeline, paralleling our own approach. It's also noteworthy that efforts have been made towards relatively under-studied and/or low-resource languages, for instance, Turkish (Safaya et al., 2022) and Polish (Rybak et al., 2020).

3 Dataset Creation

The initial phase of data gathering involves manually compiling a selection of prominent Georgian websites hosting publicly accessible data in large quantities (including webpages as well as PDFs). We have developed specific crawlers for those websites. Drawing inspiration from the CulturaX paper, we adopted their metric-based filters, refining and customizing them to better suit the Georgian context. Finally, we've added CulturaX's Georgian subset to our dataset, which only has roughly 10% overlap (url-based deduplication) with our newly collected data. The resulting set is approximately 37GB of clean data, ready for LLM training.

Our data processing pipeline can be summarized using the following steps:

1. Collection - scraping and extracting textual data from different sources;

2. Applying various filters for noisy low-quality data removal;

3. Content deduplication and train/test splitting. Final splits are in JSONL format, each entry represents document (single web page or PDF) as well as metadata (e.g. URL and timestamp).

Below we explain each step in detail.

3.1 Data Collection

3.1.1 Scraping Web

We use website-specific crawlers for each website (Table 1), which take into account HTML layout and only retrieve main text without ads or any irrelevant information. These crawlers extract textual content from urls by adhering to Robots Exclusion Protocol[3] and nofollow[4].

3.1.2 Extracting data from PDF documents

We leverage PyMuPDF[5] library which allows us to extract textual content with font metadata. The challenge is that PDFs can contain Georgian text in many different encodings (see Figure 5), therefore, it's necessary to normalize everything in unicode, leveraging encoding-specific character sets. In order to handle those non-unicode texts, we have developed a method of identifying text encodings using font metadata extracted from PDF files and mapping characters back to unicode encoding. It should be noted that this method only works for text-based PDF files. Even though there were substantial amount of PDF documents containing scanned texts, we decided to discard those, since extracting texts from scanned files requires high-quality Georgian OCR software, which isn't available as far as our knowledge extends.

3.1.3 CulturaX Georgian Subset

Derived from Common Crawl, CulturaX is 6.3 trillion token multilingual dataset, which undergoes extensive cleaning and deduplication to ensure the necessary level for training high-quality LLMs. We use CulturaX's Georgian subset as one of the sources to our data processing pipeline, along with others. Manual inspection shows that resulting

[3] https://en.wikipedia.org/wiki/Robots.txt
[4] https://en.wikipedia.org/wiki/Nofollow
[5] https://github.com/pymupdf/pymupdf

data has higher quality compared to original subset, which is not surprising as our data processing steps are tailored specifically to Georgian language.

3.2 Cleaning and Filtering

3.2.1 Metric-Based Filtering

We compute multiple metrics of the dataset in order to reveal noisy low-quality content and potential issues. Subsequently, we conduct manual analysis of the metric distributions across the documents to establish thresholds (See Figure 2 for values we use). Documents that fall outside specific threshold ranges are excluded. This widely utilized and highly regarded technique has been employed in numerous recent studies (Gao et al., 2020; Sengupta et al., 2023; Nguyen et al., 2023).

Character Category Counts: We count the percentage of Georgian alphabet characters in the documents. We expect around 50% of characters to be Georgian in high-quality documents. High percentage of symbols and punctuations also indicate low-quality texts, such as Javascript code snippets, document formatting spec symbols and math formulas.

Word / Line Count: Simple word / line count in the document. Documents with too many / few lines or words are considered noise and omitted.

Character / Word / Special-Symbol Repetition Ratio: Documents with high repetition of character / word / spec-symbol n-grams usually identify noise such as text formatting symbols or Javascript code snippets.

Stopword Ratio: We've created a Georgian stopword list and used it as an additional filter for low-quality data. Particularly, we drop documents which have stopword count exceeding some threshold. This list is released as part of the data processing pipeline code.

Flagged Word Ratio: Identifying high occurrence of flagged words (e.g. bad language, insults, toxicity) in text allows us to remove pejorative content. Because Georgian is a morphologically rich language characterized by a large number of inflected forms, compiling a comprehensive list of all potential inflections for flagged words and conducting precise word matching poses a considerable challenge. Therefore, we have chosen to utilize substring matching as an alternative approach. While lemmatization would be preferable, the absence of a high-quality open-source solution for Georgian language renders it unfeasible.

Perplexity: We experimented with publicly available KenLM (Heafield, 2011) ngram language model trained on Georgian Wikipedia subset, for filtering documents beyond certain perplexity thresholds. Even though we include this step as part of our data processing pipeline, we currently don't use it, since we couldn't find thresholds which worked well.

