A Framework for Multi-Document Extractive Summarization of Reviews with Aspect-Based Sentiment Analysis

André Seidel Oliveira[1], **Anna H. Reali Costa**[1], **Eduardo R. Hruschka**[1,2]

[1]Computer Engineering and Digital Systems Department
Data Science Center (c^2d)
Universidade de São Paulo, Brazil

[2]Data Science Team
Itaú Unibanco
São Paulo, Brazil

{andre.seidel, anna.reali, hruschka}@usp.br

Abstract. *We propose an integrated framework, named Multi-Document Aspect-based Sentiment Extractive Summarization (MD-ASES for short), to automatically generate extractive review summaries based on aspects of a large database with reviews of items such as films, businesses, and companies. Such summaries are got by extracting a subset of sentences as they are in the reviews, based on some relevance criteria. In MD-ASES, initially sentences are grouped in terms of aspects identified as predominant in the reviews. Then, sentences are selected by the similarity of the sentiment expressed about a particular aspect to the overall sentiment of the dataset reviews. Our results show that MD-ASES can successfully preserve the average sentiment of the reviews while including the most important aspects in the summary.*

1. Introduction

As the amount of textual data rapidly increases on the internet, the demand for tools to dispose this information in clear and comprehensive ways grows as well. In this context, Natural Language Processing (NLP) techniques capable of automatically processing large volumes of text are of great relevance. In particular, textual information is abundantly found on platforms containing user reviews about items such as services, products, and films. These platforms also allow users to assign scores to those items. The score assigned to an item can inform its general quality, but does not provide specific information about what aspects are worth noting. In addition, it is rather hard to analyse a large set of reviews about a given item.

In order to provide a concise description of items, explaining the user's scores, we developed an extractive summarization framework for user reviews. As user's sentiment about items are of great importance, our framework is fully supervised by the so called *aspect-based sentiment analysis*. It is aimed to output a summary of multiple reviews that keeps the average sentiment of the whole set of reviews related to each aspect as much as possible. In a nutshell, our framework consists of two modules: (i) an aspect identification module and (ii) a sentiment analysis module. Together, they pick input sentences that better approximate the *average sentiment* towards each of the *most important aspects* in

the set of reviews about the same item. Although our framework can be applied on any sentiment analysis dataset, we here focus on businesses reviews from Yelp dataset[1].

The motivation behind our framework is that the aspect-sentiment aware summaries can provide a concise and meaningful description of the reviewed item, which can later be used to: (1) Support new users or marketing choices and (2) be utilized as a textual input to review-aware recommendation systems. In fact, it is shown in [Musto et al. 2017] that *review-aware* recommendation systems can improve granularity in user's preferences identification.

The remainder of this paper is organized as follows. In Section 2 we summarize related work. Section 3 presents the proposed method, followed by the Experimental Setup in Section 4. After presenting our experimental results in Section 5, we summarize our main contributions and conclude the paper in Section 6.

2. Related Work

We deal with a task that is referred to as *multi-document summarization* (MDS), where the system input is composed of multiple reviews. It is also classified as *extractive*, because the output summary consists of sentences extracted directly from the input reviews, as opposed to *abstractive* models, in which new authorial sentences are generated [Gupta and Gupta 2019]. Therefore, reference works in this field are general extractive MDS techniques.

Maximal Marginal Relevance (MMR)[Carbonell and Goldstein 1998] is a classical extractive technique used for MDS. It is an unsupervised method that rank sentences trying to reduce redundancy while maintaining relevance, often optimized as an Integer Linear Programming (ILP) problem. Graph-based ranking methods, like LexRank [Erkan and Radev 2004] and TextRank [Mihalcea and Tarau 2004] are also used. They make use of graph structures to represent sentence similarities and then inspect the graph edges and links to rank important sentences. They play an important role in recent extractive and abstractive MDS models, like in [Mallick et al. 2019].

Recent works in extractive MDS also make use of neural networks, where the network is supervised by summary examples in specialized datasets. [Yin and Pei 2015] applied Convolutional Neural Networks (CNNs) to create sentence embeddings and then to evaluate their feasibility to the summary. More recently, [Nallapati et al. 2017] proposed a neural model with Gated-Recurrent Units (GRU) [Chung et al. 2014] to perform the sentence selection as a binary decision problem.

