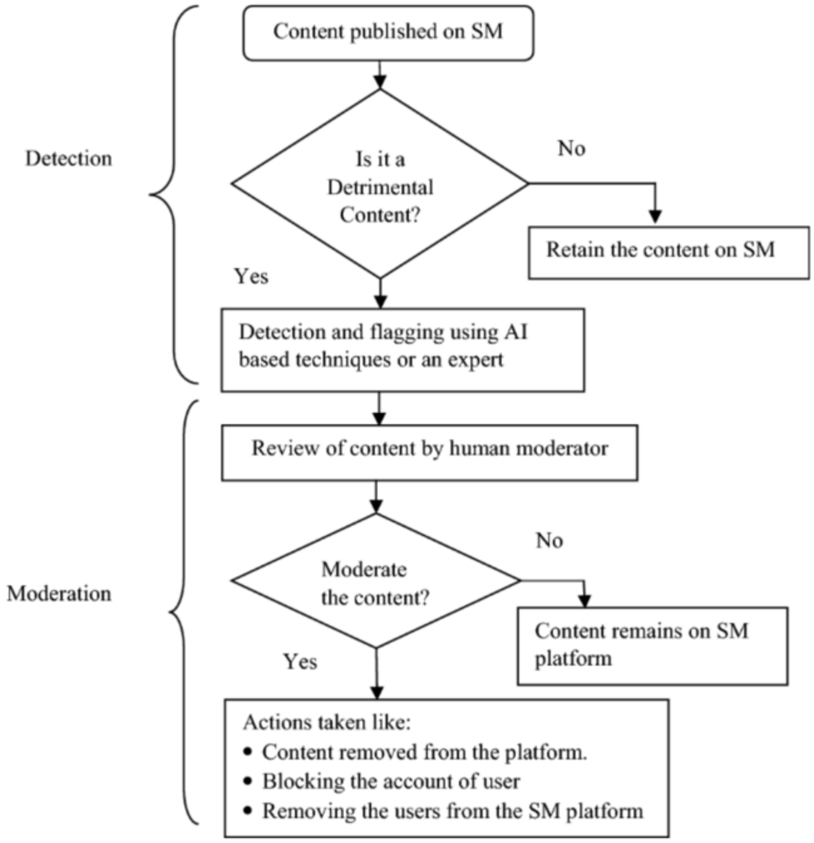
Project Report: Development of a Profanity Filter for Video Comment Streams

Introduction

In China, video platforms like Bilibili (B 站) are main ways of entertainment and information sharing. However, it isn't uncommon to encounter video comment full of inappropriate comments, such as bad words. I personally have seen some inappropriate comments on Bilibili for several times. The aim of this project is to solve this problem by developing a profanity filter for social media. Building upon what I learned in *Corpora in Speech and Language Process*, I planned to develop a profanity filter for bad words.

Related Work

This illustration outlines the processes of detecting and moderating User Generated Content (UGC) on social media (SM) platforms (Balakrishnan et al., 2020):


Detection
Content published on SM
Is it a Detrimental Content?
No
Retain the content on SM
Yes
Detection and flagging using AI based techniques or an expert
Moderation
Review of content by human moderator
Moderate the content?
No
Content remains on SM platform
Yes
Actions taken like:
• Content removed from the platform.
• Blocking the account of user
• Removing the users from the SM platform

Detection and moderation of UGC on SM platforms

To detect inappropriate contents, various methods are employed. Here are some common approaches:

1. **Keyword Filtering:** This method involves using a certain list of bad words. Comments containing these bad words are automatically flagged or removed. Keyword filtering is one of the most effective method (Xu et al., 2011). While straightforward and effective for catching common profanities, it often fails to detect disguised or creatively altered bad words.
2. **Sentiment Analysis:** This method doesn't merely focus on certain word recognition but also assist in identifying inappropriate remarks by analyzing the emotion of the text. It can help to identify negative meanings and hostile language that may not use explicit bad words but are still offensive to others. Extreme negative emotions, like anger or sadness, may indicate the presence of inappropriate content (Neethu & Rajasree, 2013).
3. **Human Moderator:** Where language is subtle and many times has its context, therefore, human judgment may be the key in this process. There is no doubt that sarcasm, cultural references, or subtleties of language are best understood by a human moderator. When comments are flagged by the automated system as suspicious, the human moderators intervene and finally decide whether the content is actually inappropriate.