Language Detection: Language detection serves as an additional way to filter out non-Georgian texts, if missed by previous steps. For this purpose, we use publicly available FastText (Bojanowski et al., 2017; Joulin et al., 2016) language classifier trained on Wikipedia.

3.2.2 Anonymization

Anonymization is a very important part of data processing to avoid exposing Personal Identifiable Information (PII). We've adopted and modified regular expressions from MST BigScience PII[6] so that it better suits Georgian language. To anonymize content, we replace identified PII with phrases like ‘PI:<PII TYPE>‘.

3.2.3 Character Normalization

Unicode Consortium relatively recently added section for Georgian capital letters[7] (Mtavruli), and in our dataset we encounter some texts containing those letters. Our understanding is that "Mtavruli" letters are mostly used for decorative purposes, so we convert them back to lowercase Georgian letters "Mkhedruli".

3.3 Deduplication and Splitting

Removing duplicate content from the dataset is crucial for high-quality LLM training. In our work, we first employ simple URL matching to make sure merging CulturaX's Georgian subset with rest of the data sources doesn't introduce duplicate documents. Afterwards, we perform MinHashLSH content-aware deduplication on document level. Datatrove library comes with built-in MinHashLSH algorithm, which is the one we use in this work.

We follow a common practice of contemporary work and provide Train / Valid / Test splits (90% / 5% / 5%) of our final dataset, thus making it easier for others to use it for their work.

[6] https://github.com/bigscience-workshop/data-preparation/blob/main/preprocessing/training/02_pii/bigscience_pii_detect_redact.py

[7] https://en.wikipedia.org/wiki/Georgian_Extended

4 Legality of Content

The datasets utilized for this project have been sourced with consideration of copyright law of Georgia. The majority of the datasets used are not subject to copyright, such as parliament records, or are permissively licensed, including content from Wikipedia. It should be noted that data obtained from web crawls may contain copyrighted texts, although current tools do not enable us to comprehensively identify copyrighted texts. In light of this, and recognizing that all utilized sources are already publicly available on the internet, we have made the decision to openly publish the dataset created from this project. However, given the ever-evolving nature of legal frameworks, particularly in the context of AI innovations, we remain prepared to reassess our decision in the future.

5 Conclusion

We have presented a novel data processing pipeline specifically designed for the Georgian language. Leveraging established filtering methods from recent literature and integrating unique features such as non-unicode Georgian text normalization, our approach offers tailored solutions for handling Georgian textual data. Furthermore, we have made publicly available a comprehensive Georgian language corpus, facilitating further advancements in language model training and research within the Georgian language domain.

6 Limitations and Future Work

A primary constraint in the present work lies in the absence of high-quality open-source NLP tools for Georgian language, a factor that significantly impacts the precision of our data collection pipeline. For example, our investigation reveals an absence of high-quality open-source lemmatizer, and the intricate morphological structure of the language poses challenges in identifying flag words with extensive coverage, a critical aspect in the identification of profanity.

It's also important to acknowledge the possibility of various biases in the dataset. For instance, we conducted basic word2vec analogy tests focusing on gender and observed analogous outcomes to those in other datasets (Gao et al., 2020). Specifically, terms like "male" exhibited proximity to words such as politician, professor, and director, while terms like "female" were closely associated with roles like cook and cleaner. Refer to the Figure 4 in the Appendix for further details.

Building an LLM with this data might still pose a challenge, due to absence of relevant benchmarks for testing downstream task performance - which we leave as a future work.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Computation and Language*, arXiv:1607.04606. Version 2.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Computation and Language*, arXiv:2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Computation and Language*, arXiv:2101.00027.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *WMT@EMNLP*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Computation and Language*, arXiv:2310.06825.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *Computation and Language*, arXiv:1607.01759. Version 3.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *Computation and Language*, arXiv:2309.09400.

Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. 2020. KLEJ: Comprehensive benchmark for Polish language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1191–1201, Online. Association for Computational Linguistics.

Ali Safaya, Emirhan Kurtuluş, Arda Goktogan, and Deniz Yuret. 2022. Mukayese: Turkish NLP strikes back. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 846–863, Dublin, Ireland. Association for Computational Linguistics.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Computation and Language*, arXiv:2308.16149. Version 2.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Computation and Language*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Computation and Language*, arXiv:1706.03762.

Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. 2024. Low-resource languages jailbreak gpt-4. *Computation and Language*, arXiv:2310.02446. Version 2.