The models mentioned above represent of the most important and defining characteristics of the subject as a goal of the summarization. Our framework differs from the surveyed related work because we focus on preserving the average feeling expressed in the sentences, thereby being supervised by the user sentiment scores of the items to create the summaries, instead of using the reference summaries for training.

The idea of a extractive summarization of reviews that relies on aspect-based sentiment analysis is also present in [Musto et al. 2019], where the authors propose a model to output *justifications* for recommendations. More precisely, they used sentiment analysis

[1]https://www.yelp.com/dataset

as a pre-processing step to only consider sentences with *positive sentiment* as candidates for the summary. Our contribution goes beyond having the whole optimization objective as the approximation of the average sentiment of users about important aspects, including also non-positive sentiments.

3. Proposed Framework: MD-ASES

User reviews of items — such as movies, products or companies — provide passionate opinions on different aspects. In particular, aspects that are defined for the item category or that are individually remarkable from the item itself may be frequently cited among the reviews. In addition, the sentiment of different users towards the same aspect may be similar, as the users were subjected to the same experience. For example, consider two reviews about the same restaurant:

- Review 1: "(1) Holy heck this *place* is amazing. (2) I love their *chicken tacos* they're by far my favorite";
- Review 2: "(3) Great *customer service* and all round awesome experience!! (4) U must try their *chicken tacos*, the best in town!1!!".

It is possible to identify four aspects described in the four sentences from these reviews: place (sentence 1), chicken tacos (sentences 2 and 4), customer service (sentence 3) and experience (sentence 3). Place, customer service, and experience are defining aspects for restaurants in general, while chicken taco is a remarkable aspect from this specific restaurant. In addition, the chicken taco may be more important than the other aspects for this restaurant, as both reviewers mentioned it. It is also worth noting that the restaurant is probably good, as the sentiment associated to words in the same sentences to the mentioned aspects like love, amazing, and awesome, are positive. As we are trying to summarize a set of data made up of these opinionated user reviews, we exploit aspect-based opinion mining in our approach.

Aspect-based opinion mining focuses on extracting aspects or features from opinionated text and analyzing sentiments to infer values of polarity associated with these aspects [Moghaddam and Ester 2012]. In order to finally provide a robust textual output with size constraints, we extract relevant sentences about aspects from the input reviews to compose the summary. In this sense, we are dealing with a *sentence-level extractive multi-document summarization* task.

The problem we are tackling can be defined as follows. Let R be a set of sentences from reviews about a specific item, $R = \{s_{1,1}, ..., s_{1,|s_1|}, ..., s_{i,j}, ..., s_{J,1}, ..., s_{J,|s_J|}\}$. Here, i represents the review index and j the sentence position in review i. J is the total number of reviews, $|s_i|$ is the number of sentences of the review i. Reviews have variable number of sentences, $|R| = |s_1| + |s_2| + ... + |s_J|$, and for each reviewed item there might be a different number of reviews J. The sentence-level extractive multi-document summarization challenge is to choose a subset $S \subset R$ that better represents the whole set R, constrained to a given number of sentences chosen by the user, N, *i.e.*, $|S| = N$, with $N \ll |R|$.

We propose a framework, called Multi-Document Aspect-based Sentiment Extractive Summarizer (MD-ASES), that makes sentiment-aware summaries of multiple reviews with variable word-length and no reference summary. MD-ASES is fed with a

set of textual reviews R about a topic and the desired number of sentences, N, for the summary. To create the summary S with relevant sentences, as well as to address the problem of redundancy, MD-ASES ranks aspects in the set of reviews and selects the N most important ones. Thus, the summary S has N sentences, each one referring to one of the most important aspects selected. Among the sentences that refer to the same aspect, the one that better approximate the sentiment of the whole set is chosen to compose the summary.

This framework relies on the assumption that each sentence in the set of reviews usually cites only one aspect and that the overall number of cited aspects is also larger than the number of sentences, N, in the summary. Thus, $N \ll |R|$ is desirable so that multiple candidate sentences are present for each aspect.

