Decomposing tourists' sentiment from raw NL text to assess customer satisfaction

Maurizio Romano, Francesco Mola, Claudio Conversano

1. Introduction

Starting from Natural Language text corpora, considering data that is related to the same context, we define a process to extract the sentiment component with a numeric transformation. Considering that the Naïve Bayes model, despite is simplicity, is particularly useful in related tasks such as spam/ham identification, we have created an improved version of Naïve Bayes for a NLP task: Threshold-based Naïve Bayes Classifier (Romano et al. (2018) and Conversano et al. (2019)).

The new version of the Naïve Bayes classifier has proven to be superior to the standard version and the other most common classifiers. In the original Naïve Bayes classifier, we face two main problems:

- A response variable is needed: we need to know a priori the "Positive" ("Negative") label of a consistent amount of comments in the data;
- There is some hand-work to be done: consistently reducing the dimensionality of the problem, is a keystone for a sentiment classification task. That means to "merge words by their meanings", and usually it is done by hand. This leads to major problems in terms of subjectivity while those words are merged, moreover it prevents to consistently run an automatic program.

2. The data

For this study, we have collected two separated – but related – datasets obtained from: Booking.com and TripAdvisor.com. More in detail, with an ad hoc web scraping Python program, we have obtained from Booking.com data about:

- 619 hotels located in Sardinia
- 66,237 reviews, divided in 106,800 comments (in Italian or English): 44,509 negative + 62,291 positive
- Period: Jan 3, 2015 – May 27, 2018

Furthermore, for a comparison purpose, we have downloaded additional data from TripAdvisor.com:

- 1,450 hotels located in Sardinia
- 39,780 reviews (in Italian or English): 879 rated 1/5 stars; 1,205 rated 2/5 stars; 2,987 rated 3/5 stars; 10,169 rated 4/5 and 24,540 rated 5/5 stars
- Period: Feb 10, 2006 – May 7, 2020

3. The framework

Considering that the downloaded raw data is certainly not immediately usable for the analysis, we start with a data cleaning process. We start with some basic filtration of the words to

Maurizio Romano, University of Cagliari, Italy, romano.maurizio@unica.it, 0000-0001-8947-2220
Francesco Mola, University of Cagliari, Italy, mola@unica.it, 0000-0001-6076-1600
Claudio Conversano, University of Cagliari, Italy, conversa@unica.it, 0000-0003-2020-5129

FUP Best Practice in Scholarly Publishing (DOI 10.36253/fup_best_practice)

Maurizio Romano, Francesco Mola, Claudio Conversano, *Decomposing tourists' sentiment from raw NL text to assess customer satisfaction*, pp. 147-151, © 2021 Author(s), CC BY 4.0 International, DOI 10.36253/978-88-5518-304-8.29, in Bruno Bertaccini, Luigi Fabbris, Alessandra Petrucci, *ASA 2021 Statistics and Information Systems for Policy Evaluation. Book of short papers of the opening conference*, © 2021 Author(s), content CC BY 4.0 International, metadata CC0 1.0 Universal, published by Firenze University Press (www.fupress.com), ISSN 2704-5846 (online), ISBN 978-88-5518-304-8 (PDF), DOI 10.36253/978-88-5518-304-8

remove the meaningless ones (i.e. stopwords). Next, we convert emoticons and emoji and we reduce words to their root or base form (i.e., "fishing," "fished," "fisher" are all reduced to the stem "fish").

We use Word Embeddings to reduce the dimensionality of text data.

We recall few fundamentals concepts and terminologies, mostly related to the lexical database WordNet (Miller (1995)), to better understand the next steps:

- Words Embeddings: a vectorial way for representing words. "Each word is associated with a vector in $\mathbb{R}^d$, where the "meaning" of the word with respect to some task is captured in the different dimensions of the vector, as well as in the dimensions of other words." Goldberg (2017)
- Synsets: a collection of words that have a similar meaning. These inbuilt vectors of words are used to find out to which synset belongs a certain word.
- Hypernyms: These are more abstract terms concerning the name of particular synsets. While organizing synsets in a tree-like structure based on their similarity to each other, the hypernyms allow to categorize and group words. In fact, such a structure can be traced all the way up to a root hypernym.
- Lemmas: A lemma is a WordNet's version of an entry in a dictionary: A word in canonical form, with a single meaning. E.g., if someone wanted to look up "mouses" in the dictionary, the canonical form would be "mouse", and there would be separate lemmas for the nouns meaning "animal" and "pc component", etc.
- Words merging by their meaning: we iterate through every word of the received text and, for each word, we fetch the synset which it belongs to. Using the synset name, we fetch the hypernym related to that word. Finally, the hypernym name is used to find the most similar word, replacing the actual word in the text.

