Memorial Content Filtering and Sentiment Analysis

Department of Veterans Affairs

Overview

The Department of Veterans Affairs' (VA) National Cemetery Administration (NCA) seeks to create an interactive digital experience that enables virtual memorialization of the millions of people interred at VA national cemeteries. This online memorial space will allow visitors to honor, cherish, share, and pay their respects and permit researchers, amateurs, students, and professionals to share information about Veterans.

Objectives

To accomplish the decorum of the site, a profanity filter and sentiment analysis would be implemented to help manage commenting.

The profanity analysis should:

- Accept a comment as string
- Output the list of profane words, and the comment with profane words filtered out

The sentiment analysis should:

- Accept a comment as string
- Return a range representing a good or bad comment

Background

Approach

The project can be developed by using a custom solution, an existing solution as a service, or a mix of both.

Custom Solution

A custom solution is developed with custom code, and optionally using existing third-party libraries.

The main advantage of a custom solution is that it can be more flexible when compared to an existing solution as a service. A custom solution can be tweaked with configurations and code in order to achieve a desired output.

For example, a sad yet respectful and honorary comment parsed with a solution as a service might be deemed a negative sentiment. However for this project such a comment might expect to be accepted as a positive sentiment. A custom solution could allow to make some adjustments.

The cost associated with a custom solution might have a higher upfront pricing for development, and an ongoing cost. The ongoing cost would be for hosting, and some maintenance.

Solution as a Service

A solution as a service is an existing hosted and managed solution by a third-party. This solution can generally easily be added to an existing flow.

The main advantage of using a solution as a service is inheriting a proven solution. Most hosted solutions have been through in-depth research, and are generally complex systems that can give accurate results.

The main disadvantage is it usually less flexible than a custom solution. Tweaking and adjusting false-positive might not be possible.

The cost associated with a solution as a service might have a lower upfront pricing, and an ongoing subscription or pay-per-use cost.

Methods

There are two methods commonly used for content filtering and sentiment analysis. First method is predefined rules and conditions, and the second is an artificial intelligence solution. Both methods apply to the custom solution, and solution as a service approaches.

Predefined Rules and Conditions

This method requires the development of rules and conditions. Some of the rules and conditions might be borrowed from existing solutions, or created from scratch.

For example, filtering content based on profanity would require to manually determine and configure what are deemed as bad words. The words might already existing in an existing libraries. In this example, if such words do not exist then an in-depth research would need to be performed.

Artificial Intelligence

An artificial intelligence allows to dynamically evaluate content, and in a way create rules. A common method of artificial intelligence is machine learning. In machine learning a system is trained with comments flagged as positive and negative. Thereafter the system can determine if a new comment is positive or negative based on its training.

It's possible that there needs to be a layer of natural language processing in order to have a precise output. Often such a system is borrowed from existing libraries or services.

Research

Defining Profanity

Not all offensive words are equal. Profanity can depend on context. The context can depend on where the comment is posted, and by the text surrounding a profane word.

In order to help determine how likely words are to be used as either profanity or clean text, ranking can be applied to a wordlist. An example of ranking:

Ranking	Used as a profanity	Used in clean text	Example
2	likely	unlikely	asshat
1	maybe	maybe	addict
0	unlikely	likely	rabbit

The same words and ranking can be used to help with content filtering and sentiment analysis.

Resources (Wordlists)

List of Offensive/Profane Word by the Luis von Ahn's Research Group
(Creator of reCAPTCHA and Duolingo) 1,384 words
http://www.cs.cmu.edu/~biglou/resources/

Map of English Profane Words to a Rating of Sureness
1,768 words
https://github.com/words/cuss

Swearjar
350+ words
https://github.com/joshbuddy/swearjar
https://github.com/raymondjavaxx/swearjar-node

AFINN 165
3,382 words/symbols
https://github.com/words/afinn-165

emoji emotion
https://github.com/words/emoji-emotion

Testing Dataset

To test different services and libraries we need to have seed comments with a similar context to the comments expected in the final project. Comments were therefore extracted from the following location:

- Afghan and Iraqi Translators Saved American Lives. Make them "Honorary Veterans."
- Help Defend Veteran Benefits
- Grant my wounded veteran husband the benefits he deserves now!
- Additional custom comments with profane or negative sentiment

Comments and sources used for testing can be found in Appendix A.