The framework starts by separating review sentences and transforming inflected words to their single base form, referred to as lemmas. It is done with a predefined dictionary that maps known words to lemmas $W \rightarrow L$. All resulting lemmas are candidate aspects to the subsequent aspect identification module that ranks the importance of each lemma in the set of reviews, and then selects the N lemmas with highest scores to be considered as important aspects. For our experiments, we utilized an adaptation of TF-IDF[Ramos et al. 2003] to rank aspects (details are given in Section 3.1). Then, sentences are reorganized in such way that each sentence containing the same aspects are clustered together, and sentences that do not mention any of the important aspects are excluded. Sentences that contain more than one important aspect are assigned to multiple clusters. We denote the set of clusters as $C = \{c_1, ..., c_N\}$, where $|C| = N$ and $c_n \subset R$, $n = \{1, ..., N\}$. After the clustering procedure, one sentence from each cluster is chosen to compose the summary. To do so, MD-ASES uses a sentiment classifier to calculate probabilities distributed over K sentiment classes $X = \{x_1, ..., x_K\}$ of cluster c_n:

$$SC(x_k, c_n) = P(Sentiment = x_k|c_n), k = 1, ..., K, n = 1, ..., N. \quad (1)$$

Having defined SC as the probability of a given Sentiment Class x_k for all sentences $s_{c_n,j}$ in cluster c_n, we can similarly define SS as the probability of the sentiment class x_k for sentence $s_{i,j}$:

$$SS(x_k, i, j) = P(Sentiment = x_k|s_{i,j}). \quad (2)$$

Sentences can now be scored by computing the difference between their sentiment distributions $SS(x_k, i, j)$ to the cluster sentiment distributions $SC(x_k, c_n)$:

$$score(i, j, c_n) = \frac{\sum_{k=1}^{K} abs(SC(x_k, c_n) - SS(x_k, i, j))}{K}. \quad (3)$$

The sentence (i, j) with the lowest score for each cluster, c_n, is then chosen to compose the summary, ***i.e.***:

$$ClusterSentence(c_n) = \arg\min_{(i,j)} score(i, j, c_n). \quad (4)$$

Our framework is summarized in Algorithm 1, which was designed to work with any implementation of aspect identification and aspect-aware sentiment analysis. For simplicity, we here compute the class conditional probabilities of each sentiment by using a Naive

Bayes approach, detailed in 3.2, but we shall note that other classifiers can be employed in the framework. The aspect identification procedure is addressed in the following subsections.

Input: Review set R for an item and the number of sentences N in the summary S
Output: Summary S with N sentences from R
(1) Perform data preparation: Reviews are divided into sentences and words are transformed into their respective *lemmas*;
(2) Select the N most relevant lemmas in the reviews. They correspond to the most important *aspects* of the reviewed item;
(3) Cluster sentences that mention the same aspect, and exclude sentences that do not mention any of the selected aspects;
for *each* *cluster* c_n ***in*** *aspect clusters* C **do**
  (4) Evaluate the *overall sentiment probability distribution* SC in K levels of the concatenated sentences in the cluster c_n;
  for *each* *sentence* $s_{c_n,j}$ ***in*** *cluster* c_n **do**
    (5) Evaluate the *sentence sentiment probability distribution* SS in K levels;
    (6) Store score (Equation 3) between the *sentence sentiment distribution* and the *cluster overall sentiment distribution*;
  end
  (7) Select the cluster sentence with minimal score (Equation 4) to compose the summary S;
end
(8) **return** S

Algorithm 1: MD-ASES Framework.

3.1. Selection of the most relevant lemmas

Recall that we initially process words into lemmas. Then, we use the standard procedure known as Term Frequency - Inverse Document Frequency (TF-IDF) [Manning et al. 2008] to select the most relevant lemmas to be considered as important aspects of the reviewed item. In our case, considering a set of review data on related items, $D = \{R_1, ..., R_i, ..., R_{|D|}\}$, TF-IDF assigns a score $sc_{R_i}(l)$ for each lemma l in the set of item reviews R_i according to its frequency $f_{w,R_i}(l)$ in R_i, the amount of sets of item reviews that cite l at least once $f_{w,D}(l)$, and the number of sets $|D|$,

$$sc_{R_i}(l) = f_{w,R_i}(l) \times log(\frac{|D|}{f_{w,D}(l)}). \quad (5)$$

Equation 5 follows the idea that if a lemma is not frequent in the whole dataset D but is frequent in a specific subset R_i about an item, it is important to that item, so it receives a high score. On the other hand, if it is frequent in both datasets, D and R_i, it is just a common lemma in the language and a low score is assigned. Lemmas that are not frequent have low scores as well. This metric is used in [Ramos et al. 2003] to sort documents in a set by their relevance. The authors calculate the sum of TF-IDF scores for all words in a document to estimate its importance. In our case, lemmas are sorted by their

importance in a document, so $sc_{R_i}(l)$ is used directly to select the most important lemmas of the reviewed item, and they are called aspects. So we infer lemmas from TF-IDF for each item, and pick the N lemmas with highest $sc_{R_i}(l)$ scores for each R_i, forming the set of aspects that will compose the final summary.

