Application Project in Data Engineering and Analytics

Generative Language Models for Opinion Mining

Research Group Social Computing

Department of Informatics, Technical University of Munich (TUM)

Author	Shyam Arumugaswamy
Supervisor	PD Dr. Georg Groh
Advisor	Gerhard Hagerer
Date	Munich, October 18, 2019

Contents

1 Introduction

Data is required to create efficient word representation models for problems related to Natural Language Processing (NLP), Data and Text Mining. In particular, annotated data for specific domains is sparse as annotating is often expensive and time consuming, resulting in less data for training. The need for data augmentation techniques has increased due to more complex modelling techniques, such as Deep Neural Networks which require large corpus data, especially for languages other than English, in training. Generative Adversarial Network (GAN) has shown breakthroughs in many domains but text generation. Language Modelling (LM) is one of the major components of NLP and finds applications in various areas like text generation, machine translation, spell correction, speech recognition, summarization, question answering, sentiment analysis, etc. Language models assign probabilities to a sequence of words. N-gram model is a sequence of N words that assign probabilities to entire sequence and can also estimate probability of last word of an n-gram given the previous words. Language model is essential to represent the text to a form understandable from the machine point of view. One of them, Generative Pre-Training 2 (GPT-2), has received special attention, as it achieved state-of-the-art performance on many language modelling benchmarks without task-specific training. In this work, we wanted to examine whether GPT-2 can be applied to raw unannotated organic dataset and generate customer reviews according on pre-defined opinion . Our task consisted of the following steps:

1. Train the pre-trained GPT-2 language model on the raw organic dataset
2. Finetune the trained model on annotated organic dataset
3. Generate reviews on belief statements for pre-defined opinion
4. Evaluate the generated reviews

Section 2 talks about the related work primarily used in the project. In Section 3, the methodology and the theory is described. For a description of the organic dataset and belief statements, see Section 4. The practical experiments are illustrated in Section 5, with results in Section 6.

2 Related Work

The basis of the project is primarily built on the work of Radford et al. In their paper "Language Models are Unsupervised Multitask Learners" [1], they demonstrated that high-capacity language model GPT-2 (see Section 3.2.4) trained on large and diverse text corpus outperforms other language models trained on specific domains (like Wikipedia, news, or books) without any explicit supervision.

Instead of using any existing dataset, the authors built a new web scrape comprising of the text from outbound link from Reddit posts which were rated with at least 3 karma, thus emphasizing on document quality.

GPT-2 model is enhancement of GPT model (see Section 3.2.3) which is based on Radford et al. work in "Improving Language Understanding by Generative Pre-Training" [2]. The authors built on "Semi-supervised Sequence Learning" [3] by Dai et al., which improved the performance in document classification by using unsupervised pre-training of an Long Short-Term Memory (LSTM) followed by supervised fine-tuning. The target tasks need not be in same domain of the unlabelled corpus. It also extended Howard et al. "Universal Language Model Fine-tuning for Text Classification" research [4] (ULMFiT) (see Section 3.2.1) that showed how a single dataset-agnostic LSTM language model can be fine-tuned to get state-of-the-art performance on a variety of document classification datasets. GPT uses a Transformer-based model of this approach to succeed at a broader range of tasks beyond document classification, such as commonsense reasoning, semantic similarity, and reading comprehension.

GPT is also similar to but more task-agnostic than Embeddings from Language Models (ELMo) (see Section 3.2.2), which is based on Peters et al. work in "Deep contextualized word representations" [5], incorporates pre-training but uses task-customized architectures to get state-of-the-art results on a broad suite of tasks. ELMo uses a shallow concatenation of independently trained left-to-right and right-to-left multi-layer LSTMs, while GPT is a multi-layer transformer decoder [6]. The use of contextualized embeddings in downstream tasks are different: ELMo feeds embeddings into models customized for specific tasks as additional features, while GPT fine-tunes the same base model for all end tasks.

The focus of the project was to see if the GPT-2 model, trained on diverse Reddit corpus, can efficiently generate reviews when finetuned on organic dataset.

