Management summary - NLP Project

Nicolas BERNAL, Ugo LABBÉ, Gerhard KARBEUTZ

January 2025

1 Task Overview

The task consists of detecting whether comments on online news sites are constructive or not. The data set used contains 12000 comments annotated by crowdworkers[1]. The comments are almost all in English and mainly about US and Canadian politics. We are aiming to implement multiple solutions, from less complex to more complex to find a good compromise between speed and efficacy to predict the constructiveness of a comment. You can find below an example for both types of comment.

- **Constructive**: Rob Ford was no saint. [...] It is those opinion writers and columnists who should take some time to reflect on their own lot in life and the next time they feel they have the right to mock someone in the public eye because they do not like them they need to step away and put down the pen.
- **Non-Constructive** : Margaret Wente is such an elitist snob at times. Judging from some of her columns I find her attitude that of a typical white-collar, middle to upper class, 'I went to university so I'm better than you are' kind. Some of the posters on here tonight are not much better.

2 Challenges

Some of the key challenges we encountered were: the noisy comments, using slang, abbreviations, emojis and containing mistakes. For the detection itself, not having the article where the comment was coming from make it hard knowing if the comment had something 'constructive' or not. Later, after digging into our models results, we encountered a definition problem, in fact, how to define what is constructive on certain topics, such as politics, where subjectivity and taste plays a big role.

3 External resources used

We used open-source modules in the programming language Python, such as sickit-learn, conllu or transformers, as well as the help of different websites that showed similar tasks. We also used the collaborative plateform GitHub in order to work together on the task.

4 Solution and limits

- **Naive Bayesian Classifier**

Our first try to solve the task was using a simple and explainable classifier using probabilities which allowed us to make quick test with decent results.

Results: Explainable outputs, allowing us to make changes to our data, dictionary of words, and model to get better results (about 70% accuracy). But not the best algorithm to capture the complexity of the data.

- **Feature Based Models**

[1] C3 constructive comments corpus

The next step was to use the features available from the comments to feed Machine Learning models (Logistic Regression, Random Forest, and KNN) in order to classify the text. Some of the used features included: number of tokens, average word length, number of nouns, verbs, adjectives, adverbs, etc.

Results: The number of tokes, nouns, verbs, adjectives, and determines was found to help the most to distinguish constructive and non-constructive comments (about 93% accuracy). However, this implementation does not take into account the content of the comments.

- **Deep Learning Models**

We also tried to classify the comments with Deep Learning models, one with a simple architecture which gave correct results without outperforming the Feature Based Models. We also used a LLM (named BERT) which outperformed the Feature Based Models by not much (about 93.5% accuracy), but with way higher complexity and explainability.

- **Rule based annotations**

In order to remove possible subjectivity on the original annotations in the data, and after exploring the reasoning behind our Feature Based Models, we established our own criteria to define constructive comments and reassigned the constructive/non-constructive labels accordingly. Based on the tests from Naive Bayes and Feature Based models we came out with the following approaches:

Feature based annotations: Describe constructive and non-constructive comments based on the number of tokens, verbs and adjectives.

Feature + keywords based annotations: This method combines feature-based annotations with keywords from the most common associated with each category of comments.

Results: feature-based annotations were simple and effective, filtering out many comments that seemed constructive but lacked feedback or meaningful contribution. Similarly, the feature + keyword-based annotations produced comparable results, but these didn't show significant improvement over the feature-based approach.

The biggest flaw with our initial approach is handling complaint comments and those containing quotes, as they tend to be long and usually challenging to determine if they are constructive or not.

Also the dictionary of **constructive** keywords ended up being too specific. Causing keyword-based notations to misclassify **constructive** comments.

5 Conclusion

This task had many challenges, from a poor use of English online, lack of context, or the difficulty to be sure that a comment was really constructive, we came up with an alternative solution: rules that defines constructiveness, a long comment, with many verbs and adjectives. However, previous tries to classify the comments to the correct class assigned by the crowdworkers were pretty good, with an accuarcy of more than 90%.

6 Next steps

As the comments we deployed our different solutions were mostly about politics, we could ask ourselves if there is a better solution for topic specific comments, as we could think the topic change the way to see constructiveness (politics vs science for example).