STATISTICS IN TRANSITION new series, September 2024
Vol. 25, No. 3, pp. 175–185, https://doi.org/10.59170/stattrans-2024-034
Received – 30.11.2023; accepted – 08.07.2024

Language independent algorithm for clustering text documents with respect to their sentiment

Jerzy Korzeniewski[1], **Adam Idczak**[2]

Abstract

Determining the sentiment of a written text is an important task in text research. This task can be performed either in the supervised or unsupervised version. In this paper, we propose a novel unsupervised algorithm for documents written in any language using documents written in Polish as an example. The clustering of Polish language texts with respect to their sentiment is poorly developed in the literature on the subject. The novelty of the proposed algorithm involves the abandonment of stoplists and lemmatisation. Instead, we propose translating all documents into English and performing a two-stage document grouping. In the first step of the algorithm, selected documents are assigned to a class of positive or negative documents based on a set of lexical and grammatical rules as well as a set of key-terms. Key-terms do not have to be entered by the user, the algorithm finds them. In the second step, the remaining documents are attached to one of the classes according to the rules based on the vocabulary found in the documents grouped in the first step. The algorithm was tested on three corpora of documents and achieved very good results.

Key words: text mining, document sentiment, document clustering.

1. Introduction

Text clustering is becoming more and more important every day with the rapid development of media massive output of news, posts, comments, articles, etc. Media and official government departments can effectively handle news and grasp the development trend of news popularity. Therefore, clustering of texts has become an important research topic in text clustering. Text clustering with respect to text sentiment is a very vital part of this research topic. By text sentiment we understand either positive or negative opinion expressed by the author about the subject of the document under study. Other variants of the meaning of the term 'sentiment' are possible like, e.g. 'sentence sentiment', but we refer it to the sentiment of the whole

[1] Department of Demography, Faculty of Economics and Sociology, University of Lodz, Lodz, Poland. E-mail: jerzy.korzeniewski@uni.lodz.pl. ORCID: https://orcid.org/0000-0001-6526-5921.

[2] Department of Statistical Methods, Faculty of Economics and Sociology, University of Lodz, Lodz, Poland. E-mail: adam.idczak@uni.lodz.pl. ORCID: https://orcid.org/0000-0001-9676-2410.

document. This approach is probably most difficult as it creates problems when there are several objects being described in the document. We propose a novel algorithm tested on Polish language texts, but the algorithm is independent of the language because its intrinsic feature is text initial translation into English. Another important feature of our proposal is the abandonment of any stoplists and lemmatisations. The reasons are quite obvious, in the case of the Polish language none of implementation of lemmatisation is free of errors, for example a word that has multiple meanings is converted into the base form of a word with a different meaning. Moreover, some words are not recognized and are not lemmatized at all. Even if there was any such procedure it would often kill the meaning of the text through harsh simplification of grammatical and lexical structures. Eder and Górski (2023) showed that there is no substantial difference between using lemmatised words and its original forms in classification problem.

The paper is organized as follows. In Part 2 we give a short summary of related work in this research topic. In Part 3 we describe the proposal of our algorithm. Part 4 contains empirical evaluation of the algorithm in the example of three Polish language text corpora and conclusions are given in Part 5.

2. Related work review

Although the task of unsupervised text clustering is closely related to the supervised version it is significantly less researched. It can be even considered to be a relatively new topic as its beginnings go back to hardly 2002 (Pang et al., Turney). For example, Turney used a specific unsupervised learning technique based on the mutual information between document phrases and the words "excellent" and "poor", where the mutual information is computed using numbers collected by a search engine. Li and Liu (2012) proposed an algorithm for document clustering whose idea was to cluster repeatedly the documents via the *k*-means with random choice of starting points with subsequent majority voting mechanism. The number of running times needed to be large enough to lessen the effect of outliers and instability. In the second stage the polarity of clusters was determined on the basis of external sources (WordNet). Souza (Souza et al., 2017) proposed a Particle Swarm Optimization (PSO) algorithm to cluster documents with respect to their sentiment. The cosine distance and silhouette index were used in assessing the clustering quality. The results were not very impressive. For example, the accuracy of clustering ranged from 40% to 60% and was sometimes losing to *k*-means. The algorithm mimicked the behavior of insects in their search for food and attracted much attention, particularly in biological sciences. Probably, the first to propose a probability-based model approach to sentiment analysis were Lin and He (Lin et al., 2009). Their model was based on the Latent Dirichlet Allocation (LDA) and