This reduces the number of false positives and deals with complex situations where automated technologies might not be successful (Balakrishnan et al., 2020). However, relying solely on human moderation is also not always reliable and sometimes might be subject to bias and error as well.

4. **Machine Learning and Natural Language Processing (NLP):** Machine learning and NLP can learn from vast datasets and recognize content that indicate negative meanings. These models can be trained to understand the context and evolving language trends, thus identifying not only straightforward texts but also more subtle instances of bad words**.** For example, it can make use of text classification algorithms to train models to recognize inappropriate comments. This includes supervised learning, where models are trained using a dataset labeled as "bad" or "good". Other techniques include Support Vector Machines (SVM), Random Forest, and various types of neural networks.(Muneer & Fati, 2020; Sultan et al., 2023)

These methods play a crucial role in detecting inappropriate content on social media platforms. However, the focus of related studies is often on English content or English remarks on social media like Twitter (Balakrishnan et al., 2020; Muneer & Fati, 2020; Neethu & Rajasree, 2013). However, Chinese malicious content frequently overlooked on social media like Bilibili or Douyin. Chinese has a special kind of inappropriate contents, such as variants of profanity ("傻逼" , "操") that often evade detection by replacing characters with similar sounds (e.g., "煞笔", "傻比" for "傻逼") or using similar characters (e.g., using "日" for "操"), a phenomenon might not present in English. Therefore, there's a need for a better filtering solution that can identify and censor these Chinese bad words and their variations effectively.

Objective

The primary objective of this project is to develop an advanced profanity filter that can identify both bad words and its creatively altered variants in Chinese social media like Bilibili.

Methodology

The project needs Natural Language Processing (NLP) techniques and the Python programming language. The specific steps taken were:

1. Corpus: A comprehensive Chinese language corpus, including colloquial and formal expressions, was used to ensure the filter's effectiveness across varied expressions.

2. Identification of target bad words and variants: Using “傻逼” and “操” as a primary example, research was conducted to identify common variants and alterations used to evade traditional filters.

3. word2vec Model Training: A model was trained using the prepared corpus, focusing on recognizing similarities and distances between words to identify bad words and their variants accurately.

Corpus

*Special thanks to Michael Bayona for his assistance during my search for a Chinese corpus.

1．Introduction

The Tencent AI Lab Embedding Corpora for Chinese and English words and phrases is an advanced resource that provides continuous distributed representations of words in both Chinese and English. This corpus facilitates a wide range of NLP tasks by providing 100-dimension and 200-dimension vector representations for over 12 million Chinese words and phrases and 6.5 million English words and phrases.

2．Training process

The corpus collects a diverse array of text data from multiple sources including news articles, web pages, and novels. For the Chinese vocabulary, phrases from Wikipedia and Baidu Baike are included, while for the English corpus, phrases are also sourced from Wikipedia. And the vocabulary building process is enhanced by employing a phrase discovery approach(Shi et al., n.d.). This approach helps to identify and include emerging phrases and terms that are gaining relevance, ensuring the corpus remains up-to-date. The embeddings are trained using a specialized algorithm known as the Directional Skip-Gram model. This model differs from traditional skip-gram models by explicitly distinguishing between the left and right contexts of words within a text.

