International Journal of Approximate Reasoning 155 (2023) 104–112

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Lexicon-based sentiment analysis in texts using Formal Concept Analysis

Manuel Ojeda-Hernández *, Domingo López-Rodríguez, Ángel Mora

Universidad de Málaga, Andalucía Tech, Málaga, 29071, Málaga, Spain

ARTICLE INFO

Article history:
Received 20 December 2022
Received in revised form 1 February 2023
Accepted 1 February 2023
Available online 3 February 2023

Dataset link: https://github.com/Malaga-FCA-group/demo-sentiment-analysis

Keywords:
Formal Concept Analysis
Sentiment analysis
Polarity analysis
Text mining
Lexicon

ABSTRACT

In this paper, we present a novel approach for sentiment analysis that uses Formal Concept Analysis (FCA) to create dictionaries for classification. Unlike other methods that rely on pre-defined lexicons, our approach allows for the creation of customised dictionaries that are tailored to the specific data and tasks. By using a dataset of tweets categorised into positive and negative polarity, we show that our approach achieves a better performance than other standard dictionaries.

1. Introduction

At present, knowledge extraction from texts is a hot research topic. Corporate reports, legal documents, online publications, emails, web, Social Networks, mobile phone chat apps, and consumer reviews in online companies, such as those in the fields of tourism, medicine, travel, and e-commerce, contain a wealth of unstructured knowledge. The term text mining, also known as natural language processing (NLP) or computational linguistics, involves not only the extraction of a bag of words and their frequencies (probably the simplest approach), but also methods to enrich chunks of text, including words, terms, and sequences, with metadata retrieved from textual sources. Textual data mining allows for the addressing of real-world problems in social networks, the identification of leaders, the detection of topics and fake news, and the analysis of the sentiment of texts, among others.

To properly establish our framework, we may refer to [10] to demonstrate the evolution of research in the field, distinguishing between early approaches to NLP that focused on the linguistic structure of language and served as the foundation for advancements in machine translation, speech recognition, and speech synthesis, and more recent, complex developments.

In [14], the authors summarise the current state of the art of the field and its key challenges, including the “absence of structure in the texts, the heterogeneous and distributed nature of the documents, the context and domain dependence or the diversity and ambiguity of languages, multilingualism.” They suggest that mathematical tools such as fuzzy set theory, fuzzy association rules, and ontologies could be valuable tools for extracting more useful knowledge.

* Corresponding author.
E-mail address: manuojeda@uma.es (M. Ojeda-Hernández).

https://doi.org/10.1016/j.ijar.2023.02.001

As these authors have noted, the goal is to "obtain pieces of new and useful knowledge with a semantics as rich as that of text, instead of patterns in data as clusters." In other words, the aim is not simply to collect knowledge in a dataset of terms, but to find knowledge bases to reason with. They also state that "fuzzy and probabilistic logics are appropriate for these purposes," albeit with a "trade-off between expressivity of the logic and complexity of the reasoning." As we will demonstrate later, our work aligns perfectly with this perspective, focusing on the representation of knowledge extracted from text mining by means of a lattice of co-occurring terms.

Applications of automated methods in areas such as social network analysis, searching for consumer and political opinions, detecting topics, classifying individuals, and detecting leaders or fake news, hold significant potential for the creation of new market niches. In [25], a survey of the primary methods of text-based sentiment analysis was published.

In these applications, the key factor is the determination of valuations (positive, negative, neutral or using grades) for the opinions expressed by users of the social network. The complexity of the methods used ranges from simple word frequency counting and comparison to pre-established dictionaries (bags of words) to the use of machine learning techniques such as naive Bayesian or support vector machines [23].

Carpineto and Romano [4] explored the use of Formal Concept Analysis (FCA) in text data mining, while Castellanos et al. [5] applied FCA to topic modelling, using it to compute concepts and determine topics from a dataset of documents and terms. Poelmans et al. [20] developed a text mining tool assisted by humans using FCA, which they applied to textual police databases and hospital logs with the aim of visualising the concept lattice, navigating through the concepts (pairs of texts and terms), and identifying suspects or problems with patients.

Ravi et al. [21] combined FCA and sentiment analysis, applying the method to the financial sector. The approach involves computing concepts using FCA and employing learning methods to classify the sentiments of the concepts using dictionaries expanded with WordNet. Additionally, association rules are computed using an FCA tool and a sentiment is assigned to each rule, allowing for the determination of whether consumers are unhappy or not.