achieved accuracy equal to about 70%. Similar to this approach is the one based on SOM (Self Organizing Map) type of neural networks, e.g. Chifu et al. (2015). In their research the algorithm achieved accuracy of about 60%.

As far as researching the Polish language text is concerned, the only methodological work known to the authors is Kocon et al. (2019), however, the methods developed in this work are dependent on external sources.

3. Description of the proposed algorithm

The main idea behind the construction of the algorithm is the lack of use any extensive usage of stoplists and lemmatizations. Instead, the algorithm uses the translation of documents written in Polish into English. Translated documents are grouped based on a two-stage algorithm. In the first stage, documents are grouped based on rules using lists of positive and negative terms and key-terms found. In the second stage, documents are grouped based on positive and negative bigrams. These positive and negative bigrams are found close to key-terms in positive and negative documents grouped in the first stage respectively. Below are the steps that make up the algorithm:

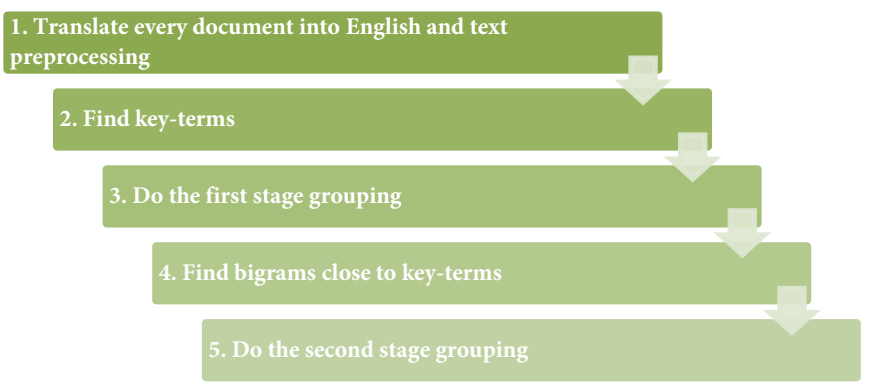

Figure 1: Algorithmic description of the experiment

Source: own work.

Translation of documents

The document corpora are translated fully automatically using the R language and the translate 2 function from the deeplr package[3]. This function uses the online DeepL translator[4].

[3] R documentation is available at https://cran.r-project.org/web/packages/deeplr/deeplr.pdf

[4] https://www.deepl.com/translator

Text preprocessing

Text preprocessing is an integral part of every text mining application. We reduce this step only to (1) remove all unwanted punctuation marks except sentence-ending marks, i.e. "?", "!", ".", (2) use our own shortened stoplist containing only stopwords such as: "a", "an", "the", "and", "or". These words are very common in the English language and rather do not convey any meaning.

In this step, lemmatization is usually carried out, which is to remove inflections and map a word to its root form. To perform lemmatization, dictionaries (e.g. *wordnet*) are needed to enable this type of conversion of an inflectional word to its basic form. We propose to skip of direct lemmatization Polish language at this stage by swapping highly inflected Polish for less inflected English. This action has a twofold effect on the text. First, implicitly, a 'soft' lammatization is performed. Hence, for example, Polish words *samochód, samochodu, samochodowi, samochodem, samochodzie* can be replaced by one English word *car*. Secondly, morphologically complex languages with relatively free word-order which is Polish are converted into more structured English, which is more suitable for creating grouping rules.