3 Methodology

3.1 Aspect-Based Sentiment Analysis

Sentiment analysis [7] extracts the subjective information from a textual context and classifies it into one of the predefined sets of classes, e.g., positive, negative, and neutral. Aspect extraction detects the part of a document or sentence to which a given sentiment actually refers to. Aspect-Based Sentiment Analysis (ABSA) [8] solves both of these problems jointly. The problems associated with ABSA are that the detection of a subset of aspects occurring in given piece of text is non-trivial. Sometimes, the general sentiment and the sentiment of each aspect can each be completely different from each other. So, the model must be able to distinguish aspects in the text and make independent decisions for each of them.

3.2 Pre-trained Language Models

3.2.1 ULMFiT

ULMFiT uses inductive transfer learning to achieve state-of-the-art results for various NLP task. The LM is trained on a large general-domain corpus to capture generic features of the language in different layers. The full LM is fine-tuned on the target task data using discriminative fine-tuning ('Discr') and slanted triangular learning rates (STLR) to learn task-specific features. The classifier is fine-tuned on the target task using gradual unfreezing. The method is universal and works across tasks varying in document size, number, label type and requires no custom feature engineering/pre-processing and no additional in-domain documents or labels.

3.2.2 ELMo

ELMo is deep contextualized word representation that models both complex characteristics of word use (e.g., syntax and semantics), and how these uses vary across linguistic contexts (i.e., to model polysemy). These word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. They can be easily added to existing models and significantly improve the state-of-the-art across a broad range of challenging NLP problems, including question answering, textual entailment, and sentiment analysis.

3.2.3 GPT

1. **Unsupervised pre-training**

   Given unsupervised corpus of tokens $U = \{u_1, ..., u_n\}$, maximize the following likelihood of the objective:

   $$L_1(U) = \sum_i \log P(u_i | u_{i-k}, ..., u_{i-1}; \Theta) \tag{1}$$

   GPT model applies multiple transformer blocks over the embeddings of input sequences. Each block contains a masked multi-headed self-attention layer and a pointwise feed-forward layer, described in Figure 1. The final output produces a distribution over target tokens after softmax normalization.

   $$h_0 = UW_e + W_p \tag{2}$$

   $$h_l = transformer\_block(h_{l-1}) \forall i \in [1, n] \tag{3}$$

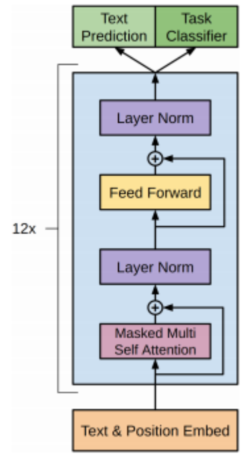


Figure 1: Architecture of GPT. Figure taken from [2]

$$P(u) = softmax(h_n {W_e}^T) \tag{4}$$

where n is the number of layers, W_e is token embedding matrix and W_p is position embedding matrix.

2. **Supervised fine-tuning**

   The parameters to the supervised target task are adapted after training the model with objective in (Eq. (1)). In the labeled dataset, each input has n tokens, $x_1, \ldots, x_n$, and a label y. GPT first processes the input sequence x through the pre-trained transformer decoder to obtain final block's activation ${h_l}^m$, which is then added to linear output layer with parameters W_y to predict y

$$P(y|x^1, ..., x^m) = softmax({h_l}^m W_y) \tag{5}$$

   This gives following objective to maximize:

$$L_2(C) = \sum_{(x,y)} \log P(y|x^1, ..., x^m) \tag{6}$$

   Adding the LM loss as an auxiliary loss is found to be beneficial, because it helps accelerate convergence during training and it is expected to improve the generalization of the supervised model. We optimize the following objective (with weight λ)

$$L_3(C) = L_2(C) + \lambda * L_1(C) \tag{7}$$

3.2.4 GPT-2

GPT-2, a direct scale up of GPT, is large transformer-based language model with 1.5 billion parameters, trained on a dataset of 8 million web pages to predict the next word, given all the previous words within some text. GPT-2 zero-shots to state-of-the-art performance on 7 out of 8 tested language modeling datasets without explicit supervision and outperforms models trained on domain-specific datasets. The model generates conditional synthetic text samples of unprecedented quality, when primed with an arbitrary input. It adapts to the style and content of the conditioning text and allows user to generate realistic and coherent continuations about a topic of their choice.