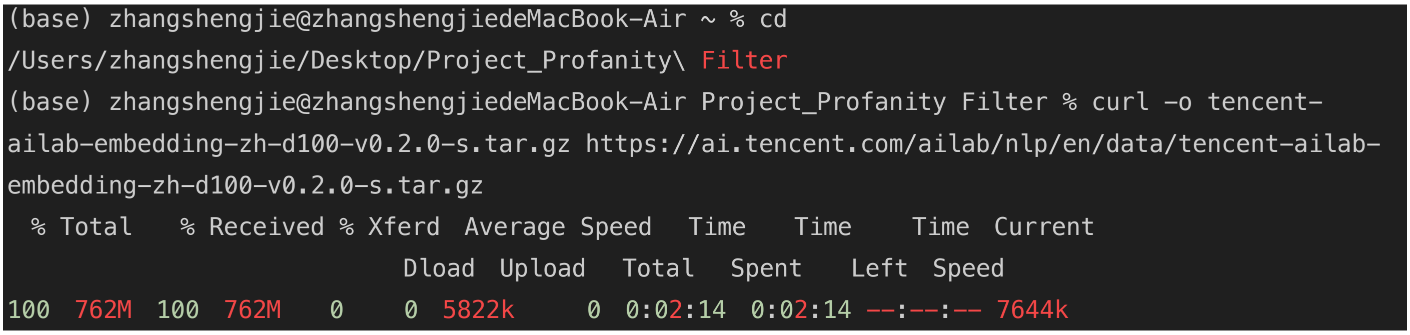
3. **Advantages**

One of the primary strengths of this corpora is its extensive coverage of domain-specific words and modern slang, and is regularly updated. This extensive inclusion of colloquial language is crucial for our study of profanity, as overly formal and traditional corpora often lack the real-world expressions and variations needed to effectively understand and filter contemporary informal speech.

Code Execution Process for Profanity Filter Project

Step 1: Corpus Installation

In the beginning, I navigate to the project directory on my MacBook Air and download the Tencent AI Lab Chinese embedding corpus using the curl command:

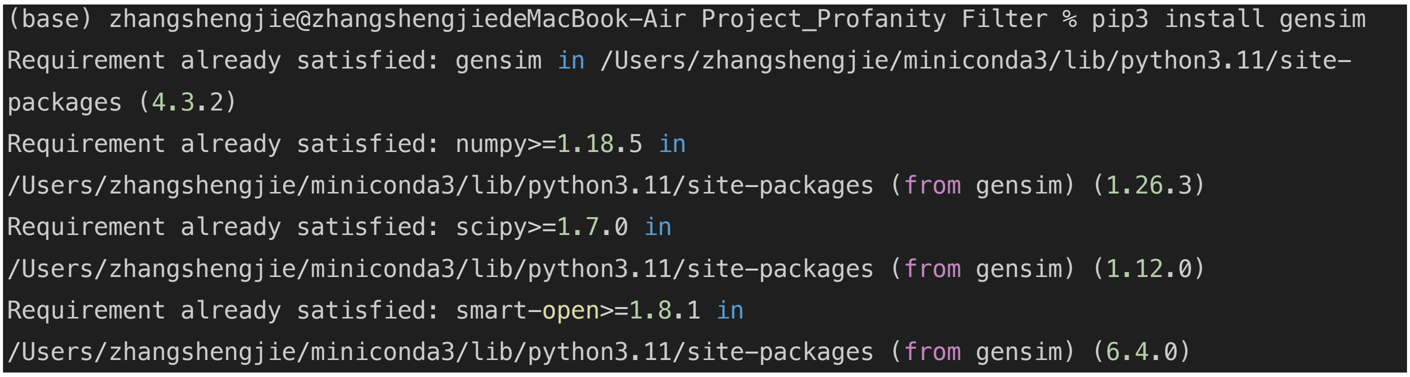


```
(base) zhangshengjie@zhangshengjiedeMacBook-Air ~ % cd
/Users/zhangshengjie/Desktop/Project_Profanity\ Filter
(base) zhangshengjie@zhangshengjiedeMacBook-Air Project_Profanity Filter % curl -o tencent-
ailab-embedding-zh-d100-v0.2.0-s.tar.gz https://ai.tencent.com/ailab/nlp/en/data/tencent-ailab-
embedding-zh-d100-v0.2.0-s.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  762M  100  762M    0     0  5822k      0  0:02:14  0:02:14 --:--:-- 7644k
```

The corpus, approximately 762MB in size, downloads successfully with an average speed of 5822kb/s.