Table 1 shows three examples of aspect retrieving in the Yelp dataset, with $N = 5$. The Yelp dataset contains ratings and reviews about companies and services. As it can be seen, the important aspects often show the main services provided by businesses, with words like "pizza", "gym" or "pie", as well as unique features about them, like "welcoming", "functional", and "claustrophobic".

Table 1. Example of aspect retrieving in Yelp dataset.

Business Type	Highest TF-IDF word lemmas
Pizza restaurant	"pizza", "though", "slide", "welcoming", "absolutely"
Cafe	"hitch", "though", "refund", "john", "pie"
Gym	"gym", "functional", "claustrophobic", "consultant", "amenities"

3.2. Naive Bayes Sentiment Classifier

After completing the retrieval of the N aspects and grouping the sentences into N clusters, each mentioning a certain aspect, the general sentiment expressed in the cluster and the one expressed in each sentence is evaluated by a sentiment classifier. Then, we selected a sentence from each cluster based on the similarity of its sentiment in relation to the cluster's aspect to compose the final summary.

In this work, for simplicity we use the classic Naive-Bayes approach, in which the estimated probability of each level of sentiment in relation to an aspect is updated by evidence given by lemmas in the same sentence $s_{c_n,j}$ or in the same sentence cluster c_n to which the aspect is associated. So, for each $s_{c_n,j}$ or c_n, we have a set of lemmas $\{l_1, ..., l_O\}$, where $l \in L$, and L is the dictionary of possible lemmas, $|L| = N$. First, we calculate the number of times that each lemma in the training set is associated with the best rating of an item review in the dataset. In our case, we use the Yelp dataset, the item refers to *business* and the best rating corresponds to *five stars*.

We use the maximum likelihood assumption, where the proportion between the number of times a lemma is associated to a sentiment and the total frequency of the lemma on the dataset is the estimated conditional probability $P(l_i|Sentiment = x_k)$. Then, we use these estimated probabilities $P(l_i|Sentiment = x_k)$, as well as the estimated prior sentiment probabilities $P(Sentiment = x_k)$ to calculate the posterior probabilities for each sentence given the set of lemmas in a sentence $s_{c_n,j}$ or cluster of sentences c_n with Bayes Rules, stated as follows:

$$P(Sentiment = x_k|l_1, ..., l_O) = \frac{P(l_1, ..., l_O|Sentiment = x_k) \times P(Sentiment = x_k)}{P(l_1, ..., l_O)}. \tag{6}$$

The "naive" assumption in this case is that the evidences are independent in relation to the sentiment hypothesis, so that their conditional probability can be computed

from the multiplication of each evidence contribution:

$$P(l_1, ..., l_O|Sentiment = x_k) = P(l_1|Sentiment = x_k) \times ... \times P(l_O|Sentiment = x_k). \quad (7)$$

The sentence is classified by the sentiment associated to the higher probability in the distribution of K sentiments levels of sentiment, $K = 2$ in the binary case, and $K = 5$ in the fine-grained sentiment. For instance, Table 2 shows the binary sentiment distribution of the lemmas "opportunistic", "sculpt", and "excellency". While "opportunistic" has a strong bias to negative sentiments, "sculpt" and "excellency" are more related to positive sentiments.

Table 2. Example of lemma conditional binary probability distributions, considering $K = 2$ (positive or negative).

Lemma	Negative	Positive
opportunistic	0.83	0.17
sculpt	0.38	0.62
excellency	0.22	0.78

4. Experimental Setup

In this section we present details about the current implementation of the framework, as well as the batch of experiments designed to test its capability of making sentiment aware summaries.

4.1. Data Preparation and Implementation Details.

The system was implemented in Python3, utilizing the Natural Language Toolkit (NLTK) package[2]. Before any processing, sentences and words were split into tokens with *sent_tokenize()* and *word_tokenize()* functions. The Python project *langdetect* [3] was also used to detect review languages, so that non English or blank reviews were excluded.

