Aspect-Based Sentiment Quantification

Vladyslav Matsiiako, Flavius Frasincar, and David Boekestijn

Abstract—In the current literature, many methods have been devised for sentiment quantification. In this work, we propose AspEntQuaNet, one of the first methods for aspect-based sentiment quantification. It extends the state-of-the-art QuaNet deep learning method for sentiment quantification in two ways. First, it considers aspects and ternary sentiment quantification concerning these aspects instead of binary sentiment quantification. Second, it improves on the results of QuaNet with an entropy-based sorting procedure instead of multisorting. Other sentiment quantification methods have also been adapted for ternary sentiment quantification instead of binary sentiment quantification. Using the modified version of the SemEval 2016 dataset for aspect-based sentiment quantification, we show that AspEntQuaNet is superior to all other considered existing methods based on obtained results for various aspect categories. In particular, AspEntQuaNet outperforms QuaNet often by a factor of 2 on all considered evaluation measures.

Index Terms—sentiment analysis, sentiment quantification, aspect-based sentiment quantification

1 Introduction

WIDESPREAD usage of the Web has enabled customers' instantaneous and elaborate sharing of feedback or reviews, conveying useful information for businesses. This kind of information could arguably be more reliable than questionnaires people are often unwilling to answer, while simultaneously being more readily available.

Considering the speed at which the amount of data (particularly textual data like user reviews, comments, and forum posts) on the Web increases, it becomes practically impossible to analyze all of the data manually. Nevertheless, analysis of this data remains an important task for both the private and public sectors. For instance, restaurant chains prefer knowing as precisely as possible what people think about their new meals [1], and financial analysts could use sentiment retrieved from Tweets to potentially explain price swings in stocks [2]. It is therefore essential to use algorithms specifically designed for extracting and determining the sentiment of texts, which could afterward be used to create valuable insights.

The described branch of research is called sentiment analysis (SA) [3]. Its main objective is determining the sentiment towards a certain entity. While general SA methods aim to determine the sentiment of an entire text [4], aspect-based sentiment analysis (ABSA) is concerned with identifying the sentiment level per aspect (feature) within that text [5]. For example, the following sentence contains two aspects *scenario* and *acting*.

"The scenario of that movie was great, but the acting could have been better."

In this sentence, the sentiment towards the aspect *scenario* is positive, while the sentiment towards *acting* is negative. Using the methods of ordinary SA, we would run into the issue that the review conveys both negative and positive sentiment. However, by performing the SA task on an aspect level, we would obtain separate sentiment scores for each of the involved aspects. Therefore, with the methods of ABSA, we would be able to obtain more in-depth insights. For example, knowing particular sentiment scores for each feature of a certain product could provide useful information on how to improve it, and importantly, which parts to improve.

While for some use cases personal sentiment scores are important (e.g., in CRM marketing one wants to target every user separately) [6], there are many cases in which only aggregate numbers matter. For example, for political campaigns, it is only important to know the share of the population which supports a certain policy or candidate, while it makes no difference to know sentiment estimates on the individual level [7], [8]. Another example would be estimating the severity of an epidemic by means of identifying medical reports with a certain diagnosed disease [9]. For this reason, there exists a branch of research that explores sentiment quantification (SQ) [10].

Until now, despite the abundance of aspect-based sentiment classification methods, there has been little research connecting SQ with ABSA [11], [12], [13]. Hence, the novelty of this research lies in bridging the gap between those two fields. In particular, it compares the performance of usual sentiment quantification techniques on the aspect level as well as proposes a novel method, AspEntQuaNet, based on current state-of-the-art findings from both SQ and ABSA. All researched quantification methods are also extended to be applied to ternary (positive, neutral, and negative) instead of binary (positive and negative) sentiment classification. Additionally, for the purposes of this paper, we introduce the term aspect-based sentiment quantification (ABSQ), a field at the intersection of SQ and ABSA.

The paper is structured as follows. In Section 2, relevant literature concerning SQ and ABSA is provided. Section 3 outlines the data used in this paper. Next, Section 4 presents detailed explanations for the proposed methods. Following, results obtained using the investigated methods are assessed in Section 5. Finally, Section 6 provides a summary of the main conclusions of this paper, as well as suggests directions for future research. All methods and models researched in

• *Vladyslav Matsiiako, Flavius Frasincar, and David Boekestijn are with the Econometric Institute, Erasmus University Rotterdam, Burgemeester Oudlaan 50, 3062 PA Rotterdam, the Netherlands.*
Email: matsiiako@gmail.com, {frasincar,boekestijn}@ese.eur.nl

this paper are programmed in the Python programming language; source code can be found at https://github.com/vlad-matsiiako/ABSQ.

2 Related Work

As identified in the works of [3] and [14], SA is mainly concerned with locating opinion-related parts of texts, determining the sentiment they convey, and quantifying it. There are three main levels of sentiment identification: document-level, sentence-level, and aspect-level. The first two aim to determine the sentiment level within, respectively, a document or a sentence. When it comes to the aspect level, both documents and sentences may include multiple aspects towards which sentiments are expressed. This paper focuses on the aspect level.

A comprehensive survey of ABSA was provided by [15]. As defined by the authors, its goal is to find the sentiment towards each aspect of a certain entity. More mathematically, the task of ABSA equates to finding a quadruple of (s, g, h, t) [16]. In this quadruple, s denotes sentiment, g denotes the target, h denotes the holder (the subject that is expressing the sentiment), and t denotes the time point of the expressed sentiment. In reality, most methods, including the ones proposed in this paper, aim to only identify the pair (s, g).

The three main processing steps constituting ABSA were defined by [17] to be identification, classification, and aggregation. In the first step, the pairs of opinions and targets in a piece of text are identified. In the classification step, a certain algorithm maps the opinion onto a predetermined set of sentiment values (positive, neutral, or negative) [18], [19]. Finally, the obtained sentiment values are aggregated per aspect to generate a certain kind of sentiment report. This paper focuses on the last step: aggregation.

The strong point of ABSA is its greater utilization of the information given in a text compared to SA. However, it is essential to define the granularity of the utilized texts, because, for ABSA, it could vary from a piece of text as small as a sentence to a large document. In this paper, we aim to carry out ABSA on the sentence level.

It should be mentioned that both sentiments and aspects can be explicit or implicit. E.g., the sentence “I expected much more.” has an explicit sentiment (expectations were not met), but only an implicit aspect since it is not mentioned in the text what was “expected much more” of. According to [20], who analyzed a different dataset of restaurant reviews proposed by [21], implicit aspect targets do not occur often. For this reason and considering that the proposed methods rely on identified aspects, the possibility of implicit aspects will be ignored in this paper. On the other hand, implicit sentiments will not be disregarded.

2.1 Sentiment Quantification Task

Historically, SA was considered a sentiment task [22]. The idea was to classify each piece of text separately and then aggregate the results in a certain way. However, there are situations in which a different approach, oriented on aggregation, is needed. For instance, when market researchers try to estimate the share of the population that likes a newly-released product, they do not care about each individual separately, but rather care only about aggregate results. It has been argued in various examples that the idea of classification and further aggregation gives a suboptimal performance [23], [24], [25]. Therefore, it is worth investigating whether it is possible to design algorithms that outperform simple classification. This area of research is called Sentiment Quantification (SQ) and was first introduced by [10].

At first sight, optimizing for classification seems equal to optimizing for quantification. This is not true. The intuition behind the fact that simple classification returns suboptimal performance on the aggregate level is implicit in the measure that is used for evaluating performance. To illustrate this, we look at the F_1 score, considered to be a common measure for classification algorithms:

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \tag{1}$$

where TP is the number of true positive predictions, FP represents the number of false positives, FN is the number of false negatives, and TN equals the number of true negatives. Consider a sample of 100 reviews with binary sentiments, for which the true share of positives vs. negatives is 70:30. Compare one classifier obtaining the results $TP = 50, FP = 20, TN = 10, FN = 20$, with another classifier obtaining different results $TP = 70, FP = 10, TN = 20, FN = 0$. The latter model achieves the better F_1 score of 0.9333, while the former achieves a score of only 0.7143. However, for the SQ task, the latter model gives an estimate of 80:20 positive vs. negative sentiments, while the former model returned the ideal answer: 70:30.