Step 2: Installing the Gensim Package

Next, I install the Gensim package, which is necessary for handling the word embeddings:

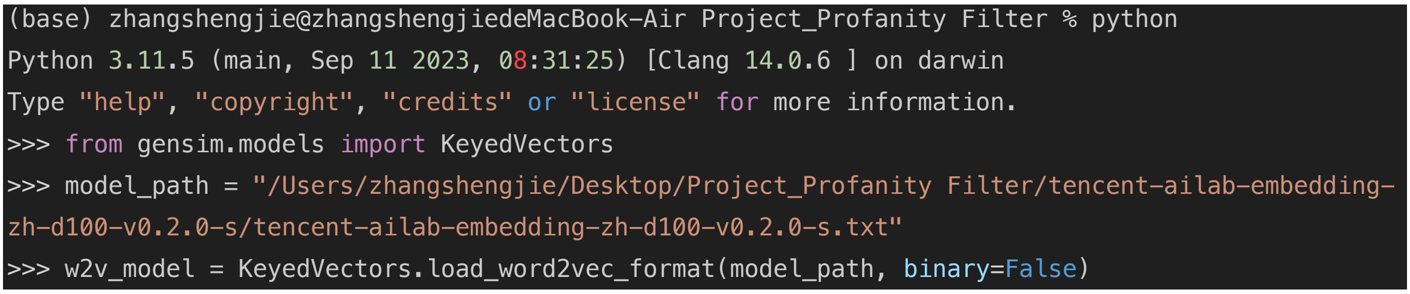


```
(base) zhangshengjie@zhangshengjiedeMacBook-Air Project_Profanity Filter % pip3 install gensim
Requirement already satisfied: gensim in /Users/zhangshengjie/miniconda3/lib/python3.11/site-
packages (4.3.2)
Requirement already satisfied: numpy>=1.18.5 in
/Users/zhangshengjie/miniconda3/lib/python3.11/site-packages (from gensim) (1.26.3)
Requirement already satisfied: scipy>=1.7.0 in
/Users/zhangshengjie/miniconda3/lib/python3.11/site-packages (from gensim) (1.12.0)
Requirement already satisfied: smart-open>=1.8.1 in
/Users/zhangshengjie/miniconda3/lib/python3.11/site-packages (from gensim) (6.4.0)
```

This step confirms that Gensim, along with its dependencies such as NumPy and SciPy, is already installed on my system.

Step 3: Loading the Model

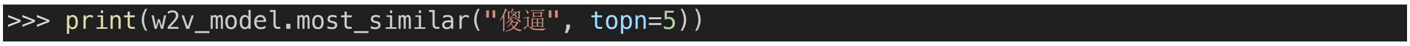
I then initiate a Python session and load the embeddings using Gensim's KeyedVectors:



```
(base) zhangshengjie@zhangshengjiedeMacBook-Air Project_Profanity Filter % python
Python 3.11.5 (main, Sep 11 2023, 08:31:25) [Clang 14.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from gensim.models import KeyedVectors
>>> model_path = "/Users/zhangshengjie/Desktop/Project_Profanity Filter/tencent-ailab-embedding-
zh-d100-v0.2.0-s/tencent-ailab-embedding-zh-d100-v0.2.0-s.txt"
>>> w2v_model = KeyedVectors.load_word2vec_format(model_path, binary=False)
```

Step 4: Identifying Profanity Variants

Using the model, I explore variations of the Chinese profanity "傻逼" and other bad words. The most_similar() method helps identify words closely related to the input term:



```
>>> print(w2v_model.most_similar("傻逼", topn=5))
```

```
[('煞笔', 0.9753351211547852), ('傻比', 0.9524455070495605), ('傻逼啊', 0.9431149959564209), ('傻
b', 0.9292600750923157), ('大傻逼', 0.9223935008049011)]
>>> print(w2v_model.most_similar("操", topn=5))
[('妈的', 0.7752609848976135), ('狗日的', 0.7749193906784058), ('他娘的', 0.7738223075866699), ('他
妈的', 0.7730444073677063), ('我操', 0.7590153217315674)]
```

