# SABRE: A Sentiment Aspect-Based Retrieval Engine

**Annalina Caputo, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro and Gaetano Rossiello**

**Abstract** The retrieval of pertaining information during the decision-making process requires more than the traditional concept of *relevance* to be fulfilled. This task asks for *opinionated* sources of information able to influence the user's point of view about an entity or target. We propose SABRE, a Sentiment Aspect-Based Retrieval Engine, able to tackle this process through the retrieval of opinions about an entity at two different levels of granularity that we called aspect and sub-aspect. Such fine-grained opinion retrieval enables both an aspect-based sentiment classification of text fragments, and an aspect-based filtering during the navigational exploration of the retrieved documents. A preliminary evaluation on a manually created dataset shows the ability of the proposed method at better identify ⟨*aspect*, *sub-aspect*⟩ with respect to a term frequency baseline.

## 1 Introduction

Looking for others' opinions, impressions, and experiences is one of the first steps we usually perform when obliged to face a decision process. This could be the next president election, the booking of a room for the next holidays, or just the purchase

A. Caputo (✉) · P. Basile · M. de Gemmis · P. Lops · G. Semeraro · G. Rossiello
Department of Computer Science, University of Bari Aldo Moro, Bari, Italy
e-mail: Annalina.Caputo@uniba.it

P. Basile
e-mail: Pierpaolo.Basile@uniba.it

M. de Gemmis
e-mail: Marco.deGemmis@uniba.it

P. Lops
e-mail: Pasquale.Lops@uniba.it

G. Semeraro
e-mail: Giovanni.Semeraro@uniba.it

G. Rossiello
e-mail: Gaetano.Rossiello@uniba.it



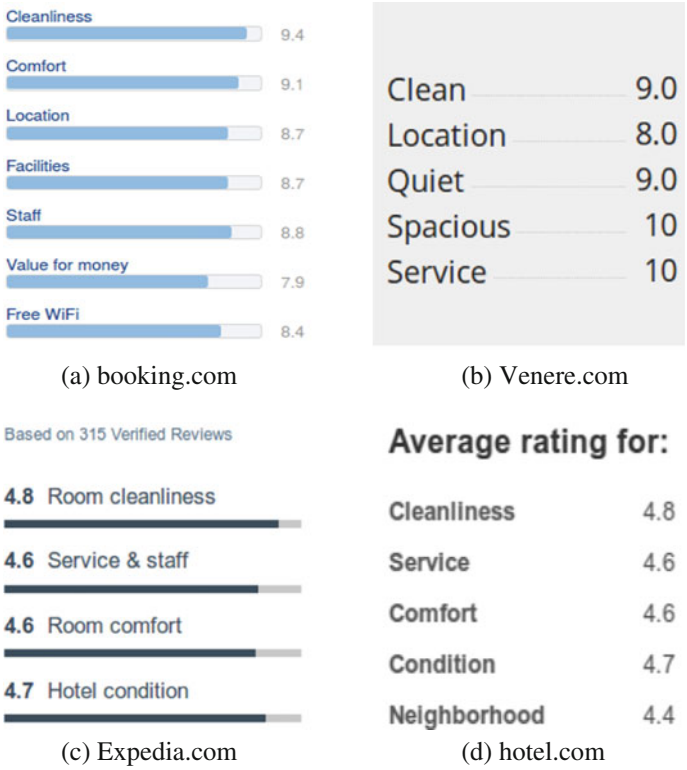
C. Lai et al. (eds.), *Information Filtering and Retrieval*,
Studies in Computational Intelligence 668, DOI 10.1007/978-3-319-46135-9_4

of a new product. Whatever the task, we start the process of making up our own opinion about a topic exploring both the available information and comments from others' experience. In this context, the concept of "relevance" is more than something pertaining an information need, like in a standard retrieval task. Indeed, valuable and relevant information should also bear a *subjective* point of view on a given topic or entity. Opinion retrieval (OR) aids such a process, since beyond the topical relevance of information retrieval (IR) systems, it requires documents to be opinionated.

Aspects play an important role in sentiment analysis and opinion mining. While the general sentiment or expression of opinion toward an entity is important to grasp the "overview" on a given subject, and can help during the initial investigation on a topic of interest, deeper in the process of decision-making, users are somehow more interested in specific aspects (or features) of interest. Classical examples are product reviews, where usually the user has a specific "aspect of interest" that leads her/him towards the thumb up/thumb down final decision. For example, searching for a hotel, someone may be more interested in the location, while others give more prominence to the value for money. These are perceived as different *aspect*s of the same *entity* (i.e. the target hotel). However, the extraction and organization of aspects from opinionated sources does not always match the user's interests and preferences. Usually, the assignment of aspects does not reflect the text content, but rather follows a manually created list of points of interest for a given domain. Figure 1 shows different lists of aspects from four well-known on-line booking services. The lists differ from one another, although there is some overlap, and this suggests that there is not a unique way of organizing aspects of interest for a given domain.