Our approach follows in the vein of using formal methods, as demonstrated by Kovalchuk et al. [16], who explored knowledge discovery in text (KDT) with the goal of categorising all legislation in their country. They combine topic modelling and FCA, requiring no prior domain knowledge. We aim to address a common challenge in text mining, namely, how to extract the meaning of a written message in terms of emotion, a branch of text mining known as sentiment analysis or opinion mining/subjectivity analysis. The primary goal of sentiment analysis is to detect the feelings expressed about a target, attempting to identify various emotions in the text, such as hatred, pain, joy, etc. The simplest form of sentiment analysis is polarity detection or binary classification of positive and negative sentiments.

In this paper, we propose that dictionaries can be automatically constructed, allowing for a more nuanced evaluation of the source texts. Our experiments suggest that the resulting lexicons are smaller, more specialised, and overall superior to those used in previous literature. Specifically, we present a supervised method that utilises the knowledge within the concept lattice of a formal context to build dictionaries for the use in sentiment analysis.

The paper is organised as follows: In Section 2, we discuss the background of dictionary construction methods and the use of dictionaries in textual sentiment analysis. Section 3 summarises the main notions about FCA and introduces the process of constructing a polarity lexicon from the knowledge obtained through FCA. In Section 4, we present experiments and comparisons demonstrating the promising results of our proposed method. Finally, we summarise our proposal and outline future work in Section 5.

2. Dictionaries for sentiment analysis

To tackle this problem of sentiment analysis, the most used strategy is that of expert-made dictionaries, also called lexicons, where each word has a proportional (positive or negative) weight according to its sentiment content. However, accurately gauging sentiment is complicated by several common linguistic subtleties such as negation, irony, ambiguity, idioms, and neologisms [13]. Therefore, these techniques require the experts to have access to the raw texts and are usually supervised.

Dictionary techniques typically consist of two main steps, as described in [3]:

1. The first step of dictionary techniques involves defining a list of keywords that are relevant and important for the analysis of the documents. This list of keywords, also known as dictionary, should include terms that are commonly used in the documents and that are indicative of their content and meaning.
   To create this dictionary, the researcher must carefully review the documents and identify the most important and relevant keywords. This process may involve preprocessing the documents to remove irrelevant information, such as stop words and punctuation, and to stem or lemmatise the words to reduce their variations. Once the relevant keywords have been identified, they can be used to create the dictionary, which will be used in the next step of the process.
2. Representing the documents in terms of the frequency of the keywords in the dictionary. This means that for each document, the frequency of each keyword in the dictionary is calculated and used as a feature for that document. This allows for the efficient representation of the content of the documents in a numerical form, making it easier to analyse and classify.
   For example, if the dictionary includes the keywords *love* and *hate*, and a document contains the sentence *I love chocolate but I hate broccoli*, the representation of that document in terms of the frequency of the keywords in the

Table 1
Number of positive and negative terms of the dictionaries taken as reference in this work.

	Positive terms	Negative terms
QDAP	1280	2952
GI	1637	2005
LM	354	2355
SentiWordNet	13459	16663
Subjectivity	1423	2471
Opinion	2006	4783

dictionary would be $(1, 1)$, indicating that the keyword *love* appears once and the keyword *hate* appears once in the document.

This two-step process allows for the efficient and effective representation of documents in terms of their content, allowing for further analysis and classification, such as determining the overall polarity of the document.

In practice, two types of dictionaries are commonly used for polarity detection in texts:

- Binary dictionaries: These dictionaries consist of two lists of terms, each representing one of two polarity categories: positive and negative. This type of dictionary is based on a binary classification of terms into these two categories.
- Weighted dictionaries: For each term m, a polarity score $\mathrm{s}(m) \in [-1, 1]$ is defined such that the closer this score is to 1 for a term, the greater the positive charge for that term, while values close to -1 indicate a greater negative charge. This type of dictionary allows for a more nuanced and fine-grained classification of terms based on their polarity.

In both cases, the dictionaries are used to classify the terms in the documents and to detect the overall polarity of the documents. It is clear that a weighted dictionary can give rise to a binary one, it being sufficient to define as a list of positive terms those m such that $\mathrm{s}(m) > 0$, and the list of negative terms in an analogous way. Therefore, weighted dictionaries are more general and versatile than binary dictionaries. We can consider a binary dictionary to be weighted with the function $\mathrm{s}(m) \in \{-1, 1\}$, where $\mathrm{s}(m) = 1$ if, and only if, m appears in the list of positive terms, and $\mathrm{s}(m) = -1$ otherwise.

With this last point, the estimated polarity for a text is given by the sum of the scores of each of the terms that appear in the text. If we denote T by the text and D by the dictionary used, we will have

$$\mathrm{p}_D(T) = \sum_{m \in T \cap D} \mathrm{s}(m)$$

i.e. the polarity estimated for a text is that induced by the terms that compose it (and are found in the dictionary).