When increasing the topn value reveals more variants, demonstrating the model's effectiveness at recognizing profanity and its variations.

```
>>> print(w2v_model.most_similar("傻逼", topn=6))
[('煞笔', 0.9753351211547852), ('傻比', 0.9524455070495605), ('傻逼啊', 0.9431149959564209), ('傻
b', 0.9292600750923157), ('大傻逼', 0.9223935008049011), ('二逼', 0.9119765162467957)]
>>> print(w2v_model.most_similar("傻逼", topn=10))
[('煞笔', 0.9753351211547852), ('傻比', 0.9524455070495605), ('傻逼啊', 0.9431149959564209), ('傻
b', 0.9292600750923157), ('大傻逼', 0.9223935008049011), ('二逼', 0.9119765162467957), ('傻逼吧',
0.9062010049819946), ('臭傻逼', 0.8960307240486145), ('傻叉', 0.8755214810371399), ('那个傻逼',
0.8723161816596985)]
>>> print(w2v_model.most_similar("操", topn=10))
[('妈的', 0.7752609848976135), ('狗日的', 0.7749193906784058), ('他娘的', 0.7738223075866699), ('他
妈的', 0.7730444073677063), ('我操', 0.7590153217315674), ('娘的', 0.7562870383262634), ('丫的',
0.754307746887207), ('我呸', 0.7516244649887085), ('操你妈', 0.7483251690864563), ('奶奶个熊',
0.7471296787261963)]
>>> exit()
```

Results

This approach of using the word2vec model training method provides a new direction for detecting inappropriate language in Chinese. The outcome after running the code is promising. I have successfully identified variants of the profanity “傻逼” and “操”.

Besides, in this project, I utilized a smaller corpus due to hardware limitations. Other larger corpora were available but could not be tested for the same reason. There are four different sets of embeddings available for download.

Version	Dimension	Vocab. Size	Download Url	Description
v0.2.0	200	Small (2,000,000)	tencent-ailab-embedding-zh-d200-v0.2.0-s.tar.gz	Original size: 3.6G; tar.gz size: 1.5G
		Large (12,287,936)	tencent-ailab-embedding-zh-d200-v0.2.0.tar.gz	Original size: 22GB; tar.gz size: 9.0G
	100	Small (2,000,000)	tencent-ailab-embedding-zh-d100-v0.2.0-s.tar.gz	Original size: 1.8G; tar.gz size: 763M
		Large (12,287,936)	tencent-ailab-embedding-zh-d100-v0.2.0.tar.gz	Original size: 12GB; tar.gz size: 4.7G

Discussion

The dynamic nature of language poses a substantial challenge in profanity detection, underlining the fact that no single technique is sufficient on its own. The endeavor to create a cleaner online environment for viewers requires a synergistic application of multiple technologies and methods.

References

Balakrishnan, V., Khan, S., & Arabnia, H. R. (2020). Improving cyberbullying detection using Twitter users’ psychological features and machine learning. *Computers & Security*, *90*, 101710.

https://doi.org/10.1016/j.cose.2019.101710

Muneer, A., & Fati, S. M. (2020). A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter. *Future Internet*, *12*(11), 187. https://doi.org/10.3390/fi12110187

Neethu, M. S., & Rajasree, R. (2013). Sentiment analysis in twitter using machine learning techniques. *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 1–5. https://doi.org/10.1109/ICCCNT.2013.6726818

Shi, S., Zhang, H., Yuan, X., & Wen, J.-R. (n.d.). *Corpus-based Semantic Class Mining: Distributional vs. Pattern-Based Approaches*.

Sultan, D., Omarov, B., Kozhamkulova, Z., Kazbekova, G., Alimzhanova, L., Dautbayeva, A., Zholdassov, Y., & Abdrakhmanov, R. (2023). A Review of Machine Learning Techniques in Cyberbullying Detection. *Computers, Materials & Continua*, *74*(3), 5625–5640. https://doi.org/10.32604/cmc.2023.033682

Xu, X., Mao, Z. M., & Halderman, J. A. (2011). Internet Censorship in China: Where Does the Filtering Occur? In N. Spring & G. F. Riley (Eds.), *Passive and Active Measurement* (Vol. 6579, pp. 133–142). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-19260-9_14