Finding key-terms

We define a key-term as any term which is directly preceded or followed by a word from both positive and negative lists at least once as shown in the diagram below:

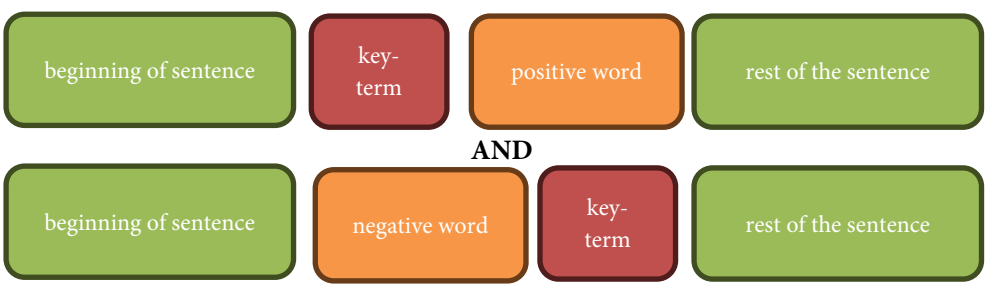

Figure 2: Key-term searching idea

Source: own work.

Example for searching *hotel* key-term:
Great hotel for business trips!
I am disappointed because of my stay in this ugly hotel.

Lists of positive and negative words were created in an approach based on data analysis with expert verification. The list of positive (negative) words was formed by terms next to (up to 3 words) to the most common nouns in the set of positive

(negative) documents. Expert verification consisted of removing words with no sentiment, in particular, these were nouns (acting as subjects) or verbs (most often the word *to be* in various forms). The positive (negative) list contains words that are commonly associated with a positive (negative) connotation. These are mainly adjectives, adverbs and nouns. The positive list is presented below:

adequate, amazing, awesome, beautiful, beautifully, best, better, classic, classics, clean, cleanliness, comfortable, competent, convincing, convenient, cool, cozy, decent, delicious, delighted, durable, ecstatic, effectively, efficient, efficiently, elegant, enjoying, excellent, exceptional, extra, fantastic, favourite, favourites, fine, firecracker, flawless, fresh, friendly, fun, good, great, happy, high-end, ideal, interesting, like, likes, long-lasting, love, lovely, magnificent, mega, neat, nice, nicely, ok, okay, outstanding, peaceful, perfect, pleasant, positive, positively, pretty, professional, professionally, reliable, revelation, right, satisfied, sensational, successful, successfully, super, superb, tasteful, tasty, timeless, top, true, truly, unbeatable, valuable, value, well, wonderful, worth, worthy.

The negative list is presented below:

bad, blatantly, break, broken, counterfeit, damage, damaged, defeat, defective, destroy, destroyed, dirty, disappointed, disappointing, disappointment, disillusion, disrupt, disturbance, disturbed, downside, dull, embarrassing, failure, faint, fake, fatal, hate, horrible, inadequate, lack, lacks, letdown, lousy, miserable, missing, monotone, mundane, plain, poor, poorest, poorly, problem, problems, scandal, scandalous, scandalously, shit, shitty, sorry, spoil, spoiled, tacky, terrible, terribly, ugly, unclean, unfortunately, uninteresting, unprofessional, unrealistic, unreliable, unsatisfied, unsuccessful, unsuccessfully, until, untrue, unvaluable, unworthy, valueless, waste, weak, worse, worst.

The first stage grouping

In the first stage grouping documents are grouped into two clusters (negatives or positives) according to the following 3 rules.

Rule 1:

- **positive label** is assigned to the document in which there is RECOMMEND (or its variation) followed by a full stop.
- **negative label** is assigned to the document in which there is RECOMMEND (or its variation) followed by a full stop and preceded with negation.