A Appendix

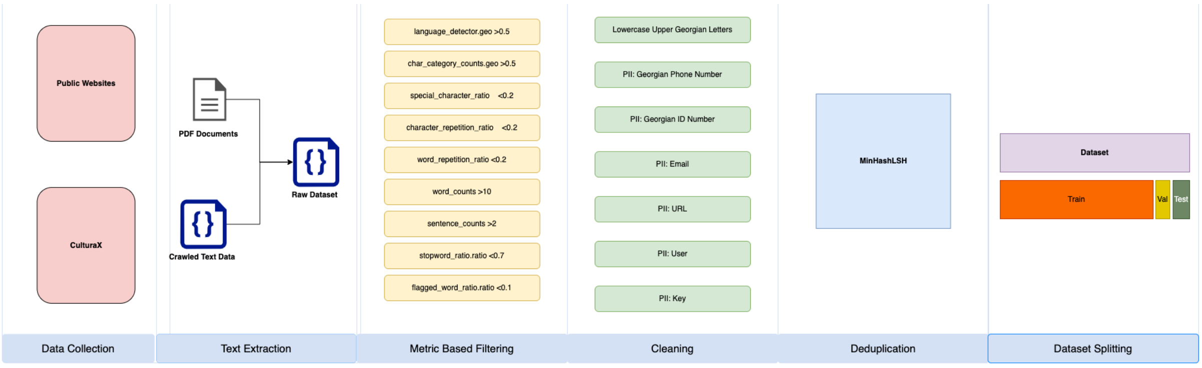

Figure 2: Diagram of our data processing pipeline.

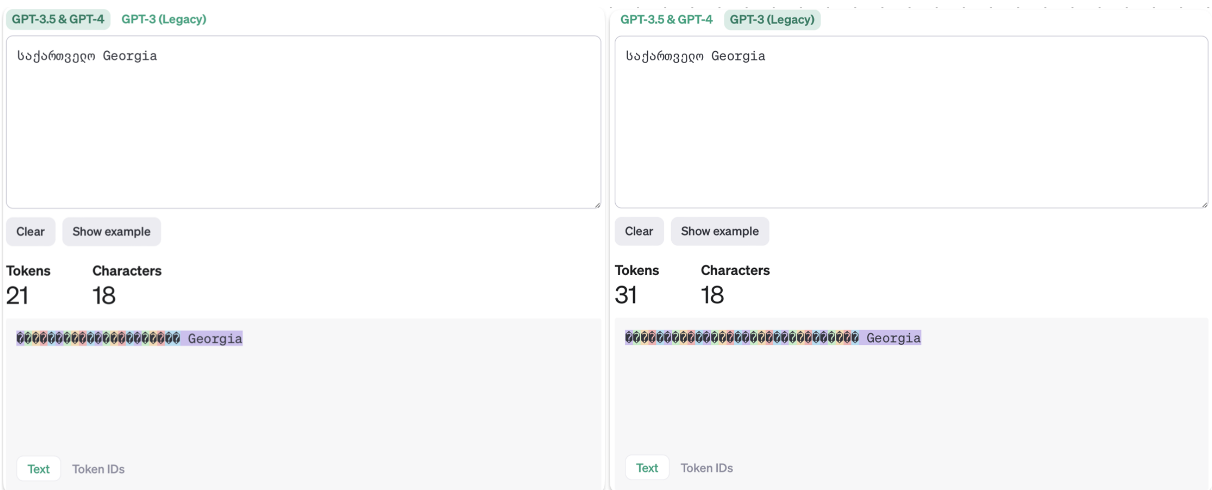

Figure 3: Illustration of inefficient performance of OpenAI's tokenizer on Georgian text. Example text is "Georgia (in Georgian) Georgia (in English)". On the left, we see GPT 3.5/4 tokenizer uses 2 tokens to represent single Georgian character, whereas, on the right, GPT 3 tokenizer uses 3 tokens per single Georgian character. Meanwhile, English word "Georgia" is only a single token. Screen taken from `https://platform.openai.com/tokenizer`