Generally, all entries cover broad aspects, but there is no way of further refining such a list. For example, Fig. 1a, c, d all report about "comfort". In two cases (booking.com and hotel.com), this aspect can refer to either room or hotel comfort. When referred to the room, the comfort aspect might be further refined as: bed/pillow comfort, spaciousness of the room, existence of special facilities (like Jacuzzi) or new furnitures, and so on. Moreover, since grades on aspects are given independently from the review, those aspects may not appear in the opinionated text.

This chapter describes SABRE, a Sentiment Aspect-Based Retrieval Engine, which takes into account aspects during both the process of sentiment classification of a given text and the navigation of retrieved documents (Sect. 2). Our main contribution is the aspect extraction algorithm, which processes text in a completely unsupervised manner to detect opinion bearing sentences and their subject. Aspects are organized in a two-level hierarchy in order to enable different levels of search granularity on the opinions of interest. The aspect extraction and weighing process is described in Sect. 3. Then in Sect. 4 we set forth an aspect-based retrieval model that takes advantages from the extracted aspects and their sentiment. Finally, we assess the proposed algorithm on a manually created dataset in order to validate its ability to recognize opinion-related aspects in a text at different levels of granularity (Sect. 5).




(a) booking.com (b) Venere.com

(c) Expedia.com (d) hotel.com

**Fig. 1** List of aspects from different hotel booking web sites

## 2 SABRE

An opinion is defined as a sentiment orientation expressed toward a given target, i.e. an entity or its attributes (commonly referred to as *aspects*). Although entities (like products, services, topics, issues, persons, organizations or events) and their aspects can be organized in a hierarchy of parts and sub-parts as nested nodes [7, 11] following the *part-of* relationships, most of the research in opinion mining/retrieval neglects such complex organization of concepts, and prefers a simpler model where the target of an opinion is generically an aspect, which denotes both parts and attributes. However, during a decision-making process many aspects at different levels of granularity can be involved.

For example, booking a room usually requires the matching of different criteria on a subjective base. *Cleanliness* and *view* can be considered as two sub-aspects of the general concept of *room*, which along with *location* represent two aspects of the entity *hotel* (Fig. 2).

However, most of the existing systems merely present a flat list of aspects. Such lists are predefined and manually created, they usually reflect broad coverage aspects,

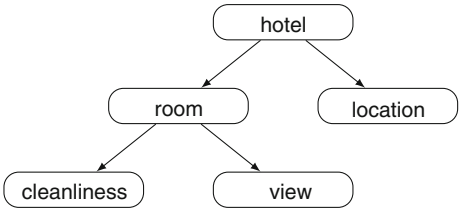



**Fig. 2** Entity/aspects/sub-aspects hierarchy

of which there is no guarantee of occurrence in the opinionated text. This is due to the fact that aspects are not extracted from text—then they not reflect the text content—but rather summarize, as a graded scale, the general opinion of the reviewer in few common points. Moreover, the hierarchical organization of aspects, which follows the *part-of* relationship, strictly depends on the target domain. *Soundtrack* is one of many aspects related to a *film*, but can be considered as an entity itself if we draw up the best top 100 soundtracks ever in the *music* domain.

This chapter proposes SABRE, an opinion retrieval system able to:

1. Extract from a given text aspects and their potential sub-aspects.
2. Associate to each aspect the corresponding opinion.
3. Detect the sentiment (positive or negative) of each opinion.
4. Retrieve documents which express an opinion about a given query.

A core component of SABRE is the aspect extraction one, which automatically extracts from text the hierarchy of aspects related to a given entity. However, in order to simplify the problem, the algorithm uses only the nodes at first level of the hierarchy (aspects) and considers all the sibling of this level as sub-aspects.

SABRE exploits such information for: (1) Re-ranking, during the second stage of the opinion retrieval, when the sentiment associated to each pair is exploited in combination to the relevance score obtained from the retrieval model; (2) Filtering, in order to improve the visualisation of reviews and help the user to filter out non relevant information during the navigation of the results.