In mathematical notation, this means that to optimize for SQ tasks, it is desirable to reduce $|FP - FN|$ rather than $(FP + FN)$. For this reason, multiple other measures that could compensate for the inaccuracy of a classifier were proposed in [26] and will be discussed further in this paper. Moreover, as discussed by [23], adjusting the classifiers for the purposes of SQ requires only a limited amount of training data compared to the size of the datasets one would need to improve the accuracy of the actual classifiers.

2.2 Development of Quantification Methods

One of the initial works in the field of quantification was published by [27]. The author was essentially the first to argue that in real-world situations, the assumption of identical characteristics between training and testing data is not necessarily true. In this work, he proposes an iterative procedure based on the “expectation-maximization” algorithm for increasing the likelihood of the new data. Additionally, a statistical test was developed to determine whether the initial class distribution is different from the real-world observations.

The next big leap in the direction of quantification was made by [28]. In his research, three methods were named and proposed: “Classify and Count”, its adjusted variant, and “Mixture Model”. The Mixture Model was found to outperform the others. However, since the first two methods have been much more widely used in academic literature, we will consider “Classify and Count” and its adjusted variant as baselines in this paper.

The previous line of research was further continued by the work proposed in [23] and [29]. Here, the author presents a new branch of quantification called "cost quantification". The idea is that in the case of attributed cost available and relevant to the problem (e.g., estimating the amount of time per day it takes for support agents to reply to all the questions), cost quantification might be preferred to ordinary quantification.

Since then, there have been multiple different directions in which the previous research developed. [30] explored the use of probability estimations of the classifier while applying an additional scaling factor similar to the one developed by [27]. In later research, these methods are called "Probabilistic Classify and Count" and "Probabilistic Adjusted Classify and Count", and will thus be referred to as such in this paper.

[31] have proposed one of the first methods that would not be based on classification to derive a quantifier. Instead, the authors designed a decision tree-based method optimized directly for quantification. Their results proved this method to be the new state-of-the-art at that time.

Additionally, [32] have explored the direction of ordinal text sentiment quantification. The authors present OQT, a tree-based method, which they use to quantify the data based on five different sentiment classes (on a scale from *very negative* to *very positive*). Even though the authors obtained promising results, there has been little following research in this domain. In this paper, we have opted for ternary sentiment quantification, with sentiment classes positive, neutral, and negative.

Based on the SVM for Multivariate Performance Measures (SVM_{perf}) developed by [33], [34] developed an SVM-based method able to optimize for multivariate loss functions created specifically for the quantification task. When released, their method outperformed all previously existing ones and became the new state-of-the-art in SQ.

Eventually, [35] came up with a new quantification-specific method, based on neural networks, which the authors called "QuaNet". Tested on an SQ text dataset, it was proven to substantially outperform all existing methods. This method will be used as a starting point for the quantification task and further advanced in the work described in this paper. To the best of our knowledge, our work is one of the first to adapt SQ methods for ABSQ.

3 Data

This section describes the datasets used in this paper. In Section 3.1, the datasets, their structure, some basic characteristics, and the employed preprocessing steps are discussed. Subsequently, Section 3.2 provides important insights into the datasets with regards to the distribution of data among aspect categories and sentiment classes of review sentences.

3.1 SemEval 2016 Dataset

The ABSA dataset used in this paper is taken from the SemEval competition held in 2016. In particular, we use task 5 of SemEval 2016 [36]. This domain dataset is considered an obvious choice for ABSA and is used in many papers researching this subject [37]. Additionally, there is also a dataset from task 12 of SemEval 2015 [38].

Both datasets contain restaurant reviews. Reviews are split into sentences, each of which might convey multiple sentiments depending on the number of target words within the sentence. Each opinion has a corresponding target and target category (FOOD#QUALITY, SERVICE#GENERAL, etc.), as well as a polarity (positive, neutral, or negative). The distribution of reviews' opinions among various polarity classes is given in Table 1 [18].

Table 1: Distribution in percentages of review sentences among sentiment classes.

	2015			2016		
	Pos.	Neu.	Neg.	Pos.	Neu.	Neg.
Training	72.43	3.20	24.36	70.2	3.80	26.0
Test	53.72	5.32	40.96	74.3	4.90	20.80

To preprocess the dataset, which is in XML format, similar procedures as in the work of [39] are utilized. Most importantly, we delete those observations where the target is implicit (equal to `NULL`). `NULL` values represent less than 10% of all observations and therefore represent a comparatively rare category. Additionally, according to [20], such implicit targets appear quite infrequently in general.

3.2 Data insights

Besides the distribution of reviews among sentiment classes, it is also of high importance to understand the distribution among aspect categories. The distributions for the training and test data are provided in Table 2. The first four dominant categories were selected for the evaluation of the results in this paper. With data from each of these categories, we will also train an aspect-specific QuaNet model, called AspEntQuaNet. Combined, the aforementioned categories represent approximately 80% of the training data and around 78% of the test data. Other categories were taken out of consideration for testing purposes for the reason that on their own, these categories do not represent a large enough part of the data to draw meaningful conclusions.

For the four categories under consideration, we also provide the distribution of review sentences among sentiment classes (positive, neutral, or negative). In Table 3, we see the distribution of sentiment classes for the observations in the training and test data. All four distributions are substantially different from each other, even though they have similar underlying features (e.g., neutral sentiment is always the least represented).

Table 2: Distribution in percentages of review sentences among aspect categories.

Aspect Category	Training	Test
FOOD#QUALITY	40.77	43.54
SERVICE#GENERAL	17.24	16.46
AMBIENCE#GENERAL	12.03	9.08
RESTAURANT#GENERAL	9.85	8.92
FOOD#STYLE_OPTIONS	6.12	7.85
FOOD#PRICES	3.73	3.38
RESTAURANT#MISCELLANEOUS	2.61	2.77
DRINKS#QUALITY	2.34	3.38
DRINKS#STYLE_OPTIONS	1.7	1.69
RESTAURANT#PRICES	1.38	0.77
LOCATION#GENERAL	1.17	1.54
DRINKS#PRICES	1.06	0.62

Table 3: Distribution in percentages of sentiment classes among the chosen aspect categories.

	Training			Test		
Aspect Category	Pos.	Neu.	Neg.	Pos.	Neu.	Neg.
FOOD#QUALITY	73.99	3.53	22.48	85.51	4.24	10.25
SERVICE#G.[1]	56.48	2.47	41.05	54.21	2.80	42.99
AMBIENCE#G.[1]	78.07	5.70	16.23	91.53	5.08	3.39
RESTAURANT#G.[1]	78.69	1.64	19.67	77.59	0.00	22.41

[1]G. denotes the GENERAL subcategory.

Table 3 shows that for some aspect categories, distributions between training and test data are quite similar. For SERVICE#GENERAL, we find a total absolute difference (TAD, the sum of absolute differences in the true prevalences of each sentiment class between training and test data) equal to 4.54%; for RESTAURANT#GENERAL, we find a TAD of 5.48%. However, this is not the case for the other two aspect categories. Namely, for FOOD#QUALITY, we obtain a TAD of 24.46%, and for AMBIENCE#GENERAL, there we find a TAD equal to 26.92%. The differences in these distributions for the training and test data are not insignificant. Therefore, it will be particularly interesting to observe the performance of the chosen methods on the test data for these categories, once trained to optimality.

4 Methodology

This section provides an extensive explanation of the methods researched and proposed in this paper. First, Section 4.1 describes the classification methods and corresponding loss functions. Following, Section 4.2 discusses the quantification methods that are built on classification algorithms.

4.1 Classification Methods

This section provides the necessary methodology for the classification methods proposed in this paper, as well as describes how to move from classification to quantification. First, Section 4.1.1 discusses the LCR-Rot-hop++ neural network architecture designed for ABSA. Then, Section 4.1.2 goes into detail about which loss functions should be applied for quantification purposes.

4.1.1 LCR-Rot-hop++ Neural Network

To utilize any of the quantification methods that are investigated in this paper, we first need to incorporate a classifier. For this purpose, we will utilize the deep learning component of a state-of-the-art Hybrid Approach for Aspect-Based Sentiment Analysis (HAABSA) which was first developed by [40] and further improved by [18] (under the name of HAABSA++).