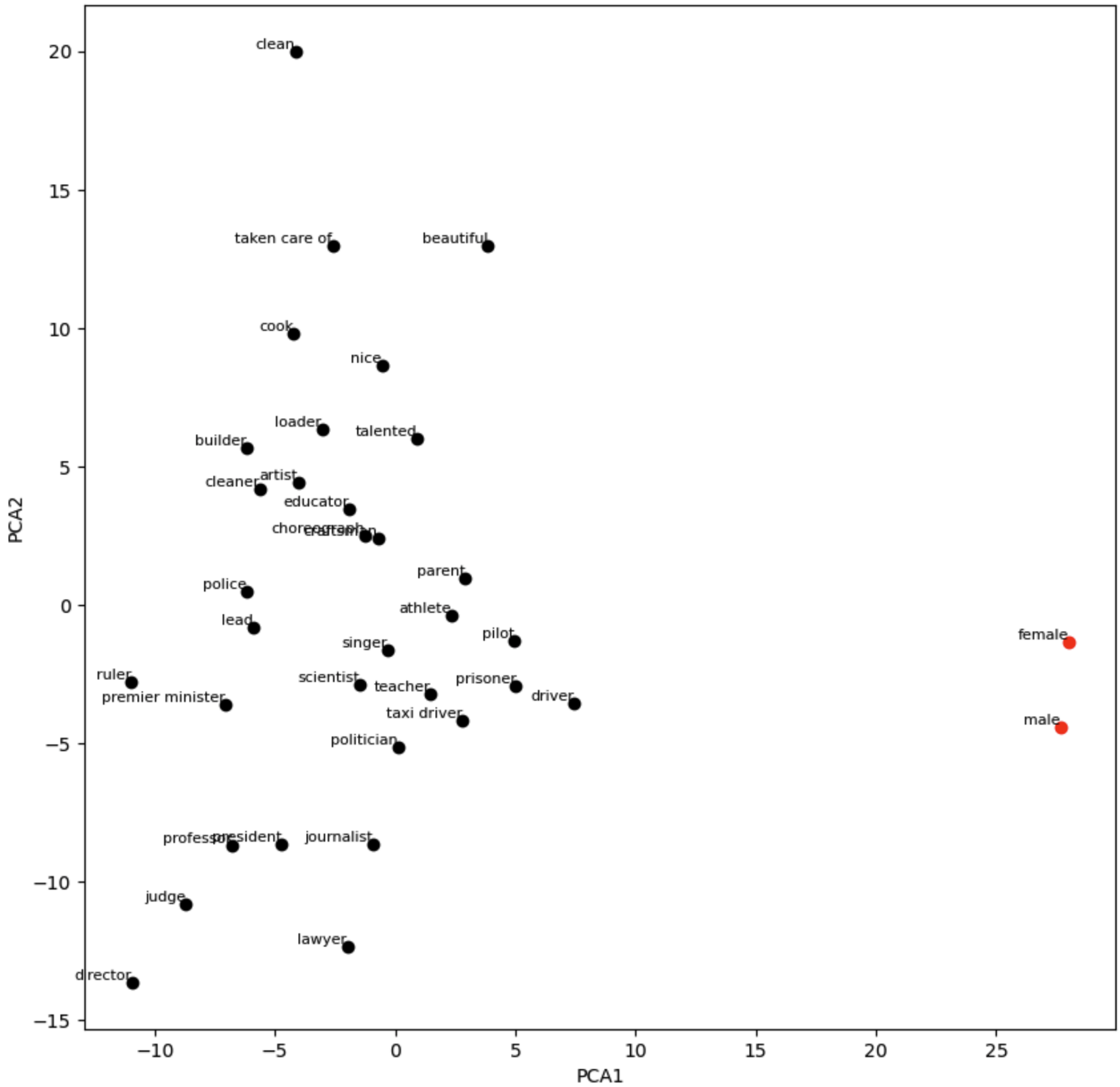

Figure 4: 2D PCA plot of FastText 300 dim word embeddings (labels have been translated from Georgian into English) illustrating gender bias. Reference words "Male" and "Female" are marked in red.