To enable these operations, given $\Sigma$ the set of available aspects, and a document $D = (p_1, p_2, \ldots, p_n)$ split up in text units , SABRE extracts a set of quintuples in the form of $(p_i, a_{ij}, a_{ijk}, s^{rel}_{ijk}, s^{sent}_{ijk})$, where:

- $p_i$ is the text unit, it can be anyone of the possible ways of splitting a document, like sentences, paragraphs, or sliding windows;
- $a_{ij}$ are the main aspects, $a_{ij} \in \Sigma$;
- $a_{ijk}$ are the sub-aspect of $a_{ij}$, $a_{ijk} \in \Sigma \cup \{*\}$, with $*$ denoting the absence of sub-aspects;
- $s^{rel}_{ijk}$ is the relevance weight of the couple $\langle a_{ij}, a_{ijk} \rangle$ within the text unit $p_i$, $s^{rel}_{ijk} \geq 0$;
- $s^{sent}_{ijk}$ is the sentiment weight associated to $\langle a_{ij}, a_{ijk} \rangle$, it represent the polarity of the opinion expressed on that given pair, $s^{sent}_{ijk} \in [-1, 1]$.

The symbol $*$ denotes the lack of sub-aspects. Although a hierarchy defines the relationship between aspects and sub-aspects, the presence of a $\langle aspect, sub\text{-}aspect \rangle$in a quintuple does not imply by default the existence of $\langle aspect, * \rangle$; i.e. $(p_i, a_{ij}, a_{ijk}, \cdot, \cdot) \nRightarrow (p_i, a_{ij}, *, \cdot, \cdot)$. Several quintuples can be associated to the same text unit, representing in this way the possibility of different (and maybe contrasting) opinions on the same aspect/sub-aspect, like in the sentence "the hotel was clean, but quite noisy". Moreover, such a definition makes the retrieval of opinions on a target entity/aspect/sub-aspect easier, with the possibility of expressing constraints on $s_{ijk}^{sent}$, the polarity of the opinion.

## 3 Aspect Extraction

There are two main categories of aspect extraction algorithms: the *frequency-based*, which rely on statistical analysis of corpora, and the *topic modeling*, which make use of more sophisticated machine learning approaches. This work exploits two frequency-based approaches: a baseline method based on term probabilities, and a model that grasps the different use of language between a specific domain and a general context. Both these algorithm rely on the simple observation that aspects and sub-aspects frequently occur as nouns. On this assumption, we built the two different methods described below.

### *3.1 BASE: A Simple Frequency-Based Algorithm*

Frequency-based approaches compute statistics on term distributions from a training set, whose quality drives the effectiveness of the algorithm; when a new document come in, aspects are extracted on the base of their previously computed distributions.

The BASE algorithm, that will be used as a baseline algorithm, initially extracts from the training set of documents all the occurrences and co-occurrences of noun terms in a given sliding window $s$. During this process, the algorithm removes the stop-words, or the most $k$ frequent terms, in order to avoid too frequent and non informative words. Then, given the set of extracted terms $T = (t_1, t_2, \ldots, t_m)$, the algorithm computes (1) the probability of the term $t_i$ appearing as a noun in the sliding window and (2) the probability of a term $t_j$ appearing as a noun in the sliding window given the occurrence of the term $t_i$ as follows:

$$P(t_i) = \frac{freq(t_i)}{\sum_i freq(t_i)}. \tag{1}$$

$$P(t_j|t_i) = \frac{freq(t_i, t_j)}{freq(t_i)}. \tag{2}$$

Algorithm 1 shows the main steps. The list of extracted nouns represents the set of main aspects in the form $\langle a_{ij}, * \rangle$ with the associated relevance weight $s^{rel}_{ijk}$ given by $P(a_{ij})$.

Then, the list of pairs $\langle a_{ij}, a_{ijk} \rangle$ is built weighting each $\langle aspect, sub\text{-}aspect \rangle$ accordingly to the relevance weight given by $P(a_{ijk}|a_{ij})$. However, in this second step the algorithm takes into account only those terms co-occurring with the $N$ most frequent terms in the whole dataset (line 9). Then the algorithm keeps only the top $z$ aspects weighed accordingly to their relevance scores. We exploit the output of this algorithm as baseline.