Network Architecture

GPT-2 leverages transformer model similar to GPT, with few minor modifications [9]

- Moving normalization layer to the input of each sub-block
- Adding normalization layer after final self-attention model
- Modifying the initialization as a function of the model depth
- Scaling the weights of residual layers by a factor of $1/\sqrt{N}$ where N is the number of residual layers
- Using larger vocabulary size and context size

Four models of GPT-2 with different parameters were trained as shown in Table 1, out of which only 117M and 345M models have been released due to concerns about malicious applications.

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 1: Architecture hyperparameters [1]

Byte Pair Encoding (BPE)

GPT-2 neither uses word level or character level embeddings but uses BPE on UTF-8 byte sequences. In BPE, a list of subword will be calculated by using following algorithm (see Listing 1):

- Split word to sequence of characters
- Joining the highest frequency pattern

- Keeping doing previous step until it hit the pre-defined maximum number of sub-word of iterations

```
import re, collections

def get_stats(vocab):
    pairs = collections.defaultdict(int)
    for word, freq in vocab.items():
        symbols = word.split()
        for i in range(len(symbols)-1):
            pairs[symbols[i],symbols[i+1]] += freq
    return pais

def merge_vocab(pair, v_in):
    v_out = {}
    bigram = re.escape(' '.join(pair))
    p= re.compile(r'(?<!\S)' + bigram + (r'(?<!\S)')
    for word in v_in:
        w_out = p.sub(''.join(pair), word)
        v_out[w_out] = v_in[word]
    return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>' : 6, 'w i d e s t </w>' : 3}
num_merges = 10

for i in range(num_merges):
    pairs = get_stats(vocab)
    best = max(pairs, key=pairs.get)
    vocab = merge_vocab(best, vocab)
    print(best)
```

Listing 1: Algorithm of BPE [10]

Each byte can represent 256 different values in 8 bits, while UTF-8 can use up to 4 bytes for one character, supporting up to 231 characters in total. Therefore, with byte sequence representation we only need a vocabulary of size 256 and do not need to worry about pre-processing, tokenization, etc.

BPE merges frequently co-occurred byte pairs in a greedy manner to prevent it from generating multiple versions of common words (i.e. dog., dog! and dog? for the word dog), GPT-2 prevents BPE from merging characters across categories (thus dog would not be merged with punctuations like ., ! and ?). This tricks help increase the quality of the final byte segmentation. Using the byte sequence representation, GPT-2 is able to assign a probability to any Unicode string, regardless of any pre-processing steps.

3.3 Evaluation

To see how good the GPT-2 generated texts are, we use micro-F1 score as evaluation metric. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The formula for the F1 score is:

$$F1 = 2 * (precision * recall)/(precision + recall) \tag{8}$$

In micro, the metrics are calculated globally by counting the total true positives, false negatives and false positives. So, precision, recall, F1 score are equal when using micro-averaging in multi-class problem.

4 Data

4.1 Organic dataset

The organic dataset is a collection of multilingual text corpora composed of opinions from diverse websites, forums, blogs regarding organic-food related topics stored in .csv and json files.

Language	Type	Resource	No. articles	No. total	No. total comments	No. relevant comments
English	Biased	-	5	306037	306036	75866
	Unbiased	Forums	5	107599	106893	26049
		Newssites	7	101711	101711	26491
German	Biased	Blogs	11	5791	5791	1628
		Forums	2	374	362	306
	Unbiased	Forums	11	25856	25836	12654
		Newssites	29	195429	191745	89186

Table 2: Statistics for the raw organic dataset (Only comments above 15 characters are considered)

The organic dataset comprises of two subgroups: raw and annotated. The raw dataset is separated into two subgroups based on language: biased and unbiased. The biased corpus comprises of data from social media sources such as blogs, forum which tends towards biased opinion, while the unbiased comprises mostly of news articles. Each of these comments has various keys like *comment_id*, *comment_text*, *comment_replyTo*, *relevant*. The experiments are conducted only on relevant comments which have *relevant* key as one in the dataset. *comment_replyTo* key is useful when modelling dialogue conversation from this dataset. Table 2 depicts information about the English and German raw organic datasets. The annotated dataset comprises of the aspects, entities and sentiment of each sentence. For more detailed description of the data, one can refer to the original repository [11].