The most common dictionaries in the literature are *QDAP* [11], *General Inquirer* or its modern version *Harvard IV* [22], *Loughran-Mcdonald* [18], the *Subjectivity lexicon* of [24], the *Opinion lexicon* of [12] and *SentiWordNet 3.0* [2]. A minimal summary of the number of terms in each of these dictionaries can be found in Table 1.

In the following section, we explain our proposal using FCA for building dictionaries, first, some main notions about this technique are shown.

3. FCA-based sentiment analysis

FCA [8] is a helpful mathematical tool to extract implicit information in an object-attribute table. We start this section with a brief overview of what FCA is. A formal context is a triple (G, M, I) where G is a set of objects, M is a set of attributes and $I \subseteq G \times M$ is the incidence relation, if $(g, m) \in I$ we say that the object g has the attribute m. A formal context is usually represented using a binary table where the rows are the objects, the columns are the attributes, and the crosses show the incidence relation in the cells. Consider the following running example.

Example 1. Consider the formal context (G, M, I) where $G = \{o_1, \ldots, o_7\}$, $M = \{a, b, c, d\}$ and the incidence relation is described by the following table.

	a	b	c	d
o_1		×		×
o_2	×	×		
o_3	×			×
o_4			×	
o_5		×	×	×
o_6	×		×	
o_7		×	×	

The two main operations in FCA are the so-called derivation operators, usually denoted by $^{\uparrow}: 2^G \to 2^M$ and $^{\downarrow}: 2^M \to 2^G$. These are defined as follows, let $X \subseteq G, Y \subseteq M$,

$$X^{\uparrow} = \{m \in M \mid (g, m) \in I, \text{ for all } g \in X\}$$
$$Y^{\downarrow} = \{g \in G \mid (g, m) \in I, \text{ for all } m \in Y\}.$$

The pair $(^{\uparrow}, ^{\downarrow})$ is a Galois connection between G and M, that is, for all $X \subseteq G, Y \subseteq M$ we have that $X \subseteq Y^{\downarrow}$ if and only if $Y \subseteq X^{\uparrow}$. It is well-known that both compositions $^{\uparrow\downarrow}$ and $^{\downarrow\uparrow}$ are closure operators. A couple $(A, B) \subseteq G \times M$ is said to be a formal concept if $A^{\uparrow} = B$ and $A = B^{\downarrow}$. The subset A is said to be the extent of the formal concept, and B is said to be the intent of the formal concept. Given a formal concept (A, B), all the objects in A share all the attributes in B and do not share any other attributes. Moreover, we can define an order relation between formal concepts: given two formal concepts (A, B) and (C, D), we say that $(A, B) \leq (C, D)$ if and only if $A \subseteq C$ (or equivalently, if and only if $D \subseteq B$). Indeed, this order relation defines a structure of complete lattice in the set of formal contexts, where the supremum and infimum are given by:

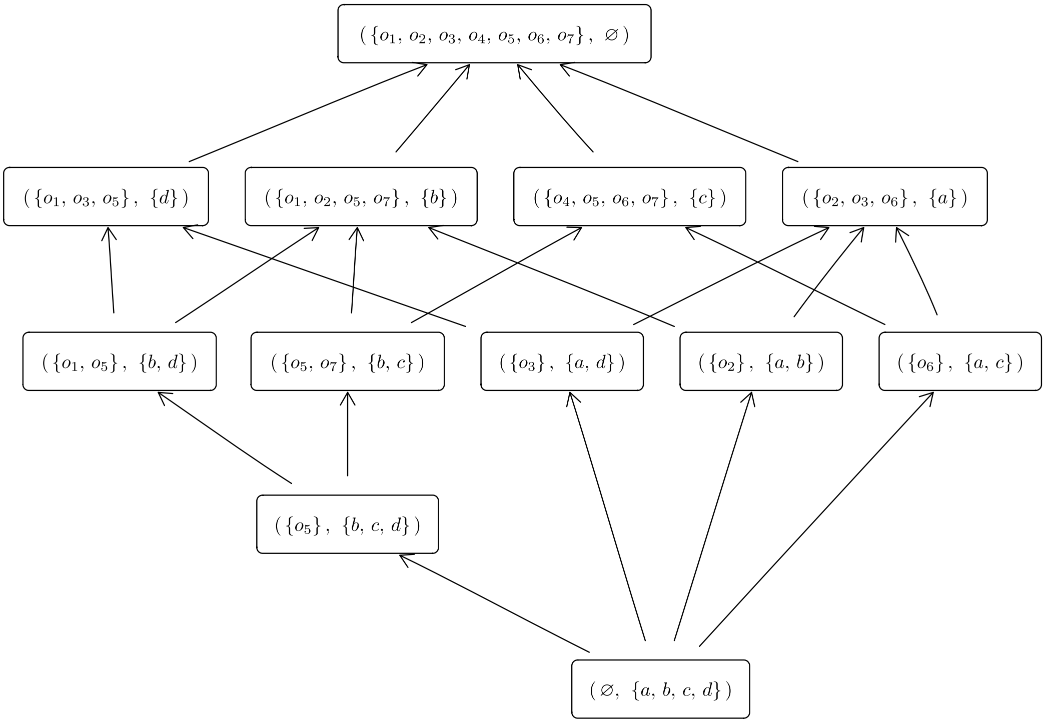
$$\sup_{j\in J}(A_j, B_j) = \Big((\bigcup_{j\in J} A_j)^{\uparrow\downarrow}, \bigcap_{j\in J} B_j\Big) \qquad \inf_{j\in J}(A_j, B_j) = \Big(\bigcap_{j\in J} A_j, (\bigcup_{j\in J} B_j)^{\downarrow\uparrow}\Big)$$