**Algorithm 1** Aspect Extraction Baseline

**Require:** Unit text $T_i$, threshold $z$
**Ensure:** List $A$ of pairs $\langle aspect, sub\text{-}aspect \rangle$ with associated weight $s^{rel}_{ijk}$

1: $A \leftarrow newList()$
2: $N \leftarrow nouns(T_i)$
3: $NC \leftarrow nounCoOccurrences(T)$
4: **for all** $t_j \in N$ **do**
5: $\quad s^{rel}_{ij*} \leftarrow P(t_j)$
6: $\quad$ add $\langle t_j, * \rangle$ to $A$
7: **end for**
8: **for all** $\langle t_j, t_k \rangle \in NC$ **do**
9: $\quad$ **if** $t_k \in mostCommonNouns()$ **then**
10: $\quad\quad s^{rel}_{ijk} \leftarrow P(t_k|t_j)$
11: $\quad\quad$ add $\langle t_j, t_k \rangle$ to $A$
12: $\quad$ **end if**
13: **end for**
14: sort $A$ by $s^{rel}_{ijk}$
15: keep first $z$ elements of $A$
16: **return** A

## 3.2 LM: Measuring the Divergence Between Languages

This algorithm is based on the idea that language differs when talking about a specific domain with respect to a general topic; then, this method aims at selecting the aspects whose distributions in a specific domain diverge from those in a general corpus, like the British National Corpus[1] (BNC).

To this extent, we exploit the Kullback-Leibler divergence (KL-divergence), a non-symmetric measure of the difference between two distributions. The KL-divergence measures the relevance of a term with respect to the difference between two distributions—one computed on the specific domain while the other on a generic corpus—as the information that the term conveys.

[1] http://www.natcorp.ox.ac.uk/.

However, to compute such a difference in a specific point, we make use of the pointwise Kullback-Leibler divergence, defined as follows:

$$\delta_t(p\|q) = p(t)log\frac{p(t)}{q(t)}. \tag{3}$$

where $p$ is the distribution over the domain corpus and $q$ is the distribution over the general corpus. Differently from the KL-divergence, the pointwise KL-divergence can assume negative values of $\delta$, which correspond to non relevant aspects. However, in order to build the list of main aspects for a given text, we consider all noun terms $t$ with $\delta_t(p\|q) > \varepsilon$ ($\varepsilon \geq 0$ threshold). Let denote with $P_{domain}$ and $P_{general}$ the two distributions of a term on a domain and a general corpora. The term $t$ can be considered as a main aspect if

$$\delta_t(P_{domain}(t)\|P_{general}(t)) > \varepsilon. \tag{4}$$

The threshold $\varepsilon$ impacts the relevance of aspects in the domain corpus. However, the method still works for $\varepsilon = 0$, since in that case all non relevant aspects will take on $\delta_t < 0$. Another interesting point is that $\delta$ induces an order relation on the set of aspects: given two aspects $a_1$ and $a_2$, $a_1$ is more relevant than $a_2$ in the given domain if and only if $\delta_{a_1} > \delta_{a_2}$. The main steps of this method are showed in Algorithm 2.

**Algorithm 2** LM Main Aspect Extraction

**Require:** Unit text $T_i$, threshold $\varepsilon$
**Ensure:** List $A$ of main aspect with associated weight $s^{rel}_{ij*}$

1: $A \leftarrow newList()$
2: $N \leftarrow nouns(T)$
3: **for all** $t_j \in N$ **do**
4: **if** $\delta_{t_j}(P_{domain}(t_j)\|P_{general}(t_j)) > \varepsilon$ **then**
5: $s^{rel}_{ij*} \leftarrow \delta_{t_j}$
6: add $\langle t_j, \cdot \rangle$ to $A$
7: **end if**
8: **end for**
9: **return** A

### 3.2.1 Sub-aspect Extraction

The output of Algorithm 2 is a list of main aspects that represents the input to the algorithm for sub-aspect extraction. This phase exploits two measure of "quality" [17] of a sub-aspect defined as:

- **Phraseness**, the information lost following the adoption of a unigram ($LM^1$) in place of $n$-gram ($LM^N$) language model:

$$\varphi_{ph} = \delta_t(LM^N_{fg} \| LM^1_{fg}) \ ; \tag{5}$$

- **Informativeness**, the information lost when assuming that $t$ is drawn from $LM_{bg}$—the *background* or general—rather than $LM_{fg}$—the *foreground* or domain—language model:

$$\varphi_i = \delta_t(LM^N_{fg} \| LM^N_{bg}). \tag{6}$$

The phraseness measures the information lost when words are considered as independent in a unigram language model rather than as a sequence in a $n$-gram model. The informativeness measures the information lost when assuming that as sentence has been drawn from the background (general) rather than the foreground (domain) corpus, in the case in point this measures the information added from the domain to the terms.