4.2 Belief statements

This section is based on Danner et. al work in "Consumer Beliefs on Organic Food: An Exploratory Analysis of Online Comments in German-Speaking Countries and the US" [12]. The authors conducted content analysis on online comments to identify and categorize consumer beliefs towards organic food in German-speaking countries and the US. They identified 66 positive and negative labels, broadly categorized into four main themes: product attributes, production, authenticity and food system as shown in Table 3. People in German-speaking countries and US hold similar beliefs but their relative importance to the themes differ. The results suggest that further studies on organic food consumption should elicit more fine-grained beliefs not only on product attributes and production but also on authenticity and the food system.

In our project, we generate customer reviews keeping these belief statements as a prefix to the GPT-2 trained model.

Main theme	Themes	Beliefs	IsPositive
Product attributes	Food safety	Organic products contain no or less chemical residues	1
Product attributes	Price	Organic products are expensive	0
Production	Environment	Organic food uses no or less chemicals	1
Authenticity	Product category	I prefer Organic products for certain product categories	1
Food system	Food security	Organic food yields can feed the world	1

Table 3: Few belief statements [12]

5 Experiments

This section describes the experiments conducted on the different organic dataset. We use 345M pre-trained GPT-2 model in all the experiments [13]. The code implementations and results can be found in the Gitlab repository [14].

5.1 Data Preparation

5.1.1 Raw Organic Dataset

Dataframe Creation

The raw organic dataset which is saved as JSON is converted into a dataframe format to facilitate convenient processing of the data for experiments. Dataframe was created for each of the categories i.e. blogs, forums, newssites which were later concatenated based on language. Each of these dataframes had common column format : *file_name*, *article_author_id*, *article_title*, *comment_id*, *comment_author_id*, *comment_author_name*, *comment_text*, *comment_rating*, *comment_replyTo*, *resource_type*, *relevant*, *is_conversation*. Column *is_conversation* is set as one if *comment_replyTo* is present for a given *comment_id* and this can be used for training dialogue conversations. We use only English comments for training in the experiments.

Preprocessing

The English comments from both biased and unbiased raw dataset were preprocessed to form the training corpus. Following steps were carried out:

1. Selecting only relevant comments (*relevant* = 1)
2. Replacing multi occurring punctuations with single (eg. ,,,, replace with ,)
3. Replacing Devanagari(Hindi scripts) with white space
4. Removing contractions (compiled in dictionary)
5. Correcting spellings of social and organic relevant terms (compiled in dictionary)
6. Replacing unicode characters (compiled in dictionary)
7. Removing weblinks
8. Curating language (see Appendix A)
9. Filtering out comments with less than 15 characters

These comments were then saved to text file with one comment per line and used for training.

5.1.2 Annotated Organic Dataset

Dataframe Creation

The annotated organic dataset which is saved as JSON is split into training, validation and test datasets and has information on Entity (*Organic products : p*, *conventional farming : cf* etc), Sentiment (*positive : p*, *negative : n* , *neutral : 0*), Attribute (*general : g*, *price : p* etc) [15]. Each of these datasets are converted into a dataframe format filtered upon Entity and Sentiment to facilitate convenient processing of the data for experiments (see Table 4) (eg. Entity : *p* and Sentiment : *p* for finetuning model for positive sentiment in organic products). Each of these dataframes had common column format : *Sentence*, *Sentiment*.

Entity	Sentiment	Count
c	0	104
	n	80
	p	105
cc	0	9
	n	12
cf	0	70
	n	100
	p	34
cg	0	19
	n	15
	p	7
cp	0	93
	n	145
	p	60
f	0	347
	n	214
	p	288
g	0	582
	n	253
	p	258
gg	0	116
	n	86
	p	55
p	0	472
	n	470
	p	693

Table 4: Entity and Sentiment distribution of annotated training dataset [15]

Preprocessing

The *Sentence* present in training, validation and test datasets were preprocessed as per the following steps:

1. Removing contractions (compiled in dictionary)
2. Correcting spellings of social and organic relevant terms (compiled in dictionary)
3. Replacing unicode characters (compiled in dictionary)
4. Filtering out duplicate sentences.