for any family of formal concepts $\{(A_j, B_j) : j \in J\}$. The complete lattice defined by this order is called the concept lattice of the formal context $\mathbb{K} = (G, M, I)$ and we denote it by $\mathbb{B}(\mathbb{K})$. In addition, it can be proved that every complete lattice L can be seen as a concept lattice of a certain formal context [7, Chapters 3 & 7].

Example 2. Consider the formal context from the Example 1. The pair $(\{o_1, o_5\}, \{b, d\})$ is a formal concept of the context since

$$\{o_1, o_5\}^{\uparrow} = \{m \in M \mid (g, m) \in I, \text{ for all } g \in \{o_1, o_5\}\} = \{b, d\}$$
$$\{b, d\}^{\downarrow} = \{g \in G \mid (g, m) \in I, \text{ for all } m \in \{b, d\}\} = \{o_1, o_5\}$$

The complete lattice of all formal concepts is the following



3.1. FCA-based lexicon construction for polarity analysis

Our proposal in this paper is to use document-term binary tables, as usual in text mining techniques, and apply the FCA machinery to build a *concise* dictionary usable for polarity analysis. Note that we propose to automate the construction of the polarity lexicon instead of relying on expert-made lexicons.

The rationale behind this proposal is that the concept lattice, applied to the study of terms and the specific vocabulary of a set of documents, can represent exhaustively co-occurrences and dependencies between terms, as can be deduced from the theory outlined above.

FCA is a data exploration technique that is commonly used in both supervised and unsupervised learning scenarios. In the case of supervised learning, FCA can be used to classify and detect patterns in datasets. In the context of this problem, we propose the use of FCA within a supervised learning scheme to extract a dictionary of terms from a set of labelled documents, with the aim of accurately detecting the polarity of other documents.

Thus, starting from a set of texts G and a set of terms M (meaningful for the documents), the matrix of documents and terms is constructed as the table I where $I(g, m) = 1$ if, and only if, the term $m \in M$ appears in the document $g \in G$. We can translate these definitions into the language of FCA, indicating that (G, M, I) is a formal context whose objects correspond to documents and whose attributes are the relevant terms selected from those documents, with $I \subseteq G \times M$ being the relation that indicates whether an attribute appears for a given object. Suppose, furthermore, that each document g has a polarity (positive or negative) assigned to it, which we can model employing an operator $\mathrm{p} : G \to \{-1, +1\}$, where $\mathrm{p}(g) = +1$ if, and only if, g has a positive polarity.

By determining the concept lattice of this context thus formed, the sets of terms that co-occur in the documents studied are being characterised: they are the *intents* of the concepts. On the other hand, it is logical to think that the set of documents that identifies the *extent* of a concept induces a notion of polarity in the concept itself. The following expression gives a simple way to aggregate the polarity information of the documents

$$\mathrm{p}(A, B) = \frac{1}{|A|} \sum_{g \in A} \mathrm{p}(g)$$

for all $(A, B) \in \mathbb{B}(G, M, I)$. In this way, we can speak of sets of *positive concepts* and *negative concepts*. Let us denote C^+ and C^- to these two sets.

We must not lose sight of the fact that we aim to define a vocabulary for the detection of the polarity of a text, then, following the explanations in Section 2, for each term $m \in M$, we must define a score about its polarity, $\mathrm{s}(m) \in [-1, 1]$ from which the polarity of any text can be inferred.