Since the quality of the pair $\langle aspect, sub\text{-}aspect \rangle$ is conditioned by both these factors, we define the relevance weight of the pair as:

$$\varphi = (\varphi_{ph} + \varphi_i) \times \mathcal{N}(\sigma^2). \tag{7}$$

$\mathcal{N}(\sigma^2)$ is used for smoothing the phraseness and informativeness weights, which are strongly regulated by words with high pointwise KL-divergence. $\mathcal{N}(\sigma^2)$ replaces the variance-to-mean ratio (VMR), since we observed that the distribution of co-occurrences follows a normal distribution.

For each main aspect extracted accordingly to (3), Algorithm 3 computes the relevance score $\varphi$ for the pairs $\langle t_{ij}, t_{ijk} \rangle$, where $t_{ijk}$ is a noun extracted from the text fragment. The final list of $\langle aspect, sub\text{-}aspect \rangle$ consists in all the pairs for which $\varphi > \varepsilon$.

**Algorithm 3** LM Sub-Aspect Extraction

**Require:** Unit text $T_i$, main aspect list $A$, threshold $\varepsilon$
**Ensure:** List $A$ of pairs $\langle aspect, sub\text{-}aspect \rangle$ with associated weight $s^{rel}_{ijk}$

```
A ← newList()
N ← nouns(T_i)
for all t_j ∈ N do
    for all t_k ∈ N do
        compute φ for ⟨t_j, t_k⟩
        φ ← (φ_p + φ_i) × N(σ²)
        s^rel_ijk ← φ
        add ⟨t_j, t_k⟩ to A if φ > ε
    end for
end for
return A
```

### 3.3 Algorithm Extensions

Both frequency-based and language model divergence algorithms can be extended to consider phrases rather than terms. Specifically, we defined two possible extensions, which can be used either individually or combined:

Named entity recognition (NER): This module is based on conditional random fields [5] in order to extract sequences of words which correspond to three types of named entity: person, organization, and location. This model has been trained on the CoNLL 2003[2] dataset.

Collocation (CL): This module recognizes sequences of words that appear in the list of 50,000 bi-grams extracted from WordNet.

### 3.4 Sentiment Analysis

The algorithm used for assigning a sentiment score ($s_{ijk}^{sent}$) to each $\langle aspect, sub\text{-}aspect \rangle$ pair is a lexicon-based model that exploits the AFINN wordlist [12]. AFINN contains about 2500 English words that have been manually tagged with a score that can range from positive (+5) to negative (−5). The wordlist contains also some collocations while it disregards words with a neutral sentiment (score equal to zero).

For each text fragment, the algorithm computes the mean of sentiment scores associated to words in the text that appear in AFINN wordlist. However, when a "negation" word—like not, but, no, never, less, barely, hardly, rarely, aren't, weren't, won't, don't and isn't—is encountered, this reverses the sentiment score of all the words appearing at most at a distance of five terms from the negation. The score is rescaled in the interval $[-1, 1]$, and then associated to each $\langle aspect, sub\text{-}aspect \rangle$ extracted from the fragment.

## 4 Opinion Retrieval

The opinion retrieval engine is based on a two-stage approach:

1. A classical *tf-idf* vector space model [14] is employed for retrieving the top $N$ documents ranked accordingly to the query terms.
2. The opinion ranking module re-ranks the top $N$ documents in order to reflect their opinion scores.

While the document relevance in the first step is computed as in a standard vector space model ($tf\text{-}idf$ weighing schema), the second step exploits the collection of

[2] http://www.cnts.ua.ac.be/conll2003/ner/.

quintuples $(p_j, a_{ij}, a_{ijk}, s^{rel}_{ijk}, s^{sent}_{ijk})$ associated to each text unit, and specifically the relevance ($s^{rel}$) and sentiment ($s^{sent}$) weights of each pair $\langle aspect,\ sub\text{-}aspect \rangle$.

Let $S = \{s \mid s = \langle s^{rel}, s^{sent} \rangle,\ s^{rel}$ and $s^{sent}$ relevance and sentiment score of$\langle a_{ij}, a_{ijk} \rangle\}$ be the set relevance and sentiment score pairs extracted for all the $\langle aspect, sub\text{-}aspect \rangle$in a document, the opinion score is calculated as:

$$\frac{\sum_{s \in S} |s^{rel} \cdot s^{sent}|}{|S|} \tag{8}$$

The opinion score has the advantages of taking into account both the relevance of the opinion and its polarity, in addition it normalizes this value with respect to number of $\langle aspect, sub\text{-}aspect \rangle$pairs in a document.

Figure 3 shows the result set for the query "location breakfast" performed on a set of hotel reviews collected from TripAdvisor. Aspects extracted from the result set are listed on the left side of the main result list. Aspects are aggregated, on the basis of the pair $\langle aspect, sub\text{-}aspect \rangle$they belong to, with their sentiment scores computed on the whole set of retrieved documents. Duplicate aspects denote the presence of the same aspect belonging to different pairs $\langle aspect, sub\text{-}aspect \rangle$.