It has to be noted that considering the fact that for QuaNet we will need to obtain document embeddings and sort them in the order of increasing probabilities (as will be discussed in detail in Section 4.2.6), we can only use the second part of HAABSA++ for this method (LCR-Rot-hop++). Therefore, to ensure a fair comparison with the other methods, we will omit the ontology part completely.

LSTM with attention has become a standard practice and starting point for the ABSA task [39], [41]. LCR-Rot is a model proposed by [42] with the intention of allowing the opinionated expressions to interchange information with their contexts on both sides of the target. In the research work of [40], this model is further extended under the name of LCR-Rot-hop by adding repeated attention.

[18] extends this model even further under the name of LCR-Rot-hop++ by experimenting with various types of word embeddings and adding hierarchical attention. In fact, the largest disadvantage of LCR-Rot-hop is that for the computation of target2context and context2target vectors only the local information is used. Hierarchical attention is meant to solve exactly this issue. By design, it is supposed to represent the input sentence on a higher abstraction level.

Additionally, we have to notice that in the research work of [18], there are four ways of implementing hierarchical attention. As an outcome of various experiments, Method 4 (in which the weighting of attention procedure is employed one by one at each iteration of the rotatory attention mechanism for the pairs of intermediate context and target vectors) was determined to perform the best. For that reason, it will be the only method utilized in this research work.

Multiple components of LCR-Rot-hop++ will be needed in this paper. The first and most important is the last layer of this model. This layer should be extracted and used as an input for one of the quantification methods, as it can be considered a representation (embedding) of a document (opinion with respect to each aspect target). In fact, this layer consists of the left and right representations, while each of those consists of two vectors (in the case of the right part: r^r and r^{t_r}; the dimensions of both are 1x600). In turn, each of the vectors is built upon bi-directional LSTMs which means that their size is doubled (initially the size is 1x300). In the end, we end up with 1x2400 vectors which are to be used as input for quantification methods.

Next to that, after experiments with word embeddings by [18], their research found BERT word embeddings to perform the best. BERT word embeddings are a type of contextual word embeddings [43]. Since we seek to identify the possible best model and build upon this in terms of estimating prevalences, only the BERT word embedding will be used for the purposes of this paper.

4.1.2 Loss Function for Quantifiers

As previously argued in Section 2.1, loss functions that are commonly used for SA are not always suitable for SQ. In fact, until recently there has not been extensive research about the ideal evaluation measure for SQ. [26] proposed a collection of 8 properties (some of them mutually exclusive) which would need to hold in order for the evaluation measure to be suitable. During extensive analysis, the authors determine that no single measure is in line with all the proposed properties. However, Absolute Error (AE) and Relative Absolute Error (RAE) were found to meet the most requirements (6 out of 8).

Interestingly, AE and RAE are considered the simplest evaluation measures for quantification. The idea behind AE is to calculate the absolute difference between the true and predicted quantification. RAE extends AE by expressing the difference between the true and predicted quantification in terms of true quantification. The formulas for these error measures are provided in (2) and (3).

$$AE(p,\hat{p}) = \frac{1}{|C|}\sum_{c\in C}|\hat{p}(c) - p(c)|, \tag{2}$$

$$RAE(p,\hat{p}) = \frac{1}{|C|}\sum_{c\in C}\frac{|\hat{p}(c) - p(c)|}{p(c)}, \tag{3}$$

where C is the set of all sentiment classes.

It should be noted that RAE is undefined in case the true quantification is equal to 0. For this reason, [26] proposes the use of a certain type of additive smoothing:

$$p_s(c) = \frac{|\epsilon + p(c)|}{\epsilon|C| + \sum_{c\in C} p(c)}, \tag{4}$$

where $\epsilon = \frac{1}{2|\sigma|}$ and σ is the set of opinions. However, even with such smoothing, RAE may remain numerically unstable in case of extreme true or predicted prevalences [35].

Another commonly used evaluation measure is Kullback-Leibler Divergence (KLD) [35], [44], [45]. While developed much earlier, KLD was first used for SQ only in [28]. It is computed using (5):

$$KLD(p,\hat{p}) = \sum_{c\in C} p(c)\log\frac{p(c)}{\hat{p}(c)}. \tag{5}$$

The ideal scenario is to have $KLD = 0$. In that case, both distributions (predicted and original) coincide. Similar to RAE, KLD is undefined when $\hat{p}(c) = 0$; we use the same smoothing technique as shown in (4), although this could also still result in numerical instability of the smoothed value.

4.2 Quantification Methods

This section elaborates on each of the five various quantification methods reviewed in this paper. Section 4.2.1 discusses the method "Classify and Count" followed by the explanation of "Adjusted Classify and Count" in Section 4.2.2. Further, Section 4.2.3 presents the adaptation of these methods for the case of three sentiment classes. Next, Section 4.2.4 discusses the probabilistic version of those methods. It is followed by the adaptation of these methods for three sentiment classes in Section 4.2.5. Section 4.2.6 describes the methodology behind QuaNet. Last, Section 4.2.7 provides the adapted QuaNet for three sentiment categories and with entropy-based sorting, named EntQuaNet.

4.2.1 Classify and Count

Considered to be the simplest method for the SQ task, the idea of "Classify and Count (CC)" is to classify each data point separately and then compute the share of population that is estimated to be of a certain sentiment. This can be achieved using the formula below:

$$\hat{p}_c^{CC}(D) = \frac{|\{x\in D|h(x) = c\}|}{|D|} = \frac{TP_h^c + FP_h^c}{|D|}, \tag{6}$$

where $h(\cdot)$ is the hard classification function, subscript h denotes hard predictions, and c is a sentiment class.

Essentially, this can be viewed as sentiment classification with further aggregation. It is obvious that the perfect classifier is also the perfect quantifier, but the opposite is not always the case.

Although we have used three sentiment classes in this paper instead of two, this does not form a problem for CC. The only difference is that the formula, written out, is slightly more extensive:

$$\begin{aligned}\hat{p}_1^{CC3}(D) &= \frac{|\{x\in D|h(x) = 1\}|}{|D|} = \frac{T1_h + F1_h}{|D|}\\ &= \frac{T1_h + F1/2_h + F1/3_h}{|D|}.\end{aligned} \tag{7}$$

In the above equation, $T1_h$ is the number of correctly predicted observations of the first category, $F1/2_h$ is the number of predicted first category observations when the true category is the second one, $F1/3_h$ is the number of predicted first category observations when the true category is the third one. Furthermore, in (7) we denote the sentiment classes positive, neutral, and negative by 1, 2, and 3, respectively. Without loss of generality, the formula has been written out for the first category.

4.2.2 Adjusted Classify and Count

Considering the formula of CC in (6), it can be seen that it gives optimal estimates when $FP = FN$. Obviously, this is rarely the case, which is why an adjusted method was proposed in [23], [28]. The concept behind "Adjusted Classify and Count (ACC)" is slightly more intricate, as it involves adjusting for the number of false positives versus false negatives if one or the other prevails. In fact, provided the "true positive rate" and "false positive rate" are known, we could adjust the share to obtain the optimal result. The idea can be seen in the following equation:

$$\hat{p}_c^{ACC}(D) = \frac{\hat{p}_c^{CC}(D) - \widehat{fpr}_h}{\widehat{tpr}_h - \widehat{fpr}_h}, \tag{8}$$

where $\widehat{tpr}_h = \frac{TP_h}{TP_h+FN_h}$ and $\widehat{fpr}_h = \frac{FP_h}{FP_h+TN_h}$, and we have omitted superscript c due to verbosity. The derivation for this formula is provided in Appendix A. Essentially, it shows that in the case of known true positive and false positive rates (which in reality are estimated from the training data), we can obtain the true prevalence of a certain class.

Considering that the true rates $\widehat{tpr}_h$ and $\widehat{fpr}_h$ are unknown in a real-life scenario, we need to estimate them from our data. According to [34], in ACC, one should estimate these rates from the training data via the process of k-fold cross-validation, or by using a hold-out validation dataset.

A potential problem with ACC is that it might return values that are outside the [0, 1] range. This happens because $\widehat{tpr}_h$ and $\widehat{fpr}_h$ are not always optimal for the test data. For this reason, [23] introduced a clipping procedure that adjusts values higher than 1 to 1, and values lower than 0 to 0.