Content Filtering

General

By using a wordlist with or without ranking libraries are able to determine if a specific comment contains offensive words.

Reviewed and Top Libraries

The following libraries were tested and reviewed for compatibility.

Top Library: https://github.com/wooorm/retext-profanities
This library uses retext, a natural language processor, to analyse content. More than 1,700 rated words are provided. It does not automatically remove offensive words from original content but it provides the ability to ignore some of the cuss words. On the downside it does not seem to easily allow adding new words to the list, or modifying existing words.

https://github.com/web-mech/badwords
Does not return offensive words. Easily allows to replace wordlist.

https://github.com/KanoComputing/nodejs-profanity-util
Does not return offensive words. Easily allows to replace wordlist.

https://github.com/raymondjavaxx/swearjar-node
Does not return offensive words. Easily allows to replace wordlist.

Sentiment Analysis

General

The main challenge for sentiment analysis for this project is keeping the context in consideration. Since comments might be of a sad nature, some words used might trigger it to be analysed as a negative comment when it should really be a positive comment.

One solution is to better define what is considered a positive, and a negative comment. Example comments are defined, words deemed positive and negative can be used to train or modify wordlists.

Sentiment Versus Emotion Analysis

A subject related to **sentiment analysis** is **emotion analysis**.

For **sentiment analysis**, generally one percentage is provided as an output. A threshold can be defined to determine what is deemed a negative, and positive comment. For example, if the range is 0% to 100% (where 0% is a negative comment and 100% is a positive comment) then 50% is a common threshold. Any comments returning 50% or greater is deemed positive, and any comments below 50% is deemed negative.

In some cases the output of the sentiment analysis is multiple percentages commonly split between positive, neutral and negative.

For **emotion analysis,** multiple percentages are returned but one percentage for each type of emotion. Emotions types might be things like happy, angry, excited, sad, and other. Each service and library has different emotion/feeling types.

In the case of emotional analysis, a formula and/or rules must be applied on top of the results.

The rules could, for example, deem a comment as negative if the angry threshold is surpassed, but the sad value is low and under its threshold. In this example a comment might be abusive, instead of showing a feeling of sorrow.

Test Results for Libraries and Services

The following services and libraries were tested against the sample dataset of comments. **See Appendix B for a full spreadsheet document with many more details, and sources.**

Hit = Returned expected result for the tested comment
Miss = Did not return expected result for the tested comment

Library / Service Name	Hit Rate	Data Output Type	Note
Google Natural Language	93.10%	Percent	Misses on some of the neutral and positive comments.
Aylien	86.21%	Percent	Misses on some of the most negative comments.
Azure Text Analytics	79.31%	Percent	Misses on some of the positive comments.
Sentiment Node	89.66%	Percent	Misses on some of the negative comments but this could be adjusted using configuration. Additional words could be added to alter the score output.
Retext Sentiment	89.66%	Percent	Can be adjusted and configured by altering the list of positive and negative words and by modifying the calculation formula.
ParallelDots Sentiment	72.41%	Three percent values split between positive, neutral and negative	The context of the comment does not seem to be taken into consideration. For example, some comments are marked as negative when the comment expresses sorrow. In this project this might be seen as a positive comment.
ParallelDots Emotions	-	Five percent values split between happy, angry, excited, sad, and other	It is hard to determine negative and positive comments from the provided sentiments, and sentiment values. This solution is therefore ruled out.
ParallelDots Abusive	89.66%	Two values: boolean and confidence percentage	An appropriate value is returned for most comments however the way comments are evaluated is based on "abuse" which does does not necessarily evaluate sentiment. However in cases where there was a miss, the confidence level of the algorithm was adequately lowered. Testing on a larger dataset would be required.