Moreover, a filter based on the extracted aspects can be applied on the result set through the "aspect filter" button, which opens the window showed in Fig. 4.

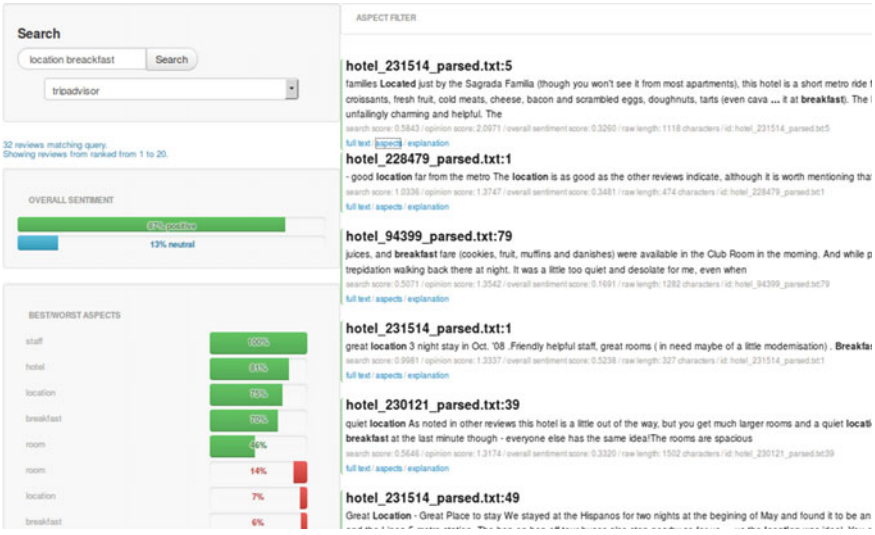



**Fig. 3** SABRE result page


**Fig. 4** Aspect filter

## 5 Evaluation

SABRE has been evaluated in two different ways with respect to its ability to extract aspect/sub-aspect pairs. Each evaluation is performed on a dataset of 164,780 reviews from TripAdvisor, where each review has been anonymized. The TripAdvisor dataset represents the specific domain corpus. As global domain corpus we exploit the British National Corpus (BNC), which consist of about 4000 documents with 100 million words from different domains. The threshold $z$ for the baseline is set to 30 aspects, while the threshold $\varepsilon$ for aspect and sub-aspect in the LM approach is set to $10^{-3}$; all the thresholds have been chosen after an empirical tuning of the system. The text has been analysed before the extraction of term distribution. The analysis comprises the tokenization, lemmatisation, stop-word removal, Part-Of-Speech tagging. Moreover, this pipeline includes also the named entity recognition and the collocation finding in the case of the two extensions explained in Sect. 3.3. Most of the text operations are performed by the Stanford CoreNLP API[3] [9], while the implementation of indexing and retrieval is performed on the top of ElasticSearch[4] engine.

[3] http://nlp.stanford.edu/software/corenlp.shtml.

[4] https://www.elastic.co/products/elasticsearch.

**Table 1** Aspect labelling evaluation: (P)recision, (R)recall, (F)-measure

| | $\langle aspect, * \rangle$ | | | $\langle aspect, sub\text{-}aspect \rangle$ | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| BASE | 0.256 | 0.869 | **0.396** | 0.092 | 0.526 | 0.154 |
| BASE-CF | 0.254 | 0.869 | 0.393 | 0.093 | 0.530 | 0.155 |
| BASE-NER | 0.251 | 0.868 | 0.389 | 0.093 | 0.527 | 0.154 |
| BASE-CF-NER | 0.255 | 0.870 | 0.394 | 0.094 | 0.533 | **0.157** |
| LM | 0.279 | 0.873 | 0.422 | 0.148 | 0.448 | 0.211 |
| LM + CF | 0.281 | 0.878 | **0.456** | 0.148 | 0.453 | 0.211 |
| LM + NER | 0.278 | 0.873 | 0.421 | 0.144 | 0.448 | 0.207 |
| LM + CF + NER | 0.274 | 0.878 | 0.418 | 0.153 | 0.465 | **0.230** |

We report in bold the best F-measure values for both the baseline and the language modeling systems in the two different experiments: $\langle aspect, * \rangle$ and $\langle aspect, sub\text{-}aspect \rangle$ identification

## 5.1 Aspect Labelling

The first evaluation method is based on a manually labelled dataset built on a random selection of 200 out of 164,780 hotel reviews from TripAdvisor. The remaining 164,580 reviews were used for training the model. The annotator had to specify a pair ⟨*aspect, sub-aspect*⟩ for each review in the test set. We compared our extraction algorithm (**LM**) against the baseline (**BASE**) testing several configurations with and without the use of the collocation (**CF**) and named entity recognition (**NER**) extensions.