Rule 2:

- **positive label** is assigned to the short document (n<5) in which there is at least one term from the list of positive terms or at least one term from the list of

negative terms preceded with negation (and there are no negative terms in a document),

- **negative label** is assigned to the short document (n<5) in which there is at least one term from the list of negative terms or at least one term from the list of positive terms preceded with negation (and there are no positive terms in a document).

Rule 3:

- **positive label** is assigned to the document in which there is a key-term directly followed or preceded by a term from the list of positive terms and there are no terms from the list of negative terms,
- **negative label** is assigned to the document in which there is a key-term directly followed or preceded by a term from the list of negative terms and there are no terms from the list of positive terms.

Finding bigrams

Bigrams are created based on up to the nearest 4 words preceding or following the key-terms separately for positive and negative documents grouped in the first stage. The procedure searches among these 4 words for the first 2 "to the left" and the first 2 "to the right" of the key-term, skipping keywords and taking into account punctuation marks ending the sentence. We will further call these two words positive or negative bigrams (depending on which cluster the document belongs to). These bigrams are an essential part of the second stage grouping. The main idea for searching such bigrams is presented in Figure 3.

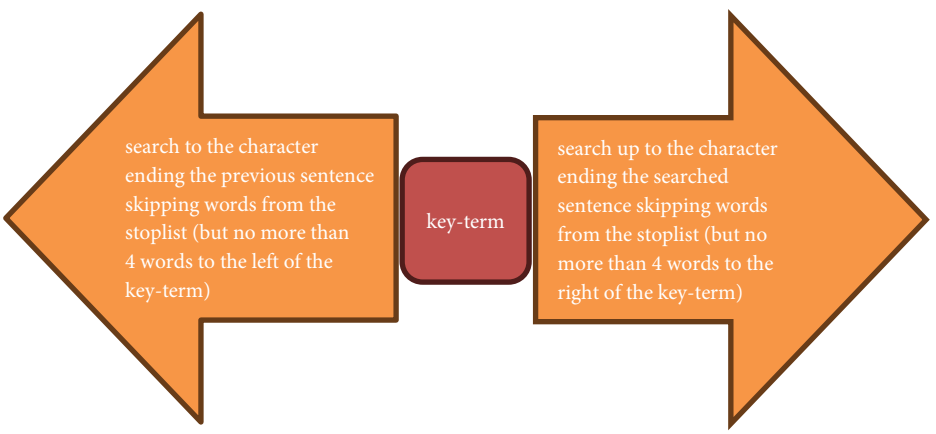

Figure 3: Bigram searching idea

Source: own work.

Real example (black words from the stoplist are skipped) – positive document with *reception* key-term:
For praise deserves the reception open 24 hours a day.
Bigrams created: *praise deserves*, *open 24*.

The second stage grouping

After the first grouping step, there are still documents that have not been assigned to one of the clusters. In the second stage, the positive and negative bigrams described above will be used to group these documents, i.e. **positive (negative) label** is assigned to the document in which there are more positive (negative) bigrams.

4. Empirical evaluation of algorithm

Data sets

Proposed method is evaluated on three datasets. These datasets were obtained from a company that collects online reviews. The first dataset called *Hotels* consists of reviews of hotels in Poland from 2020–2021. It consists of 4 385 terms. There were 221 negative and 1 667 positive documents as a total of 1 888 documents in the first dataset. The second corpus named *Perfumes* consists of reviews of perfumes from online shop from 2021. It consists of 2 675 terms. There were 271 negative and 2 333 positive documents as a total of 2 604 documents in the first dataset. The third collection of documents (*Courier*) contains reviews of courier companies from Poland written in 2021. This dataset includes 4 579 terms among 4 191 documents (541 negative, 3 650 positive).
Each document in these three datasets is labelled with positive or negative sentiment (positive or negative class). These labels were assigned manually by an opinion holder (by rating 1–5 stars where 1–2 stars documents are labelled as negative sentiment and 4–5 are marked as positive, 3 stars are excluded because of ambiguous connotation).