Indico Sentiment	82.76%	Percent	A larger dataset would be required to test since some hits and misses could be conceived as positive depending on the context of the comment.
Indico Emotions	89.66%	Five percent values split between anger, joy, fear, sadness, and surprise	Good contender since the misses could be conceived as a hit in some context.
IBM Watson Tone Analyzer	-	Seven percent values split between anger, fear, joy, sadness, analytical, confident, and tentative	Does not provide an output for most comments therefore is not a good contender.

Top Libraries and Services

(In no particular order)

Google Natural Language
Type: Service; not customizable except if adding custom rules on top
Cost: Free up to 5,000 calls per month; $1 for each additional 1,000 calls

Indico Emotions
Type: Service; not customizable except if adding custom rules on top
Cost: Free up to 10,000 calls per month; $0.006 for each additional call

Sentiment Node
Type: Library; configurations and words can be adjusted
Cost: Hosting cost and maintenance

Retext Sentiment
Type: Library; configurations and words can be adjusted
Cost: Hosting cost and maintenance

Proposed Solution

General

The top contender for **content filtering** is the ReText Profanities library. At the time of writing it contains 1,768 ranked word, and provides lots of flexibility for analysing content.

There is four top contender for **sentiment analysis**. Two hosted services: Google Natural Language, and Indico Emotions; and two libraries: Sentiment Node, and Retext Sentiment.

By using a small dataset for testing, the hosted services provide, and libraries have similar sentiment scores.

Flow

The following flows are proposed for content filtering, and sentiment analysis. All flows can be used with the proposed services and libraries mentioned above.

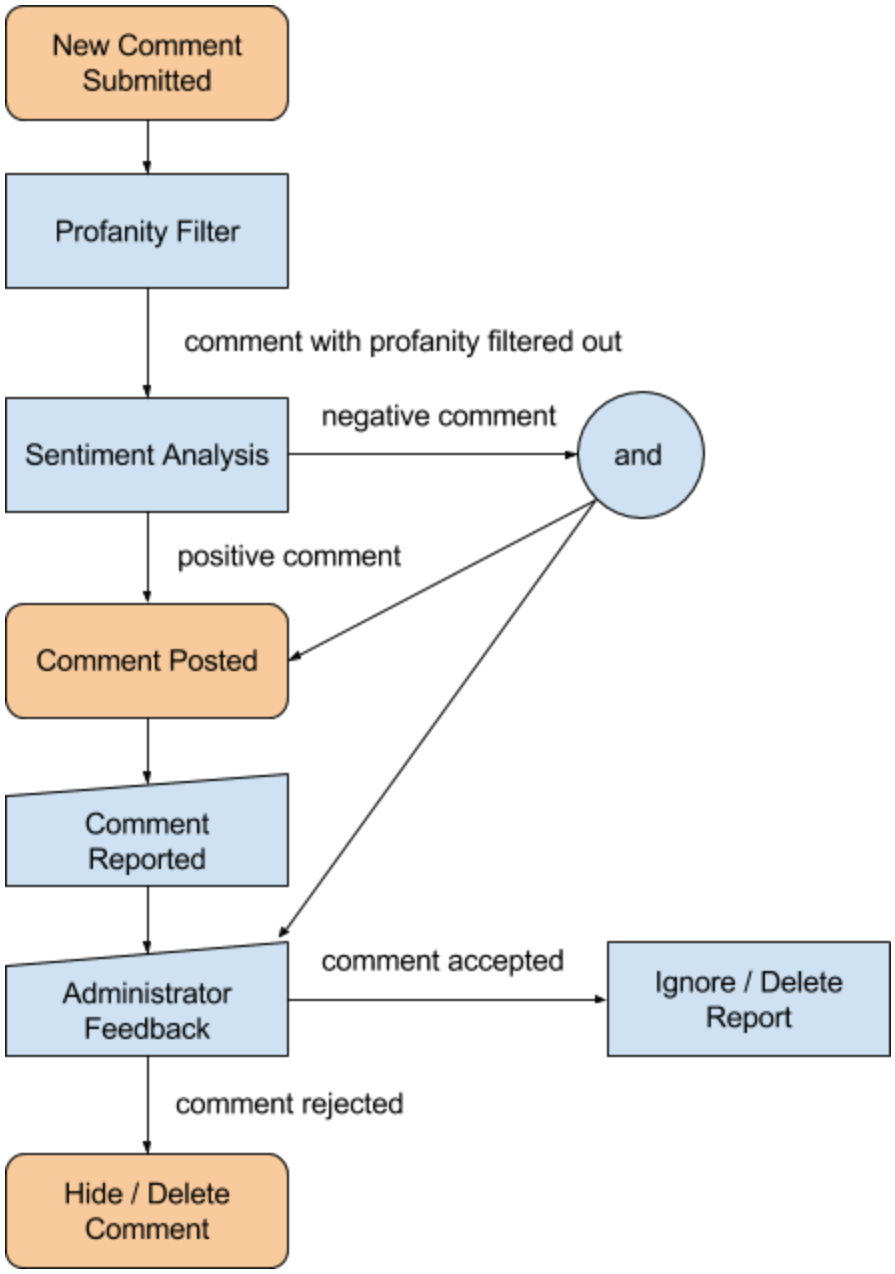
The difference between flows is:

- they allow manual feedback at different places in the flow
- they stop content from being posted at different places in the flow

Option 1

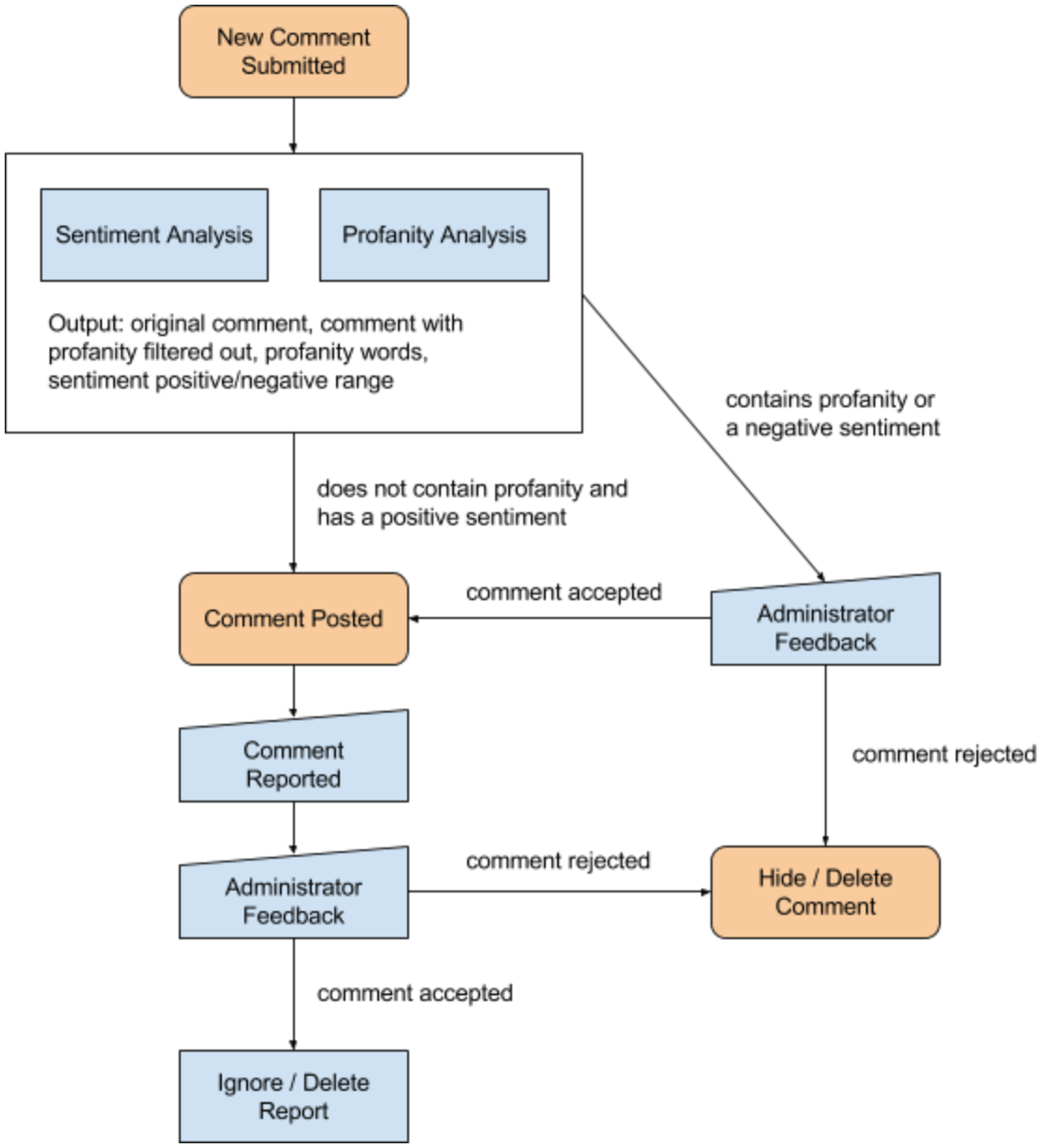
In this option comments with profanity are accepted but they're posted with profanity words filtered out. This is done as per the defined specification: "The tool should accept a comment as a string, and output [...] the comment with profane words filtered out."