Source	Final Size	Filtered	Unique	Description
1tv-ge	791M	13.77%	87.3%	News (sport, politics, science, business, etc)
4love-ge	638M	30.45%	42.5%	Website with articles related to love and life
ambebi-ge	813M	58.37%	92.4%	News (sport, politics, economics, international news, books, etc)
aversi-ge	47M	75.71%	62.9%	Website of Georgian pharmaceutical company. Contains medical descriptions of various drugs sold by this company.
bm-ge	54M	34.75%	100%	Business news
const-court-ge	128M	35.22%	93.8%	Website for Georgian Constution containing law documents.
ecd-court-ge	1.3G	36.77%	100%	Documents on court decisions.
ecd-court-notifications-ge	1.9M	94.61%	87%	Website of Georgian court containing public decision documents
eon-ge	383M	26.01%	100%	Website containing articles, blogs, quizzes and audio books
europop-ge	44M	11.94%	100%	Sport news
gemrielia-ge	12M	6.44%	85.8%	Food recipes
interpressnews-ge	1.1G	32.94%	100%	News (sport, politics, economics, international news, culture, military, etc)
iverieli-ge	6.7G	35.82%	100%	Website of National Parliament Library of Georgia. Contains books in PDF format
kvirispalitra-ge	585M	29.61%	91%	News (sport, politics, culture, military, for women, etc)
lawlibrary-ge	126M	47.71%	100%	Website with books related to law
matsne-ge	267M	67.09%	99.2%	Website with law related news documents
mkurnali-ge	153M	16.46%	76.2%	Articles about medicine and health.
mshoblebi-ge	128M	21.02%	100%	Articles related to parents and children
mtavari-ge	239M	9.34%	98.9%	News (sport, politics, science, business, etc)
netgazeti-ge	325M	19.73%	73.6%	News (sport, politics, art, etc)
ombudsman-ge	146M	78.64%	96.7%	News related to Georgian ombudsman
on-ge	339M	18.44%	70%	News
openscience-ge	731M	6.17%	100%	Georgian science website with publications in PDF format
parliament-ge	16M	73.3%	100%	Website of Georgian Parliament with small news articles
parliament-library	800M	81.6%	100%	PDF documents about law and legal context
mc4	18G	8.04%	0%	CulturaX's Georgian Subset
OSCAR-2019	920M	4.65%	0%	CulturaX's Georgian Subset
OSCAR-2301	1.2G	16.62%	0%	CulturaX's Georgian Subset
OSCAR-2109	506M	5.73%	0%	CulturaX's Georgian Subset
OSCAR-2201	480M	17.41%	0%	CulturaX's Georgian Subset

Table 1: List of source domains (bottom 5 are from CulturaX) that we used for collecting Georgian text. Column "Final Size" tells final data size in MB or GB. Column "Filtered" shows what percentage of original RAW data has been dropped after filtering and deduplication. Column "Unique" tells percentage of URLs which are only found in our data and not in CulturaX.

Unicode Georgian Character Code	Unicode Georgian Character	"BalavMtavr" Code	"BalavMtavr" Character	"GeoTimes" Code	"GeoTimes" Character	"Literaturuly" Code	"Literaturuly" Character	"Sylfaen" Code	"Sylfaen" Character
4304	ა	97	a	192	À	102	f	4304	ა
4305	ბ	98	b	193	Á	44	,	4305	ბ
4306	გ	103	g	198	Æ	117	u	4306	გ
4307	დ	100	d	195	Ã	108	l	4307	დ
4308	ე	101	e	196	Ä	116	t	4308	ე
4309	ვ	118	v	216	Ø	100	d	4309	ვ
4310	ზ	122	z	220	Ü	112	p	4310	ზ
4311	თ	84	T	225	á	39		4311	თ
4312	ი	105	i	201	É	98	b	4312	ი
4313	კ	107	k	203	Ë	114	r	4313	კ
4314	ლ	108	l	204	Ì	107	k	4314	ლ
4315	მ	109	m	205	Í	118	v	4315	მ
4316	ნ	110	n	207	Ï	121	y	4316	ნ
4317	ო	111	o	208	Ð	106	j	4317	ო
4318	პ	112	p	209	Ñ	103	g	4318	პ
4319	ჟ	74	J	222	Þ	59	;	4319	ჟ
4320	რ	114	r	211	Ó	104	h	4320	რ
4321	ს	115	s	212	Ô	99	c	4321	ს
4322	ტ	116	t	214	Ö	110	n	4322	ტ
4323	უ	117	u	215	×	101	e	4323	უ
4324	ფ	102	f	197	Å	97	a	4324	ფ
4325	ქ	113	q	210	Ò	109	m	4325	ქ
4326	ღ	82	R	223	ß	113	q	4326	ღ
4327	ყ	121	y	219	Û	46	.	4327	ყ
4328	შ	83	S	224	à	105	i	4328	შ
4329	ჩ	67	C	221	Ý	120	x	4329	ჩ
4330	ც	99	c	32		119	w	4330	ც
4331	ძ	90	Z	228	ä	115	s	4331	ძ
4332	წ	119	w	217	Ù	111	o	4332	წ
4333	ჭ	87	W	227	ã	122	z	4333	ჭ
4334	ხ	120	x	218	Ú	91	[	4334	ხ
4335	ჯ	106	j	202	Ê	93	]	4335	ჯ
4336	ჰ	104	h	200	È	47	/	4336	ჰ

Figure 5: Character and code mappings for unicode as well as some of the other popular encodings for Georgian alphabet.