Our proposal is based on the fact that the terms that appear in the *intent* of a concept with a given polarity must have the same polarity direction as that concept, and the more times a term appears in the intents of concepts of a fixed polarity, the greater its weight will be. We then define for each $m \in M$:

$$\begin{aligned} \mathrm{s}^+(m) &= \sum_{(A,B)\in \mathrm{C}^+ : m \in B} \mathrm{p}(A, B) \\ \mathrm{s}^-(m) &= \sum_{(A,B)\in \mathrm{C}^- : m \in B} \mathrm{p}(A, B) \end{aligned} \tag{1}$$

which we will call the positive and negative charge of a term according to the concept lattice of (G, M, I). We then define the score of $m \in M$ as

$$\mathrm{s}(m) = \frac{\mathrm{s}^+(m) - \mathrm{s}^-(m)}{\mathrm{s}^+(m) + \mathrm{s}^-(m)}$$

The normalisation performed guarantees that $\mathrm{s}(m) \in [-1, 1]$ for all $m \in M$. Thus, the (weighted) dictionary that is generated has as terms the attributes in M, and their weights are those of the previous score. As mentioned before, we can build a binary dictionary from the latter just by selecting a list of positive terms those where $\mathrm{s}(m) > 0$ and similarly for the list of negative terms.

3.2. Improvements

In this section, we propose two improvements to the previous strategy. The first one aims to refine the dictionary creation, while the second tries to optimise the dictionary construction process.

On the one hand, by considering all the positive and negative concepts, we are not considering possible redundancies or repetitions of terms between related concepts. For this, the first improvement considers only those concepts whose intents are maximal either within C^+ or C^-. Calling $\mathrm{C}^+_{max} = \{(A, B) \in \mathrm{C}^+ : B \text{ is maximal}\}$ and analogously C^-_{max}, we can redefine the polarity score of a term of Equation (1) so that it only takes into account maximal concepts.

The second proposed improvement focuses on optimising the computation time of the concepts in order to generate the dictionary more efficiently. In this sense, the proposal sacrifices the exhaustiveness of the lattice for the sake of a faster computation: only concepts whose *extent* has a minimum cardinal (i.e. concepts with a minimum guaranteed support) will be considered. The benefit is that we can directly obtain these concepts without the need to compute the whole lattice since the algorithms for their construction (such as NextClosure [9], FastCbO [17,19] or InClose [1], to name a few) allow us to retrieve only the concepts that verify this minimum support condition. As a certain granularity of the lattice is lost in this process, it can be expected that dictionaries generated in this way will not perform as well as those generated with all the knowledge implicit in the lattice.

Table 2
Comparison between all the methods tested in this work. Results are averaged among all different DTM sizes.

Dictionary	Support	Accuracy	Sensitivity	Specificity	AUC
QDAP		0.723	0.419	**0.974**	0.696
GI		0.712	0.454	0.926	0.690
LM		0.631	0.220	0.970	0.595
SentiWordNet		0.693	0.490	0.860	0.675
Subjectivity		0.697	0.361	**0.974**	0.668
Opinion		0.727	0.429	0.973	0.701
all		0.815	**0.800**	0.827	0.813
maximal		**0.820**	0.773	0.857	**0.815**
supp	10%	0.807	0.771	0.837	0.804
	20%	0.807	0.771	0.837	0.804
	40%	0.806	0.773	0.833	0.803
	60%	0.806	0.773	0.832	0.802
supp+maximal	10%	0.808	0.746	0.858	0.802
	20%	0.808	0.746	0.858	0.802
	40%	0.806	0.742	0.858	0.800
	60%	0.807	0.745	0.857	0.801

4. Experimental results

In this section, we present the results of the experiments carried out to compare our proposal, based on the use of the concept lattice, with the dictionaries listed in Section 2. We have used the dataset *Twitter Tweets Sentiment Analysis for Natural Language Processing*, which is available on the Kaggle website [15], and comprises a collection of more than 16000 actual tweets that have been manually categorised according to the polarity (positive or negative) of the emotion they evoke. In the remainder of this section, we will talk about documents or tweets indifferently.

The idea is to pre-process the texts of the tweets in order to construct the document-terms matrix (DTM), selecting those terms that are relevant and significant. To do this, the most common *stopwords* (conjunctions, articles, etc.), punctuation marks and numbers have been removed, and terms of less than three characters have been eliminated. In order to select the most relevant terms, those that appear in a significant proportion of the documents (tweets) were chosen. As the number of columns in the document-terms matrix depends on this factor, in our analysis, we evaluated four different options for the size of the problem. The first option included terms that appeared in at least 50 documents, yielding a total of 314 terms. The second option only considered terms that appeared in 100 or more documents, resulting in 155 terms. The third option only included terms appearing in 150 or more documents, yielding a total of 86 terms. Finally, the fourth option required terms to appear in 200 or more documents, resulting in 54 terms used in the matrix.

In addition, we have carried out stemming of the terms found to take into consideration a single representative term from among all the terms of the same lexical family. Thus, for example, the term *amaz-* will be the representative of *amazing*, *amazed*, or *amaze*. In our experiments, we have found no difference in the results obtained with and without stemming, so we provide the former because stemming simplifies the problem and allows for greater versatility in the dictionaries.

