Using machine translation to improve sentiment classification of Polish restaurant reviews

Tomasz Starakiewicz   
Faculty of Economic Sciences  
University of Warsaw  
tomasz.starakiewicz@student.uw.edu.pl

Dustin Pacholleck  
Faculty of Economic Sciences  
University of Warsawd.pacholleck@student.uw.edu.pl

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. \_\_ on a representative sample of text reviews of all Polish restaurants present on Google Maps as of October 2022.

# Introduction

Online reviews are an important source of \_\_ and have been used for both \_\_

At the same time, however, humans are known to make many mistakes when typing/submitting. Studies have shown that error rates range from \_\_\_ to \_\_\_ . While misspellings carry semantic information on their own, we hypothesize that this meaning is outweighed by the \_\_ lost due failures of lemmatization and other preprocessing techniques, due typical pipelines being trained on properly formed text.

# Literature review

xxx

# Data

## Sourcing

The main data source we use in the paper is Google Maps. We accessed the Google Places API in October 2022 and performed an exhaustive search of all restaurants in Poland. The API was limited to returning a maximum of 60 restaurants, paged 20 results at a time, around a single set of coordinates submitted in the­ API call. To overcome this limitation, we first split the territory of Poland into hexagons using the H3 library created by Uber. Then we called the API at the center of each hexagon. Given that the results were sorted according to the center from the submitted coordinates, if any of the restaurants returned was outside of a circle circumscribed around the hexagon, the hexagon in question was considered exhausted. If the hexagon was not exhausted, we proceeded to recursively split it into 7 smaller hexagons and call the API for each of the resulting hexagon centers. An exemplary result of the procedure, limited to Warsaw is shown on Fig. 1.

1. Exhausted hexagons in Warsaw

The resulting universe contained 41,817 restaurants, shown on Fig. 2.

1. Locations of restaurants identified on Google Maps

For each of the restaurants, we accessed the review section at the Google Maps page and scraped it using Selenium. As a result, we were able to collect 6.1M of non-empty text reviews, which represent all reviews of all Polish restaurants present on Google Maps as of October 2022. We collected both the reviews in their original languages, as well as their English translation powered by Google Neural Machine Translation, which is based on transformer architecture.

For our study, we selected only reviews longer than 150 characters, which reduced the number of reviews to 827,341. We used Compact Language Detector (*cld3*) created by Google to identify non-Polish reviews. Cld3 is a pre-trained neural network model which uses uni-, bi- and trigrams fed into an embedding layer, then single hidden layer and finally into a softmax layer. We included only reviews classified as “Polish” with probability above 0.5. This reduced the number of reviews to 792,971.

## Sampling

Each of the text reviews is accompanied by a rating on one to five scale. As expected, the distribution of ratings was highly imbalanced, as shown on Fig. 3, with just two classes responsible for 73% of all the ratings. Interestingly, the ratings also displayed significant polarization, since these two classes corresponded to the extreme ends of the rating sclae. The reviews were downsampled, with 12,500 reviews drawn randomly from each class, ensuring balanced dataset with equal representation of each rating. This resulted in the final dataset of 62,500 reviews, with 50,000 earmarked for training dataset and 12,500 for the test dataset.

1. Distribution of ratings paired with reviews of at least 150 character legnth

## Preprocessing

Both original and translated reviews were processed using pre-trained pipelines provided along with SpaCy library. SpaCy is industrial-grade NLP library offering solutions to process text at scale, using best-in-class methods for each language task. We have chosen the library to better represent real-life use cases, since the library is widely used in practice in the industry.

Two pipelines used for preprocessing included a mix of neural network and rule-based components. The reviews were tokenized, then converted into vector representations with 300 dimensions, tagged for parts-of-speech, parsed, and finally fed into lemmatization component. For each of the languages we prepared three versions of tokenized reviews: raw (with punctuation removed), lemmatized, and lemmatized with stop-words removed.

# Modeling

To predict the sentiment, we have selected two models commonly used in sentiment analysis: Naïve Bayes (NB) classifier and eXtreme Gradient Boosting (XGBoost) classifier. NB classifier serves as a baseline, due to its popularity as a text classification tool thanks to its speed combined with reasonable effectiveness. We chose XGBoost as a comparator, as it allows for modelling complex, non-linear interactions between features, which is a natural fit for text classification task.

## Naïve Bayes classifier

The NB-classifier is a \_\_, based . It requires two important simplifying assumptions:

* Order of the words doesn’t matter (Bag of Words assumption),
* Conditional probabilities of the words occurring in a given document class are independent.

For NB classification, tokenized review text was converted into a matrix of token counts, with dictionary based on training data and tokens occurring in fewer than 50 reviews ignored. We considered two candidate models: Multinomial NB classifier and Binary Multinomial NB classifier. To train the latter, we clipped the token frequencies at document level to 1. This is based on a common assumption in sentiment analysis, that word occurrence is more important that word frequency. The standard Multinomial NB classifier outperformed the Binary NB however, so we present only the results of the former.

## XGBoost classifier

XGBoost is a popular system for training ensembles of gradient boosted decision trees, used for ranking, classification and regression tasks [X].