Moreover, while using the hypernyms proprieties, we adopt a newspaper pre-trained Words Embeddings produced by Google with Word2Vec SkipGram (Mikolov et al. (2013)) for obtaining the vectorial representation of all the words in the dataset (after the data cleaning process). Finally, to finalize the "merging words by their meaning" step, we use K-Means clustering.

As a result, a λ number of clusters in produced, and the centroid-word is chosen as the word that replaces all the other words present in a cluster. In this way the model is trained using, in place of a general Bag-of-Words, a Bag-of-Centroids (of the clusters produced over the Word Embeddings representation of the dataset).

The value of λ is estimated by cross validation, considering the best accuracy (or others performance metrics) within a labelled dataset (E.g. Booking.com or TripAdvisor data).

Once the data is correctly cleaned and all the words with the same meaning are merged in a single one, it is finally possible to compute the overall sentiment score for each observation.

For this purpose, the Lexical Database SentiWordNet (Esuli and Sebastiani (2006)) allows us to obtain the positive as well as the negative score of a particular word. The sentiment score ($neg\_score - pos\_score$) allows us to determine the polarity of each word. So, the overall score of a specific text (i.e. a comment, a review, a tweet) is defined as the average of all the scores of all the words included in the parsed text.

In that way, with this framework (Fig. 1) we create a temporary sentiment label while using a simple threshold over the so produced overall score. Such a temporary label is the useful base for training the Threshold-based Naïve Bayes Classifier.

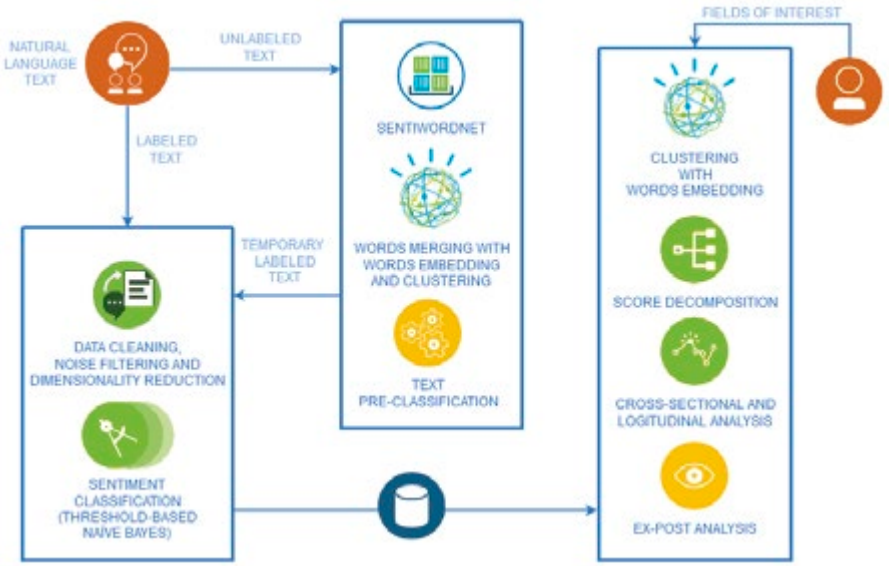


Figure 1: General Sentiment Decomposition framework

4. Threshold-based Naïve Bayes Classifier

Considering a Natural Language text corpora as a set of reviews ***r*** s.t.:

$$r_i = comment_{pos_i} \cup comment_{neg_i}$$

where $comment_{pos}$ ($comment_{neg}$) are set of words (a.k.a. comments) composed by only positive (negative) sentences, and one of them can be equal to $\emptyset$, the basic features of Threshold-based Naïve Bayes classifier applied to reviews' content are as follows. For a specific review *r* and for each word *w* ($w \in$ *Bag-of-Words*), we consider the log-odds ratio of *w*,

$$\begin{aligned} LOR(w) &= \log\left[\frac{P(c_{neg}|w)}{P(c_{pos}|w)}\right] \approx \\ &\approx \log\left[\frac{P(w|c_{neg})}{P(\bar{w}|c_{neg})} \cdot \frac{P(w|c_{pos})}{P(\bar{w}|c_{pos})} \cdot \frac{P(c_{neg})}{P(c_{pos})}\right] = \ldots = \\ &\approx pres_w + abs_w \end{aligned}$$

where $c_{pos}(c_{neg})$ are the proportions of observed positive (negative) comments whilst $pres_w$ and abs_w are the log-likelihood ratios of the events ($w \in r$) and ($w \notin r$), respectively.

While calculating those values for all the *w* ($w \in$ *Bag-of-Words*) words, it is possible to obtain an output such that reported in Table 1, where we have c_{pos}, c_{neg}, $pres_w$ and abs_w for each words in the considered *Bag-of-Words*.

	w_1	w_2	w_3	w_4	w_5	...
$P(w_i\|c_{neg})$	0.011	0.026	0.002	0.003	0.003	...
$P(w_i\|c_{pos})$	0.007	0.075	0.005	0.012	0.001	...
$pres_{w_i}$	0.411	-1.077	-1.006	-1.272	1.423	...
abs_{w_i}	-0.004	0.052	0.003	0.008	-0.002	...