In this way, we tested our proposal with different vocabulary sizes. For each size, we have compared our proposal in three versions: (a) the original proposal with all concepts; (b) the proposal considering the maximal sets of attributes; and (c) the improvement consisting of considering only concepts whose support (number of elements in their extent) is higher than a minimum threshold. For the latter case, we have considered different thresholds in our experiments: 10%, 20%, 40%, and 60% of the total number of documents. In addition, we have included a fourth alternative, consisting of the joint application of the two improvements (considering only the maximal elements while verifying minimum support). Hereafter, we will call these four alternatives **all**, **maximal**, **supp** and **supp+maximal**.

To thoroughly compare our proposals with previous dictionaries, we employed cross-validation with four folds. We assessed the effectiveness of each dictionary's classification using metrics such as accuracy (rate of correct classification), sensitivity (or true positive rate), specificity (true negative rate), and AUC (area under the ROC curve). A higher value in these metrics indicates superior performance. Among these metrics, AUC is particularly noteworthy as it measures overall performance, and will therefore be the primary metric used for comparison.

In Table 2, we present the comparison of these metrics for all the dictionaries analysed, averaging the results obtained for all the DTM sizes. We can see how the proposals presented in this work have better overall results (in terms of AUC value) than the rest of the dictionaries used in the experiments, although, going into detail, they do not achieve the high specificity values of the other dictionaries. This means that the other dictionaries detect negative polarity better, but if we consider the overall results, at the cost of having a higher false positive rate.

Among our proposals, **maximal** performs best, although its improvement with respect to **all** (with all concepts) is relatively small. Interestingly, the proposed dictionaries using only concepts with minimal support obtain slightly lower results than both **all** and **maximal**, while their computation is considerably more efficient, as discussed above.

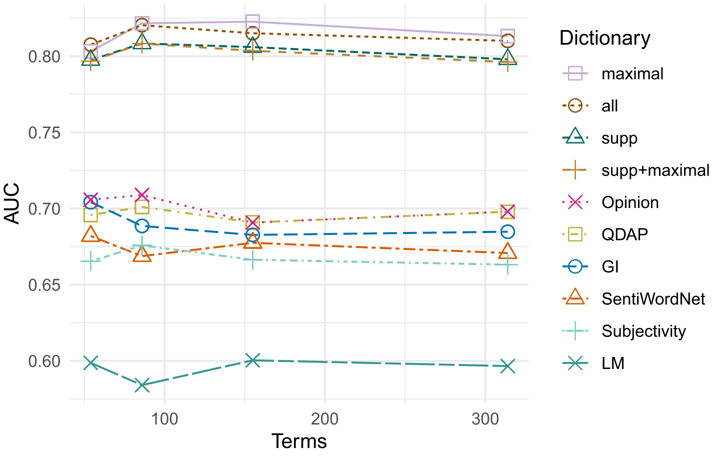


Fig. 1. Comparison of the AUC values obtained by the different dictionaries tested by the size of the DTM constructed.

Table 3
Average AUC achieved by each dictionary, according to the minimum support parameter used in the experiments.

Dictionary	Support	Number of terms in the DTM			
		54	86	155	314
all		0.808	0.820	0.815	0.810
maximal		0.803	0.822	0.823	0.813
supp	10%	0.798	0.811	0.805	0.801
	20%	0.798	0.811	0.805	0.801
	40%	0.798	0.811	0.806	0.798
	60%	0.797	0.808	0.806	0.798
supp+maximal	10%	0.792	0.810	0.805	0.801
	20%	0.792	0.810	0.805	0.801
	40%	0.791	0.808	0.804	0.796
	60%	0.797	0.808	0.804	0.796

Table 4
Average number of concepts used in the computation of each dictionary.

Dictionary	Support	Number of terms in the DTM			
		54	86	155	314
all & maximal		3755.75	6787.5	12204.5	19926.50
supp & supp+maximal	10%	2404.75	4371.0	7946.0	13467.25
	20%	2404.75	4371.0	7946.0	13467.25
	40%	1598.00	2756.5	4570.5	6879.50
	60%	1213.75	1978.0	4570.5	6879.50

In Fig. 1, we break down the average results for each DTM matrix size. We can see that our proposals perform better than the other dictionaries for all considered DTM matrix sizes. Furthermore, it can be concluded that the results are stable regardless of the DTM matrix size and that the best results are obtained for intermediate sizes (86 and 155 terms).

The following analysis studies the dependence of the results on the different thresholds for minimum support in the experiments. In Table 3, we can see that, as a general rule, the versions **supp** and **supp+maximal** have the same behaviour as **all** and **maximal**, obtaining the best results for intermediate DTM matrix sizes. We can also observe that the higher the threshold imposed for the minimum support, the worse results are obtained. However, this worsening is not marked, and even the most restrictive alternatives, with the minimum support at 60% of the number of documents, obtain competitive and clearly better results than those obtained with the other dictionaries in the comparison.