Comments with negative sentiment are posted but are flagged for human review. This is done as per the defined specification: "The idea here is that we would have a several sentiment levels (good to bad) and we'd hide (**or flag for human review**) comments bellow a threshold."



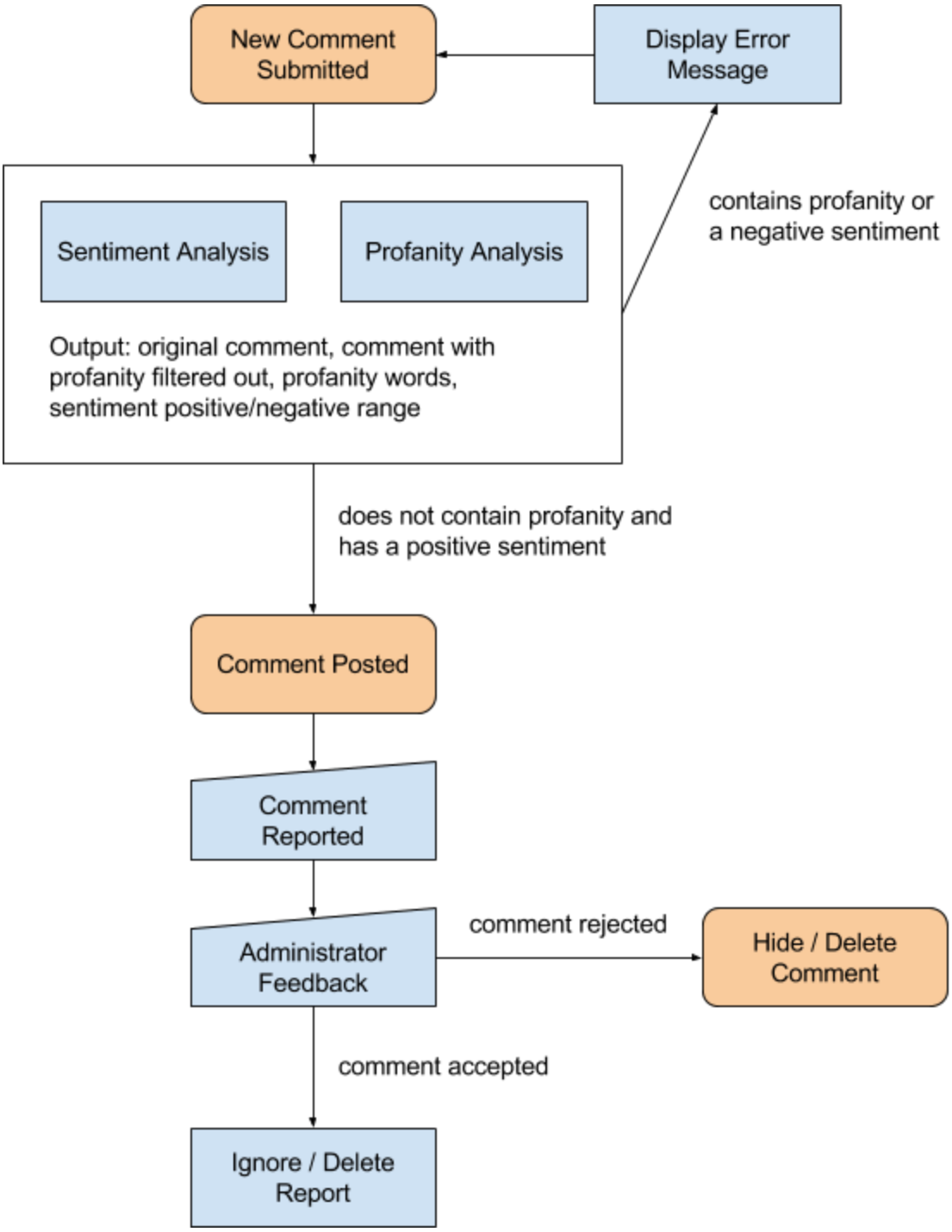
Option 2

In this option comments with profanity can be submitted but they are not posted and marked for manual administrator feedback.



Option 3

In this option the person submitting the comment must adjust the comment to pass the sentiment and profanity analysis before proceeding. The downside is the user could game the system by submitting until they understand how to submit a negative comment without triggering the detectors.



Demo / Proof of Concept

A proof of concept is included in the **demo** folder within the provided files. See the README documentation within that folder for details on how to run the proof of concept.

Additional Considerations

Spam Filtering

The content is analyzed for profanity and sentiment, but there's also the chance that spam comments make the cut. Spam comments can be created by:

- Bots
- Malicious people
- Many submission by the same person
- Other content not conforming with the decorum of the site

Some ways to reduce spam include:

- Blacklisting fields: common fields submitted by bots, which wouldn't exist in a regular transaction
- Look for missing mandatory fields: some bots remove some mandatory hidden fields such as fields manipulated by JavaScript
- Blacklisting ip addresses (dronebl.org, stopforumspam.com, ...)
- User agent detection (common user-agent/browser used by bots)
- Submissions with email addresses within the comment text: comments with email addresses might not conform with the decorum of the site.