These comments were then saved to text file with one comment per line and used for processing.

5.2 Training

5.2.1 Unsupervised Training

The preprocessed raw dataset was trained on the pre-trained 345M GPT-2 model for 1000 steps as shown in (see Listing 2)

```
import gpt_2_simple as gpt2

model_name = "345M"
file_name = 'english_filtered_comments.txt'
op_file_name = model_name + "english_filtered_comments"

gpt2.download_gpt2(model_name=model_name)

sess = gpt2.start_tf_sess()
gpt2.finetune(sess,
          dataset=file_name,
          model_name=model_name,
          steps=1000,
          restore_from='fresh',
          run_name=op_file_name,
          print_every=10,
          sample_every=200,
          save_every=500
          )
```

Listing 2: Unsupervised training [13]

5.2.2 Supervised Fine-tuning

The preprocessed raw dataset was finetuned on the trained raw dataset (see Section 5.2.1) for another 1000 steps. Listing 3 refers to finetuning the *Organic products* for positive sentiment.

```python
import gpt_2_simple as gpt2
model_name = "345M"
file_name = 'english_filtered_comments.txt'
finetune_file = 'fine_tune_organic_products_positive'
op_file_name = model_name + "english_filtered_comments"
finetune_file_name = finetune_file + '.txt'

#Restoring the model from the unsupervised training
gpt2.copy_checkpoint_from_gdrive(run_name = op_file_name)

sess = gpt2.start_tf_sess()
gpt2.finetune(sess,
          dataset=finetune_file_name,
          model_name=model_name,
          steps=2000,
          restore_from='latest',
          run_name=op_file_name,
          print_every=50,
          sample_every=500,
          save_every=1000,
          overwrite = True
          )
```

Listing 3: Supervised fine-tuning [13]

5.3 Customer Reviews Generation

Belief statements(see Section 4.2) were used as prefix for the GPT-2 trained model (see Section 5.2.2) and customer reviews of length minimum 100 words were generated for *full organic* (which included *Organic products* , *Organic general* , *Organic farmers*, *Organic companies*) and *Organic products* and each of the sentiments separately. Listing 4 shows reviews generation for positive sentiment of *Organic products*. *top_k* parameter is integer value controlling diversity. 1 means only 1 word is considered for each step(token), resulting in deterministic completions, while 40 avoids mode collapse. Higher the value of *temperature* parameter, more random completions are present. As it approaches zero, model will become deterministic and repetitive.

```python
import pandas as pd
belief_pd = pd.read_excel(Belief_statements.xlsx')

for index, row in belief_pd.iterrows():
output = gpt2.generate(sess, run_name=fine_tune_organic_products_positive,  length=150, return_as_list = True
         top_k = 40,  truncate="", include_prefix = True, prefix = row['Beliefs'])
belief_pd.at[index,'review'] = '.'.join(output[0].split(".")[:-1]) + '.'
```

Listing 4: Customer reviews generation [13]

5.4 Evaluation

5.4.1 Forward training

A softmax classifier was trained and validated on the annotated organic train and validation dataset respectively and tested on GPT-2 generated customer reviews (see Section 5.3).

5.4.2 Reverse training

Multiple customer reviews were generated for every belief statement for each of the sentiments of *Organic products* (see Section 5.3). A softmax classifier was trained on these generated reviews and tested on annotated organic test dataset.

6 Results

6.1 Evaluation on Customer Reviews Generation

On comparing the reviews generated for raw organic dataset finetuned on respective entity sentiments, the ones finetuned on specific *Organic products* were more realistic and sensible than the ones finetuned on *full organic* (see Appendix B). It was more evident in the micro-F1 scores calculated (see Section 6.2).

6.2 Evaluation on Forward Training

Table 5 refers to the confusion matrix plotted for training the softmax classifier on annotated dataset for *full organic* and Table 6 for *Organic products*. Though the negative and neutral sentiments performed closely, the positive sentiments performed better in *Organic products*, which is reflected in the micro-F1 score (calculated as per Eq. (8)).