4.2.3 Adjusted Classify and Count for Three Sentiment Classes

For the purposes of our paper, the formula of ACC given in (8) should again be generalized to the case of three sentiment classes. [46] and [25] have previously provided the general equations with which it is theoretically possible to derive the formulas of adjusted classifiers for any number of sentiment classes. However, to the best of our knowledge, the exact formulas for the case of three classes were not provided in any of the previous research work. Therefore, the formula that would be optimal for the purposes of this paper should be written out in order to be used for the case of ternary sentiment classification.

The idea of obtaining this formula is analogous to that for ACC. The starting point is the following two equations:

$$P(\hat{1}) = P(\hat{1}|1) \cdot P(1) + P(\hat{1}|2) \cdot P(2) + P(\hat{1}|3) \cdot P(3), \quad (9)$$

$$P(\hat{2}) = P(\hat{2}|1) \cdot P(1) + P(\hat{2}|2) \cdot P(2) + P(\hat{2}|3) \cdot P(3). \quad (10)$$

After rewriting these equations we obtain the following three formulas that represent the adjusted prevalence estimations:

$$P(1) = \frac{P(\hat{1}) - \dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (P(\hat{2}) - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}} - \widehat{f_{1/3}r}}{\widehat{t_1r} - \dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (\widehat{f_{2/1}r} - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}} - \widehat{f_{1/3}r}}, \quad (11)$$

$$P(2) = \frac{P(\hat{2}) - \dfrac{(\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot (P(\hat{1}) - \widehat{f_{1/3}r})}{\widehat{t_1r} - \widehat{f_{1/3}r}} - \widehat{f_{2/3}r}}{\widehat{t_2r} - \dfrac{(\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot (\widehat{f_{1/2}r} - \widehat{f_{1/3}r})}{\widehat{t_1r} - \widehat{f_{1/3}r}} - \widehat{f_{2/3}r}}, \quad (12)$$

$$P(3) = 1 - P(1) - P(2), \quad (13)$$

where $\widehat{t_1r} = \frac{T1_h}{T1_h+F2/1_h+F3/1_h}$, $\widehat{f_{2/1}r} = \frac{F2/1_h}{T1_h+F2/1_h+F3/1_h}$, $\widehat{f_{3/1}r} = \frac{F3/1_h}{T1_h+F2/1_h+F3/1_h}$, $\widehat{t_2r} = \frac{T2_h}{F1/2_h+T2_h+F3/2_h}$, $\widehat{f_{1/2}r} = \frac{F1/2_h}{F1/2_h+T2_h+F3/2_h}$, $\widehat{f_{3/2}r} = \frac{F3/2_h}{F1/2_h+T2_h+F3/2_h}$, $\widehat{t_3r} = \frac{T3_h}{F1/3_h+F2/3_h+T3_h}$, $\widehat{f_{1/3}r} = \frac{F1/3_h}{F1/3_h+F2/3_h+T3_h}$, $\widehat{f_{2/3}r} = \frac{F2/3_h}{F1/3_h+F2/3_h+T3_h}$. The letter r in these formulas stands for 'rate'. Derivations for these formulas are provided in Appendix B.

For the scenario of three sentiment classes, there still exists a problem of the output values being outside the [0, 1] range. However, in this case, simple clipping as proposed by [23] does not work since multiple numbers may be above one or below zero. For this reason, we propose a two-step clipping algorithm. Initially, if the lowest quantification estimate is lower than zero, it is made zero and then subtracted from the other two quantification estimates. And at the second step, each estimated prevalence is divided by the obtained sum of prevalences (meaning their sum is rescaled to one).

4.2.4 Probabilistic CC and ACC

CC and ACC use binary classification models for generating predicted values. The majority of such classifiers (as is also the case with LCR-Rot-hop++) are also able to output the values in terms of posterior probabilities $Pr(c|x)$. Additionally, considering that posterior probabilities represent a confidence level of certain classification outcome, which potentially represents more information, it might be beneficial to create probabilistic versions of CC and ACC [35]. This can be done via substituting counts TP_b, TN_b, FP_b, FN_b by their soft counts $TP_s = \sum_{x \in c,D} Pr(c|x)$, $TN_s = \sum_{x \in \bar{c},D} 1 - Pr(c|x)$, $FP_s = \sum_{x \in \bar{c},D} Pr(c|x)$, $FN_s = \sum_{x \in c,D} 1 - Pr(c|x)$.

Using all of the above, "Probabilistic Classify and Count" (PCC) can be defined as follows:

$$\hat{p}_c^{PCC}(D) = \frac{\sum_{x \in D} Pr(c|x)}{|D|} = \frac{TP_s + FP_s}{|D|}. \quad (14)$$

Initially, this method was deemed as non-working by [23], [28], while in later research, it was shown to perform moderately well [47].

At the same time, "Probabilistic version of Adjusted Classify and Count" (PACC) has the formula:

$$\hat{p}_c^{PACC}(D) = \frac{\hat{p}_c^{PCC}(D) - \widehat{fpr}_s}{\widehat{tpr}_s - \widehat{fpr}_s}, \quad (15)$$

where $\widehat{tpr}_s = \frac{TP_s}{TP_s+FN_s}$ and $\widehat{fpr}_s = \frac{FP_s}{FP_s+TN_s}$. It was first designed by [30] under the name of "Scaled Probability Average", and has the idea of substituting variables from (8) to their soft versions (using probabilities instead of counts) as described above.

4.2.5 Probabilistic CC and ACC for Three Sentiment Classes

The versions of PCC and PACC for ternary sentiment classification is only different because it has different components. This way, $T1_s = \sum_{x \in 1,D} Pr(1|x)$, $T2_s = \sum_{x \in 2,D} Pr(2|x)$, $T3_s = \sum_{x \in 3,D} Pr(3|x)$, $F1/2_s = \sum_{x \in 2,D} Pr(1|x)$, $F1/3_s = \sum_{x \in 3,D} Pr(1|x)$, $F2/1_s = \sum_{x \in 1,D} Pr(1|x)$, $F2/3_s = \sum_{x \in 3,D} Pr(2|x)$, $F3/1_s = \sum_{x \in 1,D} Pr(3|x)$, $F3/2_s = \sum_{x \in 2,D} Pr(3|x)$.

At the next step, the true and false positive rates are computed as in Section 4.2.3. Then, using the calculated rates, adjusted quantification values are computed using (11), (12), and (13). Additionally, since the problem of estimates going above one or below zero is also typical for PACC, the same clipping procedure is applied as proposed in Section 4.2.3.

4.2.6 QuaNet

The deep learning method for binary quantification prediction was first proposed by [35]. QuaNet requires document embeddings and a predicted prevalence score for one of the sentiment classes in the case of binary data. This input is extended for our proposed EntQuaNet for n-ary quantification, which additionally requires predicted prevalences for all sentiment classes, as well as an entropy score. The architecture of EntQuaNet, based largely on the architecture of QuaNet, is provided in Fig. 1.

For QuaNet, the first step comprises the classification of the given data (e.g., positive or negative) using LCR-Rot-hop++ as discussed in Section 4.1.1. As a result, we obtain the probabilities of each aspect opinion to belong to a certain sentiment class, along with their embeddings (concatenations of r^l, r^{t_l}, r^{t_r}, r^r). Afterwards, these two elements are concatenated into one single vector to obtain $|D|$ pairs of type $[Pr(c|x), \vec{x}]$. Additionally, these pairs are sorted in increasing order of $Pr(c|x)$ (in the binary quantification case, it does not matter which class is chosen for sorting) before being passed to a bi-LSTM layer. According to [35], the sorting step is needed for QuaNet to leverage the ordered sequence of $Pr(c|x)$ and spot the transition point between negative and positive documents.

Concurrently, using the classified data from the initial step, we calculate prevalences $\hat{p}_c^{CC}(D)$, $\hat{p}_c^{ACC}(D)$, $\hat{p}_c^{PCC}(D)$, $\hat{p}_c^{PACC}(D)$, as well as rates $\widehat{tpr}_s$, $\widehat{fpr}_s$, $\widehat{tnr}_s$, $\widehat{fnr}_s$. These eight values are then combined in a single vector before being concatenated to the hidden state of the bi-LSTM. The obtained vector is passed through the second main component of QuaNet (n fully connected layers, with ReLU activation functions). The idea is that QuaNet could adjust the quantification embedding of bi-LSTM with the help of quantification-related statistics obtained using LCR-Rot-hop++. Finally, the quantification embedding is passed through a softmax function to obtain the prevalences $\hat{p}_c^{QuaNet}(D)$.