Table 1: Threshold-based Naïve Bayes output

We have then used cross-validation to estimate a parameter τ such that: c is classified as "negative" if $LOR(c) > \tau$ or as "positive" if $LOR(c) \leq \tau$.

While comparing the performances on Table 2 and Table 3, we can then ensure that using the Threshold-based Naïve Bayes Classifier in this framework can definitely lead to more precise predictions.

ME	ACC	TPR	TNR	F1	MCC	BM	MK
0.092	0.908	0.936	0.398	0.951	0.268	0.334	0.215

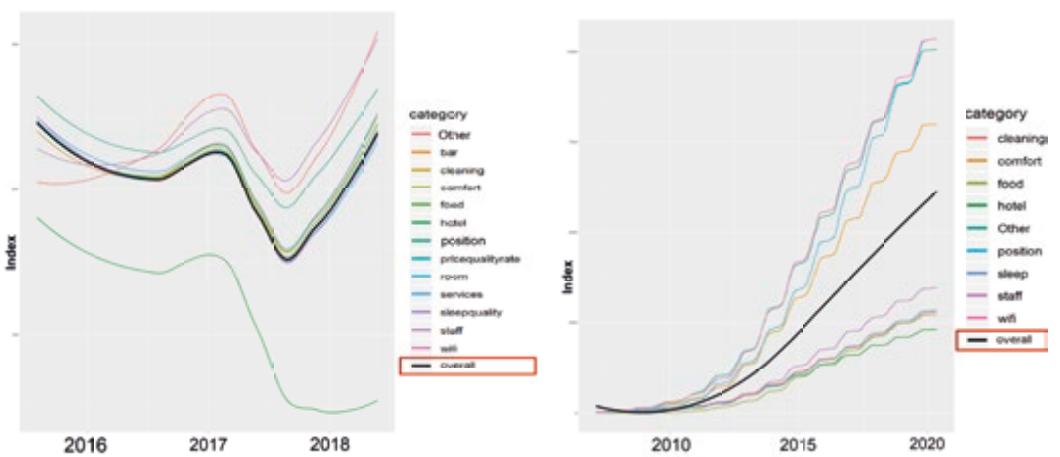
Table 2: Performance metrics obtained using the temporary sentiment label to predict the "real" label. Notice that to estimate the temporary sentiment label only text data is used, and the "real" label it is not provided in the training phase.

ME	ACC	TPR	TNR	F1	MCC	BM	MK
0.055	**0.945**	**0.973**	**0.503**	**0.973**	**0.475**	**0.476**	**0.474**

Table 3: Performance metrics obtained with Threshold-based Naïve Bayes and 10-fold CV while predicting the real label – trained with the temporary sentiment label

5. Conclusions

Compared to other kinds of approaches, the log-odds values obtained from the Threshold-based Naïve Bayes estimates are able to effectively classify new instances. Those values have also a "versatile nature", in fact they allows to produce plots like in Fig. 2a and Fig. 2b, where customer satisfaction about different dimensions of the hotel service is observed in time.



(a) Aggregated Booking.com data

(b) Aggregated TripAdvisor.com data

Figure 2: Category scores observed in time (overall sentiment in black).

References

Conversano, C., Romano, M., Mola, F. (2019). Hotel search engine architecture based on online reviews' content, in *Smart Statistics for Smart Applications. Book of Short Papers SIS2019*, eds. G. Arbia, S. Peluso, A. Pini, and G. Rivellini, Pearson, Milan, (IT), pp. 213–218.

Esuli, A., Sebastiani, F. (2006). SENTIWORDNET: A publicly available lexical resource for opinion mining, in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, eds. N. Calzolari, K. Choukri, A. Gangemi, B. Maegaard, J. Mariani, J. Odijkn, and D. Tapias, European Language Resources Association (ELRA), Genoa, (IT), pp. 417–422.

Goldberg, Y. (2017). Neural Network Methods in Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, **10**(1), pp. 1–309.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality, in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, eds. C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, Curran Associates Inc., Lake Tahoe, Nevada, (USA), pp. 3111–3119.

Miller, G.A. (1995). Wordnet: A lexical database for English. *Communications of the ACM*, **38**(11), pp. 39–41.

Romano, M., Frigau, L., Contu, G., Mola, F., Conversano, C. (2018). Customer Satisfaction from Booking, in *Selected papers Conferenza GARR_18 Data (R)evolution*, eds. M. Mieli, and C. Volpe, Associazione Consortium GARR, Cagliari, (IT), pp. 111–118.