These results are especially interesting because, as we have mentioned before, the computation time is drastically reduced by not having to compute the whole lattice but only those concepts that verify the constraint on the minimum support. To estimate the magnitude of this reduction in computation time, we show in Table 4 the average number of concepts computed in the processing. In the most restrictive situation, as discussed above, it is only necessary, on average, to compute one-third of the concepts needed for the alternatives **all** and **maximal**. It should be noted that the calculation of the entire concept lattice, for the version with 314 terms, using the NextClosure [9] algorithm, takes only 20 seconds. With other, more efficient algorithms, this computation time would clearly be reduced. However, even for algorithms such

Table 5
Number of positive / number of negative terms in each of the constructed dictionaries.

Dictionary	Number of terms in the DTM			
	54	86	155	314
all	28.5 / 25.5	41.8 / 44.2	74.2 / 80.8	143.5 / 170.5
maximal	30.0 / 24.0	45.8 / 40.2	77.8 / 77.2	151.0 / 163.0
supp	28.8 / 25.2	43.7 / 42.3	78.2 / 76.8	151.6 / 162.4
supp+maximal	29.9 / 24.1	45.9 / 40.1	82.9 / 72.1	157.2 / 156.8

Table 6
Most relevant positive and negative terms in each of the dictionaries built in these experiments according to the assigned weights, for the case where the number of attributes was 86, in which the AUC results were the best ones.

all		maximal		supp		supp_maximal	
Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
amaz-	sad	amaz-	sad	excit-	ugh	excit-	ugh
awesome	hate	awesome	hate	amaz-	bore	amaz-	bore
love-	sorry	love	suck	enjoy	sick	enjoy	sick
nice	suck	nice	sorry	cool	hurt	cool	hurt
happi-	sick	cool	sick	awesome	suck	awesome	suck
				nice	hate	nice	hate
					sad		sad

as FastCbO or InClose, for DTM matrices with a larger number of terms, it may be necessary to consider only concepts with fixed minimum support to reduce the computation time.

To complete this study, we present, in Table 5, the average number of positive and negative terms found in the different dictionaries created, noting that the most remarkable balance between both quantities is precisely in the intermediate sizes of the DTM matrix, for which the best experimental results were given. It is worth comparing this table with the analogous one (Table 1) in the Section 2 to observe that the dictionaries created in this way are more concise and precise, as they are created *ad hoc* for a collection of documents. Finally, in Table 6, we present the list of the main positive and negative terms found (according to the weights calculated according to our proposal).

5. Conclusions and future works

In this paper, we have presented the use of Formal Concept Analysis (FCA) for creating dictionaries for sentiment analysis of text documents. The idea is to use the fine granularity of the concept lattice of the formal context associated with the document-term matrix of a collection of documents to describe more accurately the polarity of each relevant term.

We have tested our approach on a dataset of actual tweets manually labelled according to their polarity. The results of our experiments show that the proposed dictionaries based on FCA have better overall performance, in terms of AUC value, than the other standard and predefined dictionaries used in the experiments. These findings have important implications for the use of FCA in the sentiment analysis of tweets. Our results suggest that FCA can be a valuable tool for creating dictionaries that are effective in detecting the overall polarity of tweets, while also being efficient in terms of computation time.

Future work in this area could focus on further exploring and refining the use of FCA in sentiment analysis, with the aim of improving the performance of the dictionaries and reducing their size, as well as its use to define more generic and exhaustive dictionaries. Also, it will be interesting to analyse the potential use of fuzzy formal concept analysis to provide a more detailed description of the terms' polarity. Additionally, further research could investigate the use of FCA in other applications of natural language processing, such as topic modelling.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The script and data needed to replicate our experiments can be found at https://github.com/Malaga-FCA-group/demo-sentiment-analysis. To perform the FCA computations, we have used the **fcaR** package [6], which can be installed from the official CRAN repository.

Acknowledgement

This research is partially supported by the State Agency of Research (AEI), the Spanish Ministry of Science, Innovation, and Universities (MCIU), the European Social Fund (FEDER), the Junta de Andalucía (JA), and the Universidad de Málaga (UMA) through the FPU19/01467 (MCIU) internship and the research projects with reference PGC2018-095869-B-I00, TIN2017-89023-P, PID2021-127870OB-I00 (MCIU/AEI/FEDER, UE) and UMA18-FEDERJA-001 (JA/ UMA/ FEDER, UE).