		Prediction		
		p	n	0
Ground truth	p´	44	12	10
	n´	13	48	5
	0´	22	16	28

micro-F1 : 0.60

Table 5: Confusion matrix for full organic

		Prediction		
		p	n	0
Ground truth	p´	53	8	5
	n´	13	48	5
	0´	26	11	29

micro-F1 : 0.66

Table 6: Confusion matrix for organic products

6.3 Evaluation on Reverse Training

Table 7 refers to the confusion matrix plotted for training the softmax classifier on GPT-2 generated customer reviews for *Organic products*. *Organic products* was chosen over *full organic* for reviews training data generation as the former performed better (see Section 6.2).

		Prediction		
		p	n	0
Ground truth	p´	39	11	33
	n´	2	10	11
	0´	23	6	45

micro-F1 : 0.52

Table 7: Confusion matrix for organic test dataset

7 Discussion and Conclusion

The highly parallelized architecture of transformer in GPT-2 achieved rapid generalization with minimum customization of the model and fine-tuning was achieved in reasonable time. Though the authors claimed GPT-2 model to generalize better irrespective of the dataset, the micro-F1 scores show that model trained on specific entity and sentiment performed marginally better than the generalized one. From micro-F1 scores calculated in Section 6.2 and Section 6.3, we observed that classifier trained on organic dataset comments performed better than the one trained on GPT-2 generated reviews.

We can anticipate exciting applications [16] of GPT-2 as:

- Better speech recognition systems
- More capable dialogue agents
- AI writing assistants
- Unsupervised translation between languages

At the same time, this also opens door for malicious applications like generating misleading news articles, automating the production of spams/fake content. We should consider the fact how generation of synthetic images, videos, audio, and text may unlock new threats and should start preparing better counter measures.

For now, its very exciting time to study deep learning in NLP and the advent of powerful transfer learning language models like GPT-2 will open up more research possibilities.

Appendix

A Language Curation

Some of the comments in the raw organic dataset has mixture of English and Devanagari(Hindi) scripts which need to be curated for efficient utilization of only the English comments for training. The cleansed data is located in *raw_source_data_spell_corrected* folder of the repository [11] with *processed_comment_text attribute*.

Features

- The algorithm involves combination of English dictionary check (with a 50% cutoff threshold) and langdetect python package for blacklisting non-English words.
- Input dataset can be either in form of pandas dataframe or json.
- In the comments with both Latin(English) and Devanagari(Hindi) scripts, Hindi ones in the comment are removed leaving behind the English ones intact
- Hindi comments in Latin script(i.e Hinglish) are removed (though a minority of them will still remain - eg. 17 cases [.06%] as shown in Fig. A.2)