- Submissions with web addresses within the comment text: comments with website addresses might not conform with the decorum of the site. If web addresses are allowed allowed then they could also be checked against a blacklist (surbl.org).
- Comment longer than expected length
- Comment with unexpected or inappropriate ASCII characters

Some third-party service and library libraries can assist in reducing spam:

- https://akismet.com/
- https://blogspam.net/
- http://spamassassin.apache.org/
- https://hackernoon.com/how-to-build-a-simple-spam-detecting-machine-learning-classifier-4471fe6b816e

Conclusion

The solutions found allow the profanity analysis, and sentiment analysis to:

- Accept a comment as string
- Output the list of profane words, and the comment with profane words filtered out
- Return a range representing a good or bad comment

Following the research, the following questions can now be answered:

Is there any feedback loop (from manual input) and is there a need for one at all?
A feedback loop is not required when using the proposed solutions. It can however be used, but its position depends on the chosen flow.

Having or not having a feedback loop is a balance between quality, and efficiency. By placing a feedback loop before posting the comments, then the quality can be improved but requires lots of manual accepting and rejecting of comments. By triggering the feedback loop when there's profanity, a negative comment, or user report, the quality might decrease a little but the workload of accepting and rejecting comments is greatly reduced.

Cases that your approach will fail to detect
The profanity filter can fail to understand context. Some negative words are acceptable in a context, but unacceptable in another. For a more accurate profanity filter, the sentiment analysis (or emotional analysis) can help by understanding context when the profanity filter does not detect profane words.