Results

To determine the optimal number of key-terms, we ran the algorithm many times setting a different number of most frequent terms in the range of 10–100, measuring the quality of classification and the percentage of unclassified documents[5]. Based on the results, we recommend and adopt in the study the use of 20 key-terms, which in the 3 datasets considered provided the smallest percentage of unclassified documents with high value of accuracy and F1 measures.
Hotels key-terms:

apartment, atmosphere, bathroom, beds, breakfast, conditions, decor, everything, food, helpful, hotel, large, location, place, price, restaurant, room, service, staff, tidy.

[5] Excessively many key-terms negatively affect clustering rules since the document (often short review) would mostly consist of keywords. Even some documents would remain unclassified because they would only consist of key-terms.

Perfumes key-terms:

as, delicate, fragrance, intense, it, lasting, long, masculine, more, my, original, packaging, perfume, price, product, quality, quiet, scent, sensual, smells.

Courier key-terms:

all, always, company, condition, contact, courier, delivery, everything, fast, it, order, package, pay, possible, price, quality, quick, service, shipment, time.

For each data set, positive and negative bigrams were determined according to the idea presented in Figure 3. The mechanical method of determining bigrams is not without drawbacks, as both lists repeat terms with no sentiment such as "have been", "has been", "can not", etc. It seems reasonable to correct such inaccuracies manually, but we wanted an algorithm that does not require human intervention. Note that this list is not fixed and will be specific to each dataset. Nevertheless, our results show that this effect does not adversely affect the quality of text clustering.

We evaluated the proposed algorithm on the abovementioned three datasets with 20 key-terms set-up. The results prove that the proposed algorithm yields high quality classification (see Table 1). The most commonly used measure of classification quality is accuracy, i.e. the percentage of correctly classified documents. The quality of classification was assessed by means of accuracy:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \tag{1}$$

where:

TP (true positives) is the number of documents with positive sentiment classified as positive,

TN (true negatives) is the number of documents with negative sentiment classified as negative,

FP (false positives) is the number of documents with negative sentiment classified as positive,

FN (false negatives) is the number of documents with positive sentiment classified as negative.

Because of the unbalanced clusters within the datasets used, the F1 measure was used, which is the harmonic mean of precision and recall:

$$F1 = \frac{2*precision*recall}{precision+recall}, \tag{2}$$

where:

$$precision = \frac{TP}{TP+FP}, \tag{3}$$

$$recall = \frac{TP}{TP+FN}, \tag{4}$$

Table 1: Detailed clustering results on three datasets

Corpus	Number of documents	Number of terms	Cluster frequencies	Percent of grouped documents	Accuracy	F1	Computing time (in seconds)
Hotels	1 888	4 385	1 667 – positive 221 - negative	95.50	94.40	96.93	15.43
Perfumes	2 604	2 675	2 333 – positive 271 - negative	92.71	94.26	96.89	16.35
Courier	4 191	4 579	3 650 – positive 541 - negative	93.35	92.84	95.86	29.95

Source: own calculation based on three datasets.

According to Table 1 the proposed two-stage algorithm achieves very high accuracy and F1 for all investigated datasets (both measures are above 90% for all cases). The highest accuracy is obtained on *Hotels* (94.40%), the lowest accuracy is obtained on *Courier* (92.84%). Similarly, in the case of F1 measure the highest score is achieved on *Hotels* (96.93%) and the lowest F1 is reported on *Courier* (95.96%).

Table 1 also presents the percent of grouped documents, i.e. documents that are grouped into one of two clusters (positive or negative). The highest percentage is obtained on *Hotels* dataset (95.50%) and the lowest is observed on *Perfumes* (92.71%). It means that the proposed algorithm does still leave from 4.5% up to 7.29% of ungrouped documents depending on the dataset.