In addition, all word tokens were stemmed (reduced to inflected words) and transformed to their respective lemmas, which are base words that are units of meaning for their flexed versions. For example, the words "produced" and "production" are transformed to their lemma "produce" that represents the concept behind both words. To map words into their lemmas, the NOW corpus[4] [Davies] word-lemma dictionary was used. At the end of the summarization process, extracted sentences were disposed as they were originally written.

We chose the Yelp Academic Dataset for our experiments because of the huge amount of reviews available, most in English. It contains over 8M reviews, based on five-level ratings, about approximately 200K businesses. We use binary (negative, positive) and fine-grained (five levels) sentiment distributions, i.e., we used $K = 2$ and $K = 5$, respectively. In the binary sentiment analysis the standard five levels ratings (five stars) from Yelp dataset were mapped into two levels, where one and two stars correspond to a negative sentiment; four and five stars are positive and three stars reviews were removed

[2]https://www.nltk.org/

[3]https://github.com/Mimino666/langdetect

[4]https://www.english-corpora.org/now/

from the training set. Considering the training methodology, Yelp reviews were firstly re-organized, sorting reviews by their businesses. Then, the businesses were split into training (70%), validation (15%) and test (15%) sets.

4.2. Naive Bayes Sentiment Analysis Module Optimization.

We optimized the Naive Bayes sentiment analysis module from a grid-search procedure, in which two hyperparameters were considered, checking it's accuracy on the validation set as (1) binary and (2) fine-grained sentiment classifiers.

Firstly, the normalization of sentiments was applied to address the problem of a biased training set. Each sentiment distribution for each word lemma was discounted by a given percentage, calculated by the number of reviews associated to the less frequent sentiment over the number of reviews associated to the actual sentiment.

Secondly, a parameter $h \in [0.5, 1.0]$ was considered to deal with weak evidences. As the majority of words in a text does not indicate a strong evidence towards any sentiment (probabilities nearly the same between sentiment levels), words in which the higher sentiment probability was less than h were not considered as evidences. In binary sentiments, the model considers every word lemma when $h = 0.5$ and only considers lemmas that have deterministic distributions ([0.0, 1.0] or [1.0, 0.0]) when $h = 1.0$, so $h_{K=2} = \{0.5, ..., 1.0\}$. In fine-grained sentiments, $h_{K=5} = \{0.2, ..., 1.0\}$.

The hyperparameters values got for binary and fine-grained sentiment classification are given in Table 3. For both binary and fine-grained sentiment-levels, the normalization of sentiments had better results. The best accuracy for fine-grained sentiment analysis, with $h_{K=5} = 0.20$, was 53.37%. While for binary sentiment analysis the accuracy was 82.74%, with $h_{K=2} = 0.70$.

Table 3. Best Naive Bayes configuration in the validation set.

Sentiment levels	Best accuracy	h	Normalized	Nº of evidences
Binary	0.8274	0.70	yes	15420
Fine-grained	0.5337	0.20	yes	36718

4.3. Experiments

We test the hypothesis that MD-ASES can keep the average sentiment of the summarized reviews as much as possible while bringing up most important aspects. From this perspective, we performed three experiments to validate this hypothesis on the test set.[5] In *Experiment 1* the Naive Bayes sentiment analysis module was tested for binary and fine-grained sentiment classification to estimate its accuracy in sentiment prediction, while comparing to gold standard models.

In *Experiment 2*, we evaluated to what extent the generated summaries preserve the sentiment of the original reviews. After summarizing the test set reviews, a binary sentiment prediction of the summaries is performed. The predicted binary sentiment of the summary should match the average sentiment of the reviews (the mean score in Yelp dataset mapped to two levels). Results are reported as accuracy in this sentiment matching task on test set. *Experiment 2* procedure is illustrated in Figure 1. Two versions of

[5]Codes for dataset manipulation, training and tests available on https://github.com/aseidelo/MD-ASES

the summarization framework were considered for both *Experiment 1* and *Experiment 2*: binary and fine-grained sentiment analysis for the selection of sentiment-aware representative sentences. Also, for *Experiment 2*, three summary sizes were tested, with the number of sentences $N = 1$ (small), $N = 5$ (medium) and $N = 10$ (large).