For the sentiment analysis, context is also hard to detect. For example, a sad comment might not necessarily be a negative comment.

How to incorporate manual feedback (ie someone manually flags the comment as inappropriate)
The method to incorporate manual feedback is defined in the flows above. Generally a comment remains posted until an administrator chooses to remove the comment.

What does the analysis do with long comments that can contain both positive and negative sentiment and how can we tweak the behavior
Some services and libraries allow to detect both negative and positive in a single comment. In the case that a comment contains negative sentiment, I think the whole comment should be flagged as negative.

I don't think that comments should be partially posted by automatically removing the negative sentences.

Now all libraries and services detect multiple sentiments in one comment. However for those who do, it's possible to get the score for each sentence, and in some cases it's possible to get additional details such as the minimum, maximum, and mean score of all positive and negative words. These details could be used to define custom rules. Some

libraries that allow this kind of splitting includes Google Natural Language, and ReText Sentiment.

Appendix

Appendix A — Comments

The comments extracted for research purposes:

Comment	0 negative, 1 neutral, 2 positive
Source: https://www.change.org/p/afghan-and-iraqi-translators-saved-american-lives-make-them-honorary-veterans/c	
We must not turn our backs on those who saved American lives and helped us in time of need. Now these people need our help; there are others who have gotten it and are nowhere near as deserving, but these really do deserve to be treated as veterans; they risked their lives to help us.	2
These men risked their life with everything to lose...we should show our gratitude.	2
They risked everything for us.	2
Those who fight for our freedom have earned my support and respect.	2
These guys deserve to be part of our country as they have already proven their loyalty to us and our troops.	2
We depend on their help (which can put them in danger) and they depend on us to treat them with fairness, honesty, and safety.	2
if we abandon people who supported us, who will ever support us?	2
They risked everything to help us! We owe them.	2
There is no reason to daly. Please take time this veterans day and pass this on. Thanks US Army and Us Air Force Veteran. G L Mann	2
These Guys are brothers!	2
Where would we be without him?	2
Of anyone who immigrated to the US, people	1

who fought for us should be the first to get respect! It's bad enough we have little respect for our own, worse if we don't respect those who left their lives to defend us.	
We owe these guys, pure and simple.	2
Protect the people who risked their lives to help Americans on a personal level . Humanity helping humanity will lead to more peace and understanding for all .	2
Clearly these people deserve our help.	1
anyone who helps our country deserves help	2
It's the right thing to do. We can't leave these brave men to be sacrificed. This will set the precedent for cooperation in other conflicts!	2
Source: https://www.change.org/p/help-defend-veteran-benefits/c	
I am an american and believe in helping our veterans. They have made sacrifices on our behalf. I love you vets! You are the greatest!!	2
Veterans need our support.	2
Disabled Veteran. Threw me to the curb bc I am a war gulf vet. I can't get anything but 30 percent. 12 surgeries later, still on 2 plus year waiting list to get a service dog	0
Veterans have sacrificed so much and have been given little to nothing in return. We need to help support veterans and their families with benefits and compensation.	2
Source: https://www.change.org/p/rep-daniel-webster-grant-my-wounded-veteran-husband-the-benefits-he-deserves-now/c	
It is sad to see someone that answered the call to defend our government, only to have the government ignore his call to in his time of need. It is appalling that someone who has sacrificed so much and is deserving of having benefits, that were promised to him, would have already received welfare for doing nothing in the general public.	0
Our veterans should be getting priority not being forgotten by our government.	2
I always want to give back to our veterans.	2
Our military deserves better.	2
Source: Custom very negative comments	

Only stupid people join the army	0
They're idiots	0
I hate those who kill our families	0
All terrorists deserve to be killed	0

Appendix B — Sentiment and Emotional Analysis Testing

A full spreadsheet document with many more details, and sources is attached within the **appendix** folder in many formats as **Sentiment and Emotional Analysis Testing**.