It is worth mentioning that all considered datasets are grouped in less than a minute. All calculations were performed in R using the tm[6], word2vec[7] and deeplr[8] packages.

5. Conclusions

In the article, a novel unsupervised algorithm for determining the sentiment of text documents written in any language was proposed. The proposal was tested using the example of three corpora of documents in the Polish language. The very important task

[6] R documentation is available at https://cran.r-project.org/web/packages/tm/tm.pdf

[7] R documentation is available at https://cran.r-project.org/web/packages/word2vec/word2vec.pdf

[8] R documentation is available at https://cran.r-project.org/web/packages/deeplr/deeplr.pdf

of establishing the sentiment of Polish texts is very poorly developed in the literature on the subject, therefore the algorithm fills this gap. The novelty of the proposed algorithm includes the abandonment of any extensive usage of stoplists and lemmatizations. Instead we first translated all documents into English and then carried out a two-stage documents grouping. In the first step the algorithm establishes key-terms characteristic for the subject and using these terms, as well as a set of lexical and grammatical rules, assigns some documents to a class of positive or negative documents. In the second step, the remaining documents are attached to one of the classes by means of the rules based on the vocabulary, especially bigrams, found in the documents grouped in the first step. The algorithm was tested on three corpora of documents and achieved very good results comparable with most popular supervised neural networks for English texts (see Chifu et al., 2015 or Sharma et al., 2013). The limitations of the algorithms include unique words appearing in one-time occasions. In such cases it is impossible to overcome this limitation without the use of external sources. Other kind of limitation comes from inadequate translations, usually concerning modern slang expressions not covered by online translators. In spite of all drawbacks we strongly believe that the algorithm proposed deserves attention and further research. For the first attempt we would suggest extending the list of special grammatical structures clearly defining the sentiment independently of the subject of the text.

References

Chifu, E., Chifu, V., Letia, T., (2015). *Unsupervised Aspect Level Sentiment Analysis Using Self-Organizing Maps*, IEEE, https://doi.org/10.1109/SYNASC.2015.75.

Eder, M., Górski, R. L. (2023). Stylistic Fingerprints, POS-tags, and Inflected Languages: A Case Study in Polish. *Journal of Quantitative Linguistics*, 30(1), pp. 86–103.

Kocon, J., Milkowski, P., Zasko-Zielinska, M., (2019). *Multi-Level Sentiment Analysis of PolEmo 2.0: Extended Corpus of Multi-Domain Consumer Reviews*, Proceedings of the 23rd Conference on Computational Natural Language Learning, pp. 980–991.

Manaa, M. E., Abdulameer, G., (2018). Web Documents Similarity using k-Shingle tokens and MinHash technique. *J. Eng. Appl. Sci.*, 13, pp. 1499–1505.

Lin, C., He, Y., (2009). *Joint sentiment/topic model for sentiment analysis*, 18th ACM conference on Information and knowledge management, pp. 375–384.

Li, G., Liu, F., (2012). Application of a clustering method on sentiment analysis. *Journal of Information Science*, 38(2) pp. 127–139.

Sharma, A., Dey, S., (2013). *Using self-organizing maps for sentiment analysis*, Cornell University Library.

Souza, E., Santos, D., Oliveira, G. et al., (2020). Swarm optimization clustering methods for opinion mining. *Nat Comput*, 19, pp. 547–575, https://doi.org/10.1007/s11047-018-9681-2.

Yuqiang Tong, Lize Gu, (2018). *A News Text Clustering Method Based on Similarity of Text Labels,* Advanced Hybrid Information Processing – Second EAI International Conference, ADHIP 2018.

Zhang, W. M., Jiang, W. U., Yuan, X. J., (2010). K-means text clustering algorithm based on density and nearest neighbor, *J. Comput. Appl.*, 30(7), pp. 1933–1935.