The reason for choosing these particular quantification-related statistics is that they do not require much computational power. Moreover, extensive ablation experiments were performed by [35], which confirmed the assumption that all inputs in the last step of QuaNet improve its performance.

Considering the EntQuaNet structure in Fig. 1, it should be noted that the QuaNet-based models will be trained separately from LCR-Rot-hop++ in a two-step approach. This two-step approach entails that we will first train the LCR-Rot-hop++ classification model. Then, using the obtained opinion embeddings as well as the predicted probabilities, these are used as training input for the QuaNet part.

4.2.7 QuaNet for Three Sentiment Classes

To adjust the QuaNet architecture for the case of three sentiment classes, some changes were made. In the explanations and in Fig. 1, sentiment classes are denoted 1, 2, and 3, instead of positive, neutral, and negative.

First, we concatenate to the document embeddings all three probabilities (each corresponding to a certain sentiment class) instead of only one. Next to that, we concatenate the value of entropy calculated from the three given probabilities. We employ the following entropy formula:

$$\text{Entropy} = -\sum_{c \in C} \hat{p}(c) \log \hat{p}(c). \tag{16}$$

We then sort the document embedding obtained with LCR-Rot-hop++ from lowest to highest entropy. Essentially, this corresponds to sorting from the easiest to hardest sentiment choices. In the case of low entropy, there is a clear prediction; when entropy is high, the prediction can be considered to be less certain.

In general, we argue that in our case, entropy-based sorting is better at treating minority data categories because it takes into account all probabilities at once. On the other hand, probability-based sorting tends to disregard the minority categories as the classification is initially decided on the majority category. Only when the probabilities of belonging to the majority category are equal (which is in our case a rare event), does probability-based sorting consider the minority categories. It is known that the sentiment of customers' reviews is predominantly positive [48], leading to unbalanced distributions like those found in this paper. As such, we consider entropy-based sorting to be a fitting option for the sentiment quantification task under consideration.

As explained in the previous section, the last hidden state of the bi-LSTM and a more extensive set of 21 statistics (instead of eight for the binary quantification case) are concatenated and passed through the fully connected layers with ReLU activation functions as shown in Fig. 1. Nine of the statistics come directly from the classified data (T1 - prevalence of correctly predicted first category observations, F1/2 - prevalence of wrongly predicted first category observations when the true value is the second category, etc.). The other 12 values are prevalence predictions for each of the three categories generated by the aforementioned quantification methods: CC, PCC, ACC, and PACC. This way, quantification results of the QuaNet-based models are built upon four simpler models. Considering the advancements with the proposed entropy-based sorting and the extension of three sentiment categories, we call this new model EntQuaNet.

Additionally, to make sure EntQuaNet indeed improves upon QuaNet for the ABSQ task in this paper, we train the ordinary QuaNet on the whole training set without including the value of entropy as a separate variable. Because we consider three probabilities instead of two, it is unclear which probability should be chosen for sorting. For the purposes of this paper, it was decided to apply multi-sorting based on the two most common sentiment classes. Hence, it was first sorted on the probability of opinions belonging to the positive class. Then, in case the probabilities of at least two opinions belonging to the positive class were equal, we employed a similar sorting based on the probability of opinions belonging to the negative class (second-most common). It is pointless to include the last class in the multi-sorting procedure: if the first two probabilities are the same

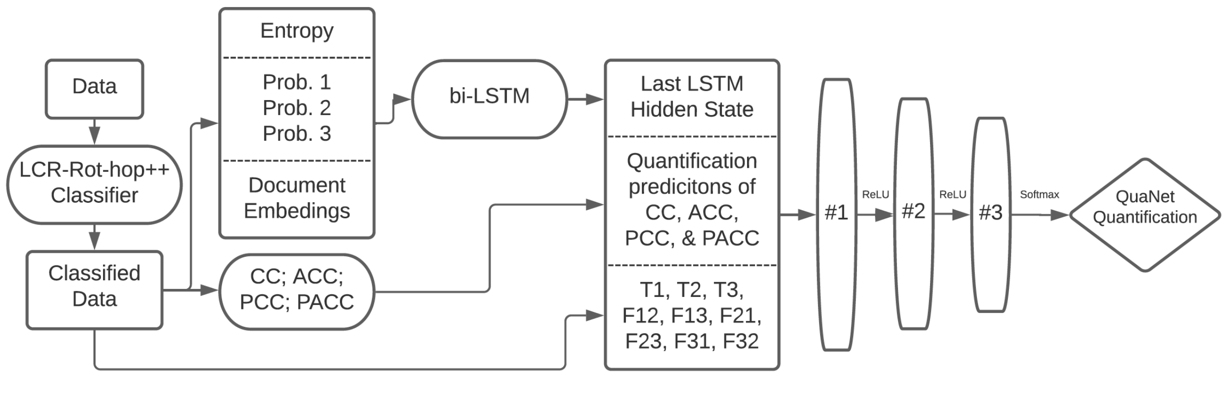


Figure 1: The architecture of the EntQuaNet quantification system for three sentiment classes.

for any two sentiments, the third probability of belonging to the neutral class is equal as well.

To account for the ABSQ task and to see that the aspect restriction of the training sample improves the performance of the quantifiers, we select the four most prevalent aspect categories (FOOD#QUALITY, SERVICE#GENERAL, AMBIENCE#GENERAL, RESTAURANT#GENERAL), and train separate EntQuaNets for each of them. These aspect-specific variants will be referred to as AspEntQuaNet.

In our implementation of various QuaNets for three sentiment classes, LSTM cells contain n hidden dimensions. These dimensions range from 128 to 512; the optimal dimension varies depending on the model and is to be found with a Randomized Grid Search (RGS) procedure. As a result, this architecture gives an output with dimensions $1 \times 2n$, as the utilized LSTM is bidirectional. Afterwards, this output is concatenated with the 21 additional quantification statistics that were discussed above in detail. Essentially, this gives the vector of size $1 \times (2n + 21)$. Subsequently, the resulting vector is passed through three or four dense layers with a varying number of dimensions (also determined via the RGS, and ranging from 64 to 512). Each of these dense layers has a ReLU activation function as well as a 0.3 dropout layer (found to be the most suitable option for the majority of the models via the RGS). At the end, there is a layer of size three with a softmax activation function (each of the output values corresponds to the quantification of a certain sentiment class).

To train the models, it is important to understand how the training and test datasets are created. In our case, for each of the aspect categories as well as the full dataset, we create subdatasets by sampling without replacement the opinions from the total or category datasets. The size of these subdatasets is determined for each of the aspect categories separately. In this paper, we chose this size as approximately 40% of the test dataset size. This percentage was chosen to make sure that the subdatasets are sufficiently different from each other, while at the same time ensuring they are of large enough size for the models to achieve stable test results after training. As such, the subdatasets for FOOD#QUALITY comprise 150 opinions; for SERVICE#GENERAL, 60 opinions; for AMBIENCE#GENERAL and RESTAURANT#GENERAL, 30 opinions. For each of these subdatasets, we calculate the corresponding quantifications of size 1×3.

The optimal number of subdatasets within the training datasets, for each of the aspects and models, was found using an RGS procedure. For this reason, depending on the specification, the number of subdatasets inside the training dataset may range from 1700 to 10000. For testing, we have opted to determine the performance over a constant set of 100 randomly generated subdatasets. This allows us to obtain more accurate and stable performance, as well as fairer comparison between the models.

Additionally, taking into account that the RAE and KLD may become numerically unstable in case the true prevalence of one of the categories is equal to zero, we make sure there is at least one opinion from every sentiment class in each of the subdatasets. However, this could not be done for the test dataset of the RESTAURANT#GENERAL aspect category, since the true prevalence of neutral sentiments is zero.

5 RESULTS

In this section, we discuss and compare results obtained from all previously mentioned ABSQ models and methods. Section 5.1 provides the training process and optimal hyperparameter configurations of the QuaNet models. Following, Section 5 and 5.3, respectively, show the quantification results for the largest aspect category (FOOD#QUALITY) and for the remaining smaller aspect categories (SERVICE#GENERAL, AMBIENCE#GENERAL, and RESTAURANT#GENERAL).