Table 1 reports the results of the evaluation when only the main aspect is considered, i.e. reducing all the labelled pairs to $\langle aspect, * \rangle$, and when the proper ⟨*aspect, sub-aspect*⟩ is identified. As expected, figures for $\langle aspect, sub\text{-}aspect \rangle$ identification are lower than those for detecting the main aspect, this is due to the more complex task, that here asks for the identification of a hierarchy between the aspects mentioned in the review.

In both experiments, the language model system achieves better performance than the baseline, however is only in the ⟨*aspect, sub-aspect*⟩ identification that the configuration with CF+NER gives the best result. However, it is important to underline here that the reported values should be considered only as a lower bound due to: (1) the small number of review considered, (2) the manual labelling performed by just one user, and (3) the inherent subjectivity in assessing the ⟨*aspect, sub-aspect*⟩ pairs.

## 5.2 User Feedback

Given the list of ⟨*aspect, sub-aspect*⟩ pairs extracted during the aspect labelling evaluation from the best system (LM + CF + NER), we asked 61 users to manually tag a sub-set of the pairs extracted from 97 reviews as relevant/not-relevant with

**Table 2** User feedback evaluation: (P)recision, (R)recall, (F)-measure

| | P | R | F |
|---|---|---|---|
| $\langle aspect, * \rangle$ | 0.416 | 0.933 | 0.547 |
| $\langle aspect, sub\text{-}aspect \rangle$ | 0.232 | 0.879 | 0.351 |

respect to the review. The evaluation aims at finding out the number of $\langle aspect, sub\text{-}aspect \rangle$pairs the user find as prominent for the given review.

The assessment took place in two-steps:

1. The user selects the main aspects from the list. Each user is given from 3 to 6 aspects from which she/he has to select those relevant for the given text unit.
2. A list of sub-aspects is generated from aspects selected at the previous step, among these the user chooses those more relevant for the given main aspect and text unit.

Table 2 reports the result of this evaluation. The evaluation shows high values of recall, these figures are expected since the labelling is performed on the list of predefined aspects returned by the algorithm. More interesting in this context are the precision values, which are higher than those reported in Table 1, and similarly to the previous experiment we notice a drop in performance when the $\langle aspect, sub\text{-}aspect \rangle$has to be identified.

## 6 Related Work

The problem of opinion retrieval with respect to specific aspects/sub-aspects of interest is quite new, and to the best of our knowledge it has still to be addressed. However, if we consider each problem on its own, i.e. opinion retrieval and aspect-based opinion mining, they are two well rooted problems in their respective fields.

The opinion retrieval (OR) has been treated as an extension of the information retrieval (IR) task. Usually, OR is performed in two-stages. First a set of relevant document is retrieved, and then this set is re-ranked according to their opinion scores [7, Chap. 9] adopting machine learning or lexicon based approaches, this is also the approach adopted in SABRE. Most of the research on OR has been conducted within the TREC Blog Track evaluations, and all best systems participating in the opinion finding task of the three Blog Task evaluation (2006, 2007, 2008) followed such kind of strategy [6, 20, 21]. However, exception exists like the system proposed by Zhang et al. [19], where the two components are merged altogether.

Although we applied a simple lexicon-based approach, more sophisticated techniques have been developed to classify a sentence with respect to the sentiment it expresses. In addition to techniques based on the presence of some sentiment words [3], many methods are based on some machine learning techniques, both supervised [10, 13] and unsupervised [18].

Topic modelling is one of the main approaches adopted for aspect extraction. These methods are usually based on latent Dirichlet allocation (LDA) [2] or probabilistic latent semantic analysis (pLSA) [4], which are statistical methods for detecting the topics of a discussion. Then, they are exploited in the aspect extraction where each topic is an aspect. However, since these techniques try to capture the different distributions of terms in documents that treat different topics, their use in review domain is a bit tricky, due to fact that reviews tend to treat always the same topics. Titov and McDonald [16] propose a system based on two levels: the first one uses LDA for entity extraction, while the second extracts aspects considering only the neighbour of the given entity, neglecting the possibility of more level of aspect organization. Although some extension of LDA have been exploited to derive hierarchy [1, 8, 15], these methods are still too complex and require big training data and parameter tuning.