Finally, in *Experiment 3* we made a qualitative analysis of two summaries generated with N = 5 and binary sentiment analysis for the selection of sentiment-aware representative sentences.

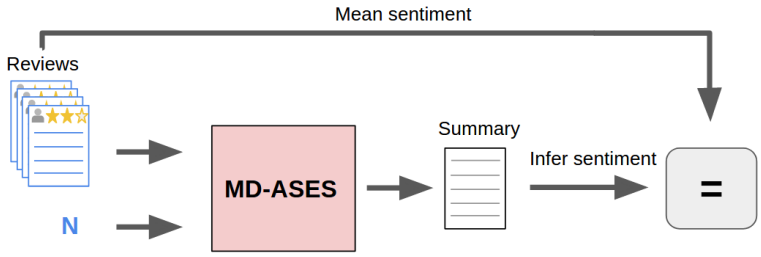

Figure 1. Experiment 2: In the analysis of binary sentiments, the average sentiment of the reviews is calculated and should correspond to the average sentiment of the summary of the reviews.

5. Experimental Results

For *Experiment 1*, Table 4 shows the results from the comparison between the optimized MD-ASES sentiment analysis module and state-of-the-art methods on Yelp dataset. Although Naive Bayes based approaches to sentiment analysis are not the gold standard, it shows reasonable results if compared with state-of-the-art models specifically trained for that matter in Yelp Dataset. For instance, our binary and fine-grained error had a difference around 6% - 7% to these deep learning models, while been fairly simpler.

Table 4. Binary and Fine-grained sentiment analysis classification error in Yelp Dataset (Experiment 1).

Method	Error (%)	
	Binary	Fine-grained
BERT large finetune UDA [Xie et al. 2019]	2.05	32.08
BERT large [Xie et al. 2019]	**1.89**	**29.32**
ULMFiT [Howard and Ruder 2018]	2.16	29.98
DPCNN [Johnson and Zhang 2017]	2.64	30.58
Proposed MD-ASES sentiment analysis module	9.64	36.66

The results for *Experiment 2* are detailed in Table 5. Accuracy in sentiment matching for the summaries are shown for three different summary sizes (N = 1, N = 5, and N = 10) and the two versions of the framework's sentiment analysis module (binary and fine-grained). Differently from what was expected, having more sentences in the summary did not augment accuracy in a significant level. It is also noticeable that the binary sentiment analysis module version had slightly better results than the fine-grained one (around 2%).

For the qualitative analysis of generated summaries, we now refer to two summarization examples shown in Table 6. It shows the whole summarization process of

Table 5. Accuracy on the sentiment matching test (Experiment 2) for N = 1,5,10.

	Summary size		
Sentiment levels (x)	Small (N=1)	Medium (N=5)	Large (N=10)
Binary	0.7610	0.7618	0.7542
Fine-grained	0.7498	0.7438	0.7330

12 reviews about **business 1** and 5 reviews about **business 2**, with summary length N = 5. The column *Name* represent a important aspect extracted with TF-IDF, the column *Sentiment Dist.* shows the binary sentiment distribution for each aspect, and the third column is the retrieved summary. The table also shows system's sentiment inference for the whole summary and the actual mean sentiment of the set of reviews. Due to privacy issues, personal and businesses names were exchanged to PNAME and BNAME.

For **business 1**, important aspects had highly one sided sentiment distributions, so the recommendation system retrieved only highly positive or negative sentences for the summary. In the second and third roll the same sentence was assigned to two different aspects (lash, vixen), so the system consider it only once. This summary is considered a true positive in Experiment 2, as the summary sentiment inference matches the mean sentiment of the set of reviews (both positive).

In the subsequent summarization example for **business 2**, also with summary length N = 5, the predicted mean sentiments for the aspects were predominantly negative and the system managed to pick sentences accordingly. We also noticed that aspects with balanced sentiment distributions, like "uncomfortably" in this example, were associated to less extreme sentences.

6. Conclusions

The proposed framework has shown to be successful in generating summaries that contemplate the overall users sentiment while bringing up important aspects about the businesses, with only a small subset of the reviews.