5.1 Hyperparameter Optimization

The performance of machine learning models depends greatly on the configuration of its hyperparameters. Hence, to optimize performance, we employ an RGS to find the optimal configuration of hyperparameters. RGS has been proven to generate models of at least the same quality as those discovered by Grid Search in a fraction of the time [49].

For a fair comparison of results, we compared the models from RGS on a validation dataset comprising 20% of the training dataset from each of the aspect categories. Considering the characteristics for each of the models, displayed in Table 4, we have identified the generally best-performing optimizer for this problem to be Adam and the best-performing loss function to be Mean Absolute Error (MAE), calculated over the constant number of test subdatasets. Additionally, for most of the models, a range of four to seven training epochs was found to be optimal. More training epochs were required to reach optimality in the case of larger datasets.

The best results for AspEntQuaNet were achieved after 16 epochs in terms of the MAE, and 19 epochs when assessed on KLD. After these points, training set losses continue to decrease while losses for the validation set start to increase. Such overfitting behavior is common for deep learning models and shows the importance of finding the right tradeoff between in-sample and out-of-sample performance. We have selected 16 training epochs for this model (MAE was the best loss function for the QuaNet-based models).

For example, AspEntQuaNet was trained specifically on the FOOD#QUALITY aspect category, on 5000 subdatasets of the training dataset. Inside the model were four stacked dense layers of dimensions 512-256-128-64 (in said order), and it was trained using an Adam optimizer with a batch size of 32, and the MAE loss function. Losses were evaluated and averaged over 100 random test samples.

Table 5 shows the optimal configurations of hyperparameters for each of the considered models. For the majority of these models, the optimal batch size trained on our data was determined to be 32. Also in the majority of the cases, four dense layers were superior to three. Additionally, as already mentioned above, smaller models required significantly

Table 4: Set of hyperparameters used in the Grid Search.

Characteristic	Domain Set
Optimizer	Adam, RmsProp, SGD
Loss Function	Mean Absolute Error, KLD
#Epochs	up to 25
#Subdatasets	1700, 2200, 3000, 4000, 5000, 7000, 10000
#Dense Layers	3, 4
Dense layer dimensions	512-256-128-64, 512-256-128, 256-128-64
Batch size	8, 16, 32, 64

Table 5: Optimal configuration of hyperparameters for each of the considered models.

Aspect Category	Model	#Epochs	#Subdatasets	#Layers	Layer dimensions	Batch size
FOOD#QUALITY	QuaNet	11	5000	3	512-256-128	32
	EntQuaNet	10	5000	4	512-256-128-64	32
	AspEntQuaNet	16	5000	4	512-256-128-64	32
SERVICE#GENERAL	QuaNet	5	5000	4	512-256-128-64	32
	EntQuaNet	5	4000	4	512-256-128-64	16
	AspEntQuaNet	7	10000	3	256-128-64	32
AMBIENCE#GENERAL	QuaNet	5	5000	4	512-256-128-64	32
	EntQuaNet	4	5000	4	512-256-128-64	32
	AspEntQuaNet	4	4000	4	512-256-128-64	32
RESTAURANT#GENERAL	QuaNet	4	4000	4	512-256-128-64	32
	EntQuaNet	3	4000	4	512-256-128-64	32
	AspEntQuaNet	3	2200	4	512-256-128-64	16

less computation time and reached optimality in a smaller number of epochs. Lastly, the most common number of subdatasets was either 4000 or 5000, with only two models giving the best results for either 2200 or 10000 subdatasets.

5.2 Results for the FOOD#QUALITY Aspect Category

This section reports results for the largest aspect category by the number of opinions. Table 6 displays results obtained by applying all models to FOOD#QUALITY. The test dataset for this aspect category comprised 283 review opinions.

The first important observation is that ACC and PACC perform the worst when evaluated by the KLD measure, and are worse than ordinary CC in terms of both AE and RAE. This can be explained by the following two arguments. First, consider the training and testing data distribution: in both datasets, there is always a category that contains less than 6% of observations; in fact, in 50% of the cases this number is less than 3%. Second, as discussed previously, ACC and PACC can in theory return probability values that are lower than zero or higher than one (for this reason, [23] proposed a clipping procedure for binary data, which we extended to be usable for ternary data). During the process of obtaining the results it was discovered that, when faced with unbalanced distributions, ACC and PACC may overshoot below zero and above one so much that the clipping procedure changes the quantification estimates completely.

The issue with this clipping procedure is illustrated in the following example. We have randomly selected a subdataset of the test dataset for the FOOD#QUALITY aspect category. First, we have run CC on it and obtained the following predictions for, respectively, positive, negative, and neutral classes: [0.893333, 0.106667, 0]. For ACC, we obtain for this subdataset: [0.978551, 0.11480645, -0.09335745]. Since one of the values is lower than zero, the quantifications are rescaled using the clipping procedure described in Section 4.2.3. This gives us [0.837381, 0.162619, 0] as a quantification output. A similar situation occurs with PACC, where the quantification output is clipped from [1.037080, 0.145764, -0.182844] to [0.787794, 0.212206, 0], changing the output considerably.

This problem is bolstered by the fact that in the majority of the cases, there is not enough training data to estimate all the probabilities and rates mentioned in Sections 4.2.3 and 4.2.5. Hence, especially for the smaller-sized categories, these coefficients may be skewed.

As expected, QuaNet, which was trained on the whole training dataset with a multi-sorting based on probabilities, performed worse than EntQuaNet for which the opinions' sorting was done based on the entropy values calculated using all three probabilities. In fact, there is approximately a 20% difference across all the evaluation measures.

However, it comes as a surprise that both QuaNet and EntQuaNet are significantly worse than CC in terms of AE and RAE. By contrast, when evaluated on KLD, they surpass ordinary CC by around three times. This can be explained by the fact that QuaNet-based models have been shown to be the best at predicting the prevalence of the minority data class. Hence, the value inside the logarithm of the KLD measure (see (5) in Section 4.1.2) does not get as high as when the prediction for the category is equal to zero (which happens often with CC, ACC, and PACC). The fact that KLD punishes more for errors in smaller data categories (ceteris paribus) is its particular feature.

The second-best method for this aspect category was found to be PCC. This is largely explained by the fact that it is much more flexible than CC for the minority classes. In fact, when CC falsely misses the minority category, the probabilistic classifier assigns a (small) probability to it. So, even when CC has predicted zero prevalence for the minority category, its probabilistic counterpart always has a small number assigned to that category, which is most times closer to the true prevalence than a zero prediction.

AspEntQuaNet came out as the best method for this aspect category. In particular, it surpassed the general QuaNet models by roughly three to seven times across all performance measures and outperformed ordinary CC by almost 20 times when evaluated on KLD. These results

Table 6: Results of the quantification method evaluated on the largest aspect category, FOOD#QUALITY (best performances in **bold**).

Model	AE	RAE	KLD
CC	0.036000	0.420337	0.261060
ACC	0.069849	0.754300	0.295884
PCC	0.030851	0.400516	0.043322
PACC	0.062101	0.506329	0.311720
QuaNet	0.114636	0.731996	0.095213
EntQuaNet	0.093901	0.652151	0.072461
AspEntQuaNet	**0.023564**	**0.235014**	**0.013482**

highlight the added value of both entropy-based sorting and training the model on separate aspect categories.

5.3 Results for SERVICE#GENERAL and Remaining Categories

This section discusses the results obtained for the other (smaller) aspect categories. Table 7 displays the results of all researched models evaluated on the SERVICE#GENERAL aspect category, comprising 107 observations in the test data.

In this case, we observe that PACC again underperforms on all three metrics. This time, however, contrary to the case for FOOD#QUALITY, ACC gives better quantification performance than both CC and PACC, although improvements are marginal. In turn, PCC gives prevalence estimations of competitive quality according to the evaluation measures, since it manages to accurately estimate the minority category.

Again, entropy-based sorting generates better results than multisorting on the first two probabilities. This is supported by the fact that EntQuaNet surpasses the performance of normal QuaNet on all three evaluation measures.

For this aspect category, all QuaNet-based models outperform ordinary CC. Additionally, AspEntQuaNet became the best method according to AE and KLD. It was only marginally surpassed by EntQuaNet when evaluated on RAE. Because they are great at predicting the prevalence of the minority category, QuaNet-based models performed particularly well when evaluated on KLD.