To create features for the model, we converted the tokenized text reviews into a term frequency-inverse document frequency matrix. We ignored tokens occurring in fewer than 200 reviews to reduce computational power requirements. We trained the model using the softmax objective, minimizing the cross-entropy loss function. We performed hyperparameter tuning using Bayesian optimization. We used stratified 3-fold cross-validation and ε = 0.3, 200 boosting rounds and early stopping after 5 rounds without improvement on multinomial log loss. We sequentially tuned depth of boosting trees, L1 regularization (alpha) and feature sampling proportion, with 20 trials for each hyperparameter. Macro F1 score was used as optimization criterion. We arrived at the maximum depth of 8, alpha of 0.2 and no random sampling of features. Each model took ca. 1 minute to train on Nvidia Tesla P100 GPU.

To determine the number of boosting rounds for the final model, we cross-validated the model with tuned hyperparameters again, with ε = 0.1 and 1000 boosting rounds and 10 early stopping rounds. We arrived at 605 boosting rounds for the final model.

# Results

Results of the classification are presented in Table I and Table II. Classifier performance varies greatly across ratings, with both performing best on the extreme ratings. Best score for each preprocessing method is underlined.

The NB-classifier performs consistently better on original Polish text across all preprocessing methods. Out of all methods, no preprocessing (except of punctuation removal) performs best, with performance degrading with each additional preprocessing step added. This performance degradation is also present for the English translation.

XGBoost classifier performs better on translated text, with the exception of lemmatized text with stop words removed, though the overall performance is not better than NB-classifier. Similar to NB-classifier, XGBoost classifier displays performance degradation progressing with each preprocessing step is present for both original and translated text versions. Notably, XGBoost doesn’t perform better than NB-classifier in general.

1. NB-classifier test results

| Rating | Test F1 scores for Naïve Bayes classifier | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Raw | | Lemmatized | | Lemmatized with stop words removed | |
| (O)riginal  (T)ranslated | **O** | **T** | O | T | O | T | |
| 1 | 0.65 | 0.64 | 0.64 | 0.63 | 0.63 | 0.61 | |
| 2 | 0.43 | 0.42 | 0.43 | 0.42 | 0.41 | 0.40 | |
| 3 | 0.43 | 0.41 | 0.41 | 0.40 | 0.41 | 0.40 | |
| 4 | 0.57 | 0.55 | 0.56 | 0.54 | 0.55 | 0.53 | |
| 5 | 0.80 | 0.78 | 0.79 | 0.78 | 0.77 | 0.76 | |
| **Macro avg.** | **0.58** | **0.56** | **0.57** | **0.55** | **0.55** | **0.54** | |

1. XGBoost-classifier test results

| Rating | Test F1 scores for XGBoost classifier | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Raw | | Lemmatized | | Lemmatized with stop words removed | |
| (O)riginal  (T)ranslated | **O** | **T** | O | T | O | T | |
| 1 | 0.68 | 0.69 | 0.68 | 0.69 | 0.67 | 0.66 | |
| 2 | 0.44 | 0.44 | 0.44 | 0.43 | 0.42 | 0.41 | |
| 3 | 0.45 | 0.44 | 0.46 | 0.44 | 0.43 | 0.39 | |
| 4 | 0.53 | 0.54 | 0.53 | 0.55 | 0.53 | 0.52 | |
| 5 | 0.77 | 0.78 | 0.77 | 0.77 | 0.75 | 0.75 | |
| **Macro avg.** | **0.57** | **0.58** | **0.57** | **0.58** | **0.56** | **0.55** | |

# Discussion

In case of NB-classifier, the differences between original/translated text are small, but consistent. Contrary to our hypothesis, machine translation doesn’t improve the classification performance. In light of these results, we hypothesize that mistyped words and misspellings carry valuable information, e.g. occurrence of misspelled words might be indicative of heightened emotional state of the review author, which could in turn be induced by bad experience, that would later result in one-star rating. The other possibility is that the linguistic noise introduced by imperfect machine translation reduces the semantic content of the reviews to the point where it starts to degrade classification performance.

Overall performance of XGBoost is surprising, as we expected the model to improve on NB-classifier due to decision trees’ ability to naturally model conditional probabilities of word occurrences. This turned out not to be the case and the significant computing power required to train these ensemble models, combined with slower inference time, do not seem to justify their use.

Nonetheless, the classifier trained on translated text performed better when no preprocessing was applied or only lemmatization, although the differences are minimal, on the verge of rounding error. We believe that these results should be viewed in the context of specific term frequency-inverse document frequency matrix used in feature engineering. As NB-classifier results have shown, the performance degrades when entropy of the training dataset is reduced (by lemmatization or stop words removal). In case of XGBoost, the TF-IDF matrix was created with higher minimum frequency threshold than token count matrix for NB-classifier, due to computing power restrictions. This significantly reduced the number of features used for training. We hypothesize that the classifier for Polish text might actually rely more on these less frequently occurring words or misstyped/misspelled versions, while anecdotally, machine translation has a tendency to reduce word variety in the text. In this context, higher threshold might disproportionately affect the classifiers built on the original text.

# Summary

Overall, we find no evidence that machine translation significantly improves performance of two popular models used for sentiment analysis, when applied to Polish reviews of restaurants. The performance either degrades (NB-classifier case) or is negligibly better (XGBoost case), while still not better than best performing model using original, non-translated text. We also find that common preprocessing techniques, lemmatization and stop words removal, degrade classifier performance in sentiment analysis of Polish reviews.

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
8. Exhausted hexagons in Warsaw