## 7 Conclusions

This chapter introduced SABRE, a two-stage aspect-based opinion retrieval system which takes into account hierarchy of aspects organized at two levels. We described the general system architecture, and explained how the information about aspects and sub-aspects is exploited for computing the opinion score during the re-rank, and for filtering during the navigation of the relevant documents.

The core of our system relies on the aspect extraction algorithm. We proposed to chose candidate terms exploiting the Kullback-Leibler divergence from a domain and a general purpose corpora. At an early stage, we conducted an evaluation to assess the capability of the proposed algorithm at extracting good candidate terms as aspects and sub-aspects. The evaluation demonstrated competitive results with respect to the baseline.

Most of current datasets for opinion retrieval rely on either TREC Blog Track or Twitter retrieval. None of them specifically focuses on aspect hierarchy extracted from the text. We plan to design a opinion retrieval evaluation that would benefit from such organization. As future work, we plan a thoroughly investigation for assessing the retrieval performance of SABRE.

## References

1. Blei, D.M., Griffiths, T.L., Jordan, M.I.: The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. J. ACM **57**(2), 7:1–7:30 (2010)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
3. Godbole, N., Srinivasaiah, M., Skiena, S.: Large-scale sentiment analysis for news and blogs. In: Glance, N.S., Nicolov, N., Adar, E., Hurst, M., Liberman, M., Salvetti, F. (eds.) Proceedings of the First International Conference on Weblogs and Social Media, ICWSM 2007, Boulder, Colorado, USA, 26–28 March 2007

4. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '99, pp. 50–57. ACM, New York, NY, USA (1999)
5. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Brodley, C.E., Danyluk, A.P. (eds.) Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, pp. 282–289. Morgan Kaufmann, 28 June–1 July 2001
6. Lee, Y., Na, S., Kim, J., Nam, S., Jung, H., Lee, J.: KLE at TREC 2008 blog track: blog post and feed retrieval. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Seventeenth Text REtrieval Conference, TREC 2008, Gaithersburg, MD, USA, 18–21 November 2008, vol. Special Publication 500-277. National Institute of Standards and Technology (NIST) (2008)
7. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. **5**(1), 1–167 (2012)
8. Lu, B., Ott, M., Cardie, C., Tsou, B.K.: Multi-aspect sentiment analysis with topic models. In: Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops. ICDMW '11, pp. 81–88. IEEE Computer Society, Washington, DC, USA (2011)
9. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
10. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of the 16th International Conference on World Wide Web. WWW '07, pp. 171–180. ACM, New York, NY, USA (2007)
11. Moghaddam, S., Ester, M.: Aspect-based opinion mining from product reviews. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '12, pp. 1184–1184. ACM, New York, NY, USA (2012)
12. Nielsen, F.Å.: A new anew: Evaluation of a word list for sentiment analysis in microblogs. In: Rowe, M., Stankovic, M., Dadzie, A., Hardey, M. (eds.) Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big Things Come in Small Packages, Heraklion, Crete, Greece, 30 May 2011, CEUR Workshop Proceedings, vol. 718, pp. 93–98. CEUR-WS.org (2011)
13. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing. EMNLP '02, vol. 10, pp. 79–86. Association for Computational Linguistics, Stroudsburg, PA, USA (2002)
14. Salton, G.: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, Upper Saddle River (1971)
15. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. J. Am. Stat. Assoc. **101**(476), 1566–1581 (2006)
16. Titov, I., McDonald, R.T.: A joint model of text and aspect ratings for sentiment summarization. In: McKeown, K., Moore, J.D., Teufel, S., Allan, J., Furui, S. (eds.) ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, 15–20 June 2008, pp. 308–316. Columbus, OH, USA (2008)
17. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis. Acquisition and Treatment, vol. 18, MWE '03, pp. 33–40. Association for Computational Linguistics, Stroudsburg, PA, USA (2003)
18. Turney, P.D.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, pp. 417–424. Association for Computational Linguistics, Stroudsburg, PA, USA (2002)
19. Zhang, W., Yu, C., Meng, W.: Opinion retrieval from blogs. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management. CIKM '07, pp. 831–840. ACM, New York, NY, USA (2007)

20. Zhang, W., Yu, C.T.: UIC at TREC 2006 blog track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Fifteenth Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, 14–17 November 2006, vol. Special Publication 500-272. National Institute of Standards and Technology (NIST) (2006)
21. Zhang, W., Yu, C.T.: UIC at TREC 2007 blog track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, 5–9 November 2007, vol. Special Publication 500-274. National Institute of Standards and Technology (NIST) (2007)