Similar results were found for the remaining smaller aspect categories AMBIENCE#GENERAL and RESTAURANT#GENERAL, which comprise 59 and 58 observations in the test data, respectively. For the former category, AspEntQuaNet outperformed the other models on all performance measures. For the latter category, an absence of the neutral sentiment class in the test dataset (see Table 3) resulted in numerical instability for the RAE performance measure. In this case, we found ordinary CC to perform best according to RAE; for the AE and KLD performance measures, AspEntQuaNet proved most effective once more. Additionally, EntQuaNet performed better than normal QuaNet for both categories on almost all performance measures, again highlighting the added benefit of sorting documents on entropy rather than multisorting on the first two probabilities.

6 Conclusion

In this paper, we proposed EntQuaNet, an aspect-based sentiment quantification (ABSQ) method built upon LCR-Rot-hop++, a state-of-the-art classification architecture for the task of aspect-based sentiment analysis. It extends QuaNet, a state-of-the-art method for sentiment quantification, in two ways. It is the first ABSQ method explicitly defined for the quantification of three sentiment classes (positive, neutral, negative), instead of two (positive, negative). Additionally, it uses a novel entropy-based sorting procedure instead of multisorting on probability estimates. Entropy-based sorting was argued to be better at handling minority data categories and resulted in better performance than multisorting for all investigated aspect categories and performance measures.

AspEntQuaNet, the aspect-specific variant of EntQuaNet, consistently outperformed the other methods on almost all evaluation measures. We conclude that AspEntQuaNet can be successfully applied to ternary ABSQ, and often outperforms all existing methods by at least a factor of 2.

Further research could investigate other sorting methods. Currently, input document embeddings are ordered via probability entropy sorting. We would like to investigate other dimensionality reduction techniques such as PCA or the use of autoencoders to represent the three probabilities in one value and sort the document embeddings on that value.

Table 7: Results of the quantification method evaluated on the SERVICE#GENERAL aspect category (best performances in **bold**).

Model	AE	RAE	KLD
CC	0.075444	0.468466	0.267874
ACC	0.065944	1.447930	0.262462
PCC	0.042534	0.254345	0.030829
PACC	0.091526	1.291486	0.094069
QuaNet	0.057297	0.261275	0.031820
EntQuaNet	0.030276	**0.187150**	0.015957
AspEntQuaNet	**0.029821**	0.188882	**0.013353**

References

[1] Y. M. Aye and S. S. Aung, "Sentiment analysis for reviews of restaurants in myanmar text," in *18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2017)*, 2017, pp. 321–326.
[2] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of twitter data for predicting stock market movements," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES 2016)*, 2016, pp. 1345–1350.
[3] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2020.
[4] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *A Practical Guide to Sentiment Analysis*. Springer, 2017.
[5] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey," *IEEE Transactions on Affective Computing*, 2020.
[6] V.-D. Păvăloaia, E.-M. Teodor, D. Fotache, and M. Danileţ, "Opinion mining on social media data: sentiment analysis of user preferences," *Sustainability*, vol. 11, no. 16, p. 4459, 2019.
[7] D. J. Hopkins and G. King, "A method of automated nonparametric content analysis for social science," *American Journal of Political Science*, vol. 54, no. 1, pp. 229–247, 2010.
[8] J. Ramteke, S. Shah, D. Godhia, and A. Shaikh, "Election result prediction using twitter sentiment analysis," in *2016 International Conference on Inventive Computation Technologies (ICICT 2016)*, vol. 1, 2016, pp. 1–5.
[9] S. Baccianella, A. Esuli, and F. Sebastiani, "Using micro-documents for feature selection: The case of ordinal text classification," *Expert Systems with Applications*, vol. 40, no. 11, pp. 4687–4696, 2013.
[10] A. Esuli and F. Sebastiani, "Sentiment quantification," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 72–75, 2010.
[11] G. Brauwers and F. Frasincar, "A survey on aspect-based sentiment classification," *ACM Computing Surveys*, 2022.
[12] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges," *arXiv preprint arXiv:2203.01054*, 2022.
[13] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey," *IEEE Transactions on Affective Computing*, vol. 13, no. 2, pp. 845–863, 2022.
[14] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.
[15] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, 2015.
[16] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[17] M. Tsytsarau and T. Palpanas, "Survey on mining subjective data on the web," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, p. 478–514, 2012.

[18] M. M. Trușcă, D. Wassenberg, F. Frasincar, and R. Dekker, "A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention," in *20th International Conference on Web Engineering (ICWE 2020)*, vol. 12128. Springer, 2020, pp. 365–380.

[19] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowledge-Based Systems*, vol. 235, p. 107643, 2022.

[20] N. Dosoula, R. Griep, R. den Ridder, and R. Slangen, "Detection of multiple implicit features per sentence in consumer review data," in *12th International Baltic Conference on Databases and Information Systems (DB&IS 2016)*, ser. CCIS, vol. 615. Springer, 2016, p. 289.

[21] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: improving rating predictions using review text content." in *12th International Workshop on the Web and Databases (WebDB 2009)*.

[22] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining Text Data*. Springer, 2012, pp. 415–463.

[23] G. Forman, "Quantifying counts and costs via classification," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 164–206, 2008.

[24] W. Gao and F. Sebastiani, "Tweet sentiment: From classification to quantification," in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2015)*. IEEE, 2015, pp. 97–104.

[25] A. Moreo and F. Sebastiani, "Tweet sentiment quantification: An experimental re-evaluation," *arXiv preprint arXiv:2011.08091*, 2020.

[26] F. Sebastiani, "Evaluation measures for quantification: An axiomatic approach," *Information Retrieval Journal*, vol. 23, no. 3, pp. 255–288, 2020.

[27] M. Saerens, P. Latinne, and C. Decaestecker, "Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure," *Neural computation*, vol. 14, no. 1, pp. 21–41, 2002.

[28] G. Forman, "Counting positives accurately despite inaccurate classification," in *16th European Conference on Machine Learning*, ser. LNCS. Springer, 2005, pp. 564–575.

[29] ——, "Quantifying trends accurately despite classifier error and class imbalance," in *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*. ACM, 2006, pp. 157–166.

[30] A. Bella, C. Ferri, J. Hernández-Orallo, and M. J. Ramirez-Quintana, "Quantification via probability estimators," in *10th IEEE International Conference on Data Mining (ICDM 2010)*. IEEE, 2010, pp. 737–742.

[31] L. Milli, A. Monreale, G. Rossetti, F. Giannotti, D. Pedreschi, and F. Sebastiani, "Quantification trees," in *13th International Conference on Data Mining (ICDM 2013)*. IEEE, 2013, pp. 528–536.

[32] G. Da San Martino, W. Gao, and F. Sebastiani, "Ordinal text quantification," in *39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*. ACM, 2016, pp. 937–940.

[33] T. Joachims, "A support vector method for multivariate performance measures," in *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*. ACM, 2005, pp. 377–384.

[34] A. Esuli and F. Sebastiani, "Optimizing text quantifiers for multivariate loss functions," *ACM Transactions on Knowledge Discovery from Data (TKDD 2015)*, vol. 9, no. 4, pp. 1–27, 2015.

[35] A. Esuli, A. Moreo Fernández, and F. Sebastiani, "A recurrent neural network for sentiment quantification," in *27th ACM International Conference on Information and Knowledge Management (CIKM 2018)*. ACM, 2018, pp. 1775–1778.

[36] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq *et al.*, "SemEval-2016 task 5: Aspect based sentiment analysis," in *10th International Workshop on Semantic Evaluation (SemEval 2016)*. ACL, 2016, pp. 19–30.

[37] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, vol. 32, no. 1, 2018, pp. 5876–5883.

[38] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in *9th International Workshop on Semantic Evaluation (SemEval 2015)*. ACM, 2015, pp. 486–495.

[39] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, 2016, pp. 606–615.

[40] O. Wallaart and F. Frasincar, "A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and attentional neural models," in *16th European Semantic Web Conference (ESWC 2019)*, vol. 11503. Springer, 2019, pp. 363–378.

[41] J. Zeng, X. Ma, and K. Zhou, "Enhancing attention-based LSTM with position context for aspect-level sentiment classification," *IEEE Access*, vol. 7, pp. 20 462–20 471, 2019.