One relevant contribution of our work is the proposal of a model that is not supervised by reference summaries and that only needs data present in sentiment analysis datasets (reviews and scores), which are abundant online. Our framework can be applied with any algorithm for both aspect identification and sentiment analysis auxiliary modules. To meet this end, classic methods were utilized here because of the straight forward adaptation to multi-document summarization and to other languages as well. More importantly, by comparing the summary's inferred sentiment with the actual mean sentiment of the reviews, the proposed procedure innovates with a quantitative evaluation to sentiment-aware extractive summaries. For example, one could interpret the textual recommendation in Table 6 as: "This summary, with only 5 sentences, has a 76% chance of representing the mean users sentiment towards this business, while utilizing only $\frac{5\times100}{60} = 8,33\%$ of the sentences in the set of reviews".

Promising future works involve exploiting state-of-the-art sentiment classifiers and aspect identification, like in neural and reinforcement learning based models, while training with the same objective of representing the overall sentiment with less words. We shall also consider adapting the framework for abstractive summarization, so that the recommendation system could prospect for more adequate words to express reviews ideas.

Table 6. Examples of summarization for 2 businesses with 5 sentences (N = 5).

Aspect		Summary
Name	Sentiment Dist.	
business 1: avg. stars = 4.2 (Positive average sentiment)		
PNAME1	[0.015, 0.985]	"**PNAME1** really takes her time and is a perfectionist."
(eye) lash	[0.000, 1.000]	"If you live in the Lake Norman area and have been looking for an affordable quality eye **lash** extension bar, I highly recommend **BNAME** Lash Studio."
BNAME	[0.000, 1.000]	"If you live in the Lake Norman area and have been looking for an affordable quality eye **lash** extension bar, I highly recommend **BNAME** Lash Studio."
PNAME2	[0.083, 0.917]	"**PNAME2** is always professional and precise when it comes to lashes."
groupon	[0.992, 0.008]	"These people never answer the phone or respond to voicemails left by **Groupon** customers."
		Summary sentiment: Positive
business 2: avg. stars = 2.7 (Negative average sentiment)		
diligence	[0.879, 0.120]	"He challenged my integrity, belittled me and ultimately refused to do anything near due **diligence** in regards to my title search."
PNAME3	[0.996, 0.003]	"I rarely take the time to review a company, but **PNAME3** was the most condescending arrogant and unprofessional I have encountered in some time."
uncomfortably	[0.614, 0.385]	"Not a big deal , I'll **uncomfortably** walk to the back offices to get someone if have to."
broker	[0.019, 0.980]	"Professional, intelligent, organized and personal describe this office whether you were a realtor or **broker** or an individual selling your home and you happen to be in the Las Vegas or Henderson area this is probably the best place you could go."
title	[0.671, 0.329]	"Run Run Run to another **title** company."
		Summary sentiment: Negative

7. Acknowledgments

The authors acknowledge the support of CNPq under grants 425860/2016-7 and 307027/2017-1. This research is being carried out with the support of *Itaú Unibanco S.A.*, through the scholarship program of *Programa de Bolsas Itaú* (PBI), and it is also financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Finance Code 001, Brazil. Any opinions, findings, and conclusions expressed in this manuscript are those of the authors and do not necessarily reflect the views, official policy or position of the Itaú-Unibanco, CAPES and CNPq.

References

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Davies, M. Corpus of news on the web (now): 3+ billion words from 20 countries, updated every day. https://digital.library.unt.edu/ark:/67531/metadc1234358/. Accessed August 25, 2020.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Gupta, S. and Gupta, S. K. (2019). Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Johnson, R. and Zhang, T. (2017). Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

Mallick, C., Das, A. K., Dutta, M., Das, A. K., and Sarkar, A. (2019). Graph-based text summarization using modified textrank. In *Soft Computing in Data Analytics*, pages 137–146. Springer.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Moghaddam, S. and Ester, M. (2012). On the design of lda models for aspect-based opinion mining. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 803–812.

Musto, C., de Gemmis, M., Semeraro, G., and Lops, P. (2017). A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pages 321–325.

Musto, C., Rossiello, G., de Gemmis, M., Lops, P., and Semeraro, G. (2019). Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 383–387.

Nallapati, R., Zhai, F., and Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the 1st Instructional Conference on Machine Learning*, volume 242, pages 133–142. Piscataway, NJ.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. (2019). Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Yin, W. and Pei, Y. (2015). Optimizing sentence modeling and selection for document summarization. In *24th International Joint Conference on Artificial Intelligence*.