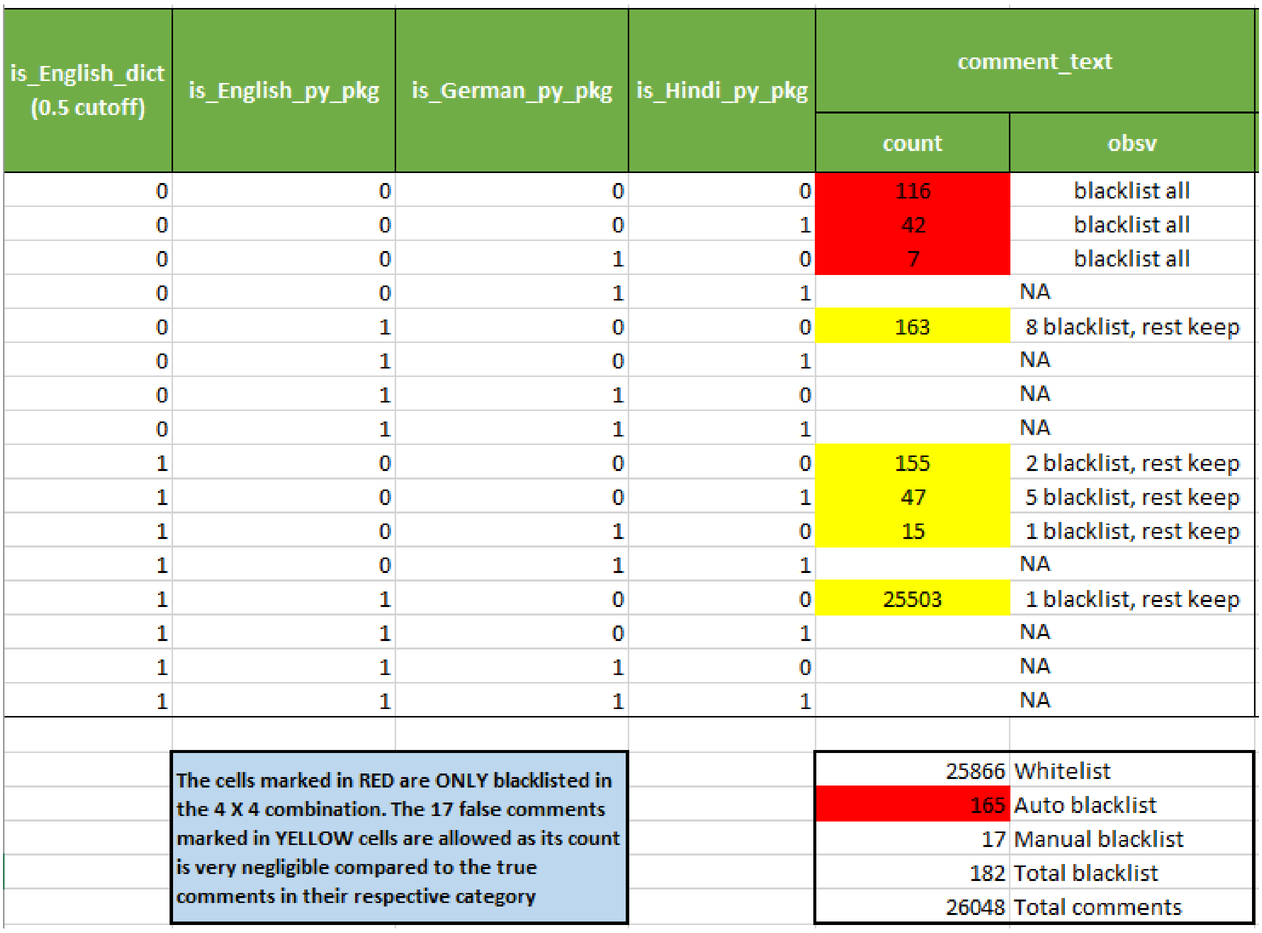
Table A.8 and Table A.9 gives illustration of the raw organic dataset after respective preprocessing stages.

Language	Type	Resource	No. articles	No. total	No. total comments	No. relevant comments
English	Biased	-	5	306037	306036	75839
	Unbiased	Forums	5	107599	106893	26050
		Newssites	7	101711	101711	26491
German	Biased	Blogs	11	5791	5791	1628
		Forums	2	374	362	306
	Unbiased	Forums	11	25856	25836	12654
		Newssites	29	195429	191745	89182

Table A.8: Statistics for the raw organic dataset (After removing unwanted punctuations)

Language	Type	Resource	No. articles	No. total	No. total comments	No. relevant comments
English	Biased	-	5	306037	306036	71987
	Unbiased	Forums	5	107599	106893	25866
		Newssites	7	101711	101711	26175
German	Biased	Blogs	11	5791	5791	1628
		Forums	2	374	362	306
	Unbiased	Forums	11	25856	25836	12654
		Newssites	29	195429	191745	89186

Table A.9: Statistics for the raw organic dataset (After removing Devanagiri scripts)



is_English_dict (0.5 cutoff)	is_English_py_pkg	is_German_py_pkg	is_Hindi_py_pkg	comment_text	
				count	obsv
0	0	0	0	116	blacklist all
0	0	0	1	42	blacklist all
0	0	1	0	7	blacklist all
0	0	1	1		NA
0	1	0	0	163	8 blacklist, rest keep
0	1	0	1		NA
0	1	1	0		NA
0	1	1	1		NA
1	0	0	0	155	2 blacklist, rest keep
1	0	0	1	47	5 blacklist, rest keep
1	0	1	0	15	1 blacklist, rest keep
1	0	1	1		NA
1	1	0	0	25503	1 blacklist, rest keep
1	1	0	1		NA
1	1	1	0		NA
1	1	1	1		NA