References

[1] Simon Andrews, Making use of empty intersections to improve the performance of CbO-type algorithms, in: International Conference on Formal Concept Analysis, Springer, 2017, pp. 56–71.
[2] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani, Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining, in: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), 2010, pp. 2200–2204.
[3] David Bholat, Stephen Hansen, Pedro Santos, Cheryl Schonhardt-Bailey, CCBS Handbook No. 33, Text Mining for Central Banks, 2015.
[4] Claudio Carpineto, Giovanni Romano, Using Concept Lattices for Text Retrieval and Mining, Springer Berlin Heidelberg, 2005, pp. 161–179.
[5] Ángel Castellanos, Juan Manuel Cigarrán, Ana García-Serrano, Formal concept analysis for topic detection: a clustering quality experimental analysis, Inf. Sci. 66 (2017) 24–42.
[6] Pablo Cordero, Manuel Enciso, Domingo López-Rodríguez, Angel Mora, fcaR, Formal Concept Analysis with R, R J. (ISSN 2073-4859) 14 (2022) 341–361.
[7] Brian A. Davey, Hilary A. Priestley, Introduction to Lattices and Order, second edition, Cambridge University Press, Cambridge, 2002.
[8] Bernhard Ganter, Rudolf Wille, 'Formal Concept Analysis' Mathematical Foundations, Springer, Berlin, 1996.
[9] Bernhard Ganter, Two basic algorithms in concept analysis, in: International Conference on Formal Concept Analysis, Springer, 2010, pp. 312–340.
[10] Julia Hirschberg, Christopher D. Manning, Advances in natural language processing, Science 349 (2015) 261–266.
[11] Minqing Hu, Bing Liu, Mining opinion features in customer reviews, in: Deborah L. McGuinness, George Ferguson (Eds.), Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, AAAI Press/The MIT Press, 2004, pp. 755–760.
[12] Minqing Hu, Bing Liu, Mining and summarizing customer reviews, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 168–177.
[13] Dan Jurafsky, Christopher Manning, Natural language processing, Instructor 212 (998) (2012) 3482.
[14] Consuelo Justicia De La Torre, Daniel Sánchez, I. Blanco, María J. Martín-Bautista, Text mining: techniques, applications, and challenges, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 26 (04) (2018) 553–582.
[15] Kaggle Datasets, Twitter tweets sentiment analysis for natural language processing, https://www.kaggle.com/datasets/yasserh/twitter-tweets-sentiment-dataset, 2022.
[16] Pavlo Kovalchuk, Diogo Proença, José Borbinha, Rui Henriques, An unsupervised method for concept association analysis in text collections, in: International Conference on Theory and Practice of Digital Libraries, Springer, 2019, pp. 18–32.
[17] Petr Krajča, Jan Outrata, Vilém Vychodil, Advances in algorithms based on CbO, in: Marzena Kryszkiewicz, Sergei A. Obiedkov (Eds.), Proceedings of the 7th International Conference on Concept Lattices and Their Applications, Sevilla, Spain, October 19–21, 2010, in: CEUR Workshop Proceedings, vol. 672, CEUR-WS.org, 2010, pp. 325–337.
[18] Tim Loughran, Bill McDonald, When is a liability not a liability? Textual analysis, dictionaries, and 10-ks, J. Finance 66 (1) (2011) 35–65.
[19] Jan Outrata, A lattice-free concept lattice update algorithm based on *CbO, in: Manuel Ojeda-Aciego, Jan Outrata (Eds.), Proceedings of the Tenth International Conference on Concept Lattices and Their Applications, in: CEUR Workshop Proceedings, vol. 1062, CEUR-WS.org, 2013, pp. 261–274.
[20] Jonas Poelmans, Paul Elzinga, Alexei A. Neznanov, Guido Dedene, Stijn Viaene, Sergei O. Kuznetsov, Human-centered text mining: a new software system, in: LNCS, LNAI, vol. 7377, 2012, pp. 258–272.
[21] Kumar Ravi, Vadlamani Ravi, P. Sree Rama Krishna Prasad, Fuzzy formal concept analysis based opinion mining for crm in financial services, Appl. Soft Comput. 60 (2017) 786–807.
[22] Philip J. Stone, Earl B. Hunt, A computer approach to content analysis: studies using the general inquirer system, in: E. Calvin Johnson (Ed.), Proceedings of the 1963 Spring Joint Computer Conference, AFIPS, ACM, 1963, pp. 241–256.
[23] Kumar Ravi, Vadlamani Ravi, A survey on opinion mining and sentiment analysis: tasks, approaches and applications, Knowl.-Based Syst. 89 (2015) 14–46.
[24] Theresa Wilson, Janyce Wiebe, Paul Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, 2005, pp. 347–354.
[25] Samira Zad, Mark Finlayson, Systematic evaluation of a framework for unsupervised emotion recognition for narrative text, in: Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events, 2020, pp. 26–37.