The cells marked in RED are ONLY blacklisted in the 4 X 4 combination. The 17 false comments marked in YELLOW cells are allowed as its count is very negligible compared to the true comments in their respective category

25866	Whitelist
165	Auto blacklist
17	Manual blacklist
182	Total blacklist
26048	Total comments

Figure A.2: Language curation algorithm

B GPT-2 Generated Customer Reviews

Table B.10 shows few examples of customer reviews generated for positive sentiment on *full organic* and *Organic products*.

Belief statements	Full organic positive	Organic products positive
Organic products are safer and better regulated	Organic products are safer and better regulated than non-organic products. Both are allowed for safe work but only one is as strong as the other. Product safety is a core competititon of many multinationals that supply organic goods and services. Organic pesticides are considered less risky from a health standpoint than conventional pesticides. Although organic food and farming are generally more sustainable and productive, there are some notable exceptions. For instance, I would be skeptical of the healthfulness of organic bananas coming from a local farmer, especially since most farmers don't bother with the extra costs associated with certification and control measures such as disease. As a result, I generally avoid organic foods and products.	Organic products are safer and better regulated than non-organic alternatives. You would have to eat more fruit and vegetables to get the vitamins one would expect to find in a high quality fruit or vegetable. But these benefits are not really heath based, they are more quality based. For instance, I would mostly prefer to buy conventional food from the US or Australia/New Zealand to buying organic food from China. Personally I prefer organic, though I'm not on a fully organic diet. For the 3rd reason is probably why Debbie thinks organic celery tastes better. Its more flavorful because it isn't as saturated with water. Same goes for watermelon, organic water melon tastes better and has a firmer texture.
Organic products are expensive	Organic products are expensive but worth it. They are healthier. They are here to stay. They are helping make a difference. There are some benefits of organic food products. This food helps to keep our mind fit and healthy. It has a great impact emotional and mental health. HEALTH BENEFITS OF ORGANIC FRUITS AND VEGETABLES: 1. It helps in weight loss. It helps in proper digestion. It reduces the risk of various diseases. It helps to keep our mind fit and healthy. It improves our eye sight. It helps to lower the cholesterol level in the body. It also helps to control the blood sugar level.	Organic products are expensive and must not be processed using industrial solvents. However, many new products - organic among them - are promising to replace conventional food with healthier, pesticide-free options. And replace it with healthier, pesticide-free options if possible. If, however, you have the wealth to shop for everything in the organic market, I see nothing wrong with that. Do that, and you'll do better for both your health and the environment. The absence of pesticide residues on fruits and vegetables is a fairly obvious benefit but there are more other far-reaching advantages of going organic.

Table B.10: Customer reviews with positive sentiment for full organic and organic products

References

[1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

[2] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[3] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[4] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[5] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[6] Generalized language models. `https://lilianweng.github.io/lil-log/2019/01/31/generalized-language-models.html`.

[7] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[8] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30, 2016.

[9] Too powerful nlp model (gpt-2). `https://towardsdatascience.com/too-powerful-nlp-model-generative-pre-training-2-4cc6afb6655`.

[10] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[11] Organic dataset original repository. `https://syncandshare.lrz.de/filestable/MlVEN0JQa2dVem9tM0U4NUYxdFZm/data/organic_dataset`, . Accessed on: 2019-04.

[12] Hannah Danner and Luisa Menapace. Consumer beliefs on organic food: An exploratory analysis of online comments in german-speaking countries and the us. In preparation.

[13] gpt-2-simple. https://github.com/minimaxir/gpt-2-simple.

[14] Project-shyam. https://gitlab.lrz.de/social-rom/project-shyam.

[15] Organic dataset overview. https://gitlab.lrz.de/social-rom/overview/wikis/Organic-Dataset, . Accessed on: 2019-05.

[16] Better language models and their implications. https://openai.com/blog/better-language-models/.