[42] S. Zheng and R. Xia, "Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention," *arXiv preprint arXiv:1802.00892*, 2018.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *2019 Annual Conference of the Association for Computational Linguistics: Human Language Technology (NAACL-HLT 2019)*, pp. 4171–4186, 2018.

[44] A. Sanyal, P. Kumar, P. Kar, S. Chawla, and F. Sebastiani, "Optimizing non-decomposable measures with deep networks," *Machine Learning*, vol. 107, no. 8, pp. 1597–1620, 2018.

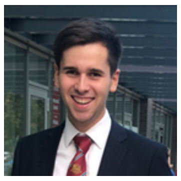
[45] P. González, A. Castaño, N. V. Chawla, and J. J. D. Coz, "A review on quantification learning," *ACM Computing Surveys*, vol. 50, no. 5, pp. 1–40, 2017.

[46] W. Gao and F. Sebastiani, "From classification to quantification in Tweet sentiment analysis," *Social Network Analysis and Mining*, vol. 6, no. 1, pp. 1–22, 2016.

[47] L. Tang, H. Gao, and H. Liu, "Network quantification despite biased labels," in *8th Workshop on Mining and Learning with Graphs (MLG 2010)*. ACM, 2010, pp. 147–154.

[48] N. Hu, J. Zhang, and P. A. Pavlou, "Overcoming the J-shaped distribution of product reviews," *Communications of the ACM*, vol. 52, no. 10, pp. 144–147, 2009.

[49] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 281–305, 2012.



Vladyslav Matsiiako finished both a Bachelor in Econometrics and Operations Research and Economics and Business Economics at Erasmus University Rotterdam, the Netherlands, and a Bachelor in Applied Mathematics at Dnipro State University, Ukraine. Currently, he is a Master student in Operations Research and Information Engineering at Cornell Tech, United States. His research interests include machine learning, deep learning, big data, and business intelligence.

Flavius Frasincar is an assistant professor in information systems at Erasmus University Rotterdam, the Netherlands. He has published in numerous conferences and journals in the areas of databases, Web information systems, personalization, machine learning, and the Semantic Web. He is a member of the editorial boards of the International Journal of Web Engineering and Technology, Information Processing & Management, Decision Support Systems, and Computational Linguistics in the Netherlands Journal, and co-editor-in-chief of the Journal of Web Engineering.

David Boekestijn finished a Bachelor in Econometrics and Operations Research at Erasmus University Rotterdam and currently specializes in Quantitative Finance as an Econometrics and Management Science Master student at the same university. He is also an Artificial Intelligence Master student at the University of Amsterdam. As a research assistant, he focuses on machine learning approaches for text mining.

Appendix A
Derivation for Adjusted Classify and Count

$$
\begin{aligned}
\hat{p}_c^{ACC} &= \frac{\hat{p}_c^{CC}(D) - \widehat{fpr}_h}{\widehat{tpr}_h - \widehat{fpr}_h} = \frac{\dfrac{TP_h + FP_h}{|D|} - \dfrac{FP_h}{FP_h + TN_h}}{\dfrac{TP_h}{TP_h + FN_h} - \dfrac{FP_h}{FP_h + TN_h}} = \frac{\dfrac{TP_h + FP_h}{TP_h + FP_h + TN_h + FN_h} - \dfrac{FP_h}{FP_h + TN_h}}{\dfrac{TP_h}{TP_h + FN_h} - \dfrac{FP_h}{FP_h + TN_h}} \\
&= \frac{\dfrac{TP_h + FP_h}{TP_h + FP_h + TN_h + FN_h} - \dfrac{FP_h}{FP_h + TN_h}}{\dfrac{TP_h \cdot FP_h + TP_h \cdot TN_h - FP_h \cdot TP_h - FP_h \cdot FN_h}{(TP_h + FN_h) \cdot (FP_h + TN_h)}} \\
&= \frac{\frac{TP_h \cdot FP_h + FP_h \cdot FP_h + TP_h \cdot TN_h + FP_h \cdot TN_h - FP_h \cdot TP_h - FP_h \cdot FP_h - TN_h \cdot FP_h - FN_h \cdot FP_h}{(TP_h + FP_h + TN_h + FN_h) \cdot (FP_h + TN_h)}}{\frac{TP_h \cdot TN_h - FP_h \cdot FN_h}{(TP_h + FN_h) \cdot (FP_h + TN_h)}} \\
&= \frac{TP_h \cdot TN_h - FN_h \cdot FP_h}{(TP_h + FP_h + TN_h + FN_h) \cdot (FP_h + TN_h)} \cdot \frac{(TP_h + FN_h) \cdot (FP_h + TN_h)}{TP_h \cdot TN_h - FP_h \cdot FN_h} = \frac{TP_h + FN_h}{TP_h + FP_h + TN_h + FN_h} \\
&= \text{Share of Positives}
\end{aligned}
$$

Appendix B
Derivations for (Probabilistic) Adjusted Classify and Count for Three Sentiment Classes

$$
\begin{aligned}
P(\hat{1}) &= P(\hat{1}|1) \cdot P(1) + P(\hat{1}|2) \cdot P(2) + P(\hat{1}|3) \cdot P(3) \qquad = \widehat{t_1r} \cdot P(1) + \widehat{f_{1/2}r} \cdot P(2) + \widehat{f_{1/3}r} \cdot P(3) \\
&= \widehat{t_1r} \cdot P(1) + \widehat{f_{1/2}r} \cdot P(2) + \widehat{f_{1/3}r} \cdot (1 - P(1) - P(2)) = (\widehat{t_1r} - \widehat{f_{1/3}r}) \cdot P(1) + (\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot P(2) + \widehat{f_{1/3}r} \\
P(1) &= \frac{P(\hat{1}) - (\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot P(2) - \widehat{f_{1/3}r}}{\widehat{t_1r} - \widehat{f_{1/3}r}} \\
P(\hat{2}) &= P(\hat{2}|1) \cdot P(1) + P(\hat{2}|2) \cdot P(2) + P(\hat{2}|3) \cdot P(3) \qquad = \widehat{f_{2/1}r} \cdot P(1) + \widehat{t_2r} \cdot P(2) + \widehat{f_{2/3}r} \cdot P(3) \\
&= \widehat{f_{2/1}r} \cdot P(1) + \widehat{t_2r} \cdot P(2) + \widehat{f_{2/3}r} \cdot (1 - P(1) - P(2)) = (\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot P(1) + (\widehat{t_2r} - \widehat{f_{2/3}r}) \cdot P(2) + \widehat{f_{2/3}r} \\
P(2) &= \frac{P(\hat{2}) - (\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot P(1) - \widehat{f_{2/3}r}}{\widehat{t_2r} - \widehat{f_{2/3}r}} \\
P(1) &= \frac{P(\hat{1}) - (\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot \dfrac{P(\hat{2}) - (\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot P(1) - \widehat{f_{2/3}r}}{\widehat{t_2r} - \widehat{f_{2/3}r}} - \widehat{f_{1/3}r}}{\widehat{t_1r} - \widehat{f_{1/3}r}} \\
&= \frac{P(\hat{1}) - \dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (P(\hat{2}) - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}} + \dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (\widehat{f_{2/1}r} - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}} \cdot P(1) - \widehat{f_{1/3}r}}{\widehat{t_1r} - \widehat{f_{1/3}r}} \\
&= \frac{P(\hat{1}) - \dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (P(\hat{2}) - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}} - \widehat{f_{1/3}r}}{\widehat{t_1r} - \widehat{f_{1/3}r}} + \frac{\dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (\widehat{f_{2/1}r} - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}}}{\widehat{t_1r} - \widehat{f_{1/3}r}} \cdot P(1) \\
P(1) &= \frac{\dfrac{P(\hat{1}) - \dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (P(\hat{2}) - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}} - \widehat{f_{1/3}r}}{\widehat{t_1r} - \widehat{f_{1/3}r}}}{1 - \dfrac{\dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (\widehat{f_{2/1}r} - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}}}{\widehat{t_1r} - \widehat{f_{1/3}r}}} = \frac{P(\hat{1}) - \dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (P(\hat{2}) - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}} - \widehat{f_{1/3}r}}{\widehat{t_1r} - \dfrac{(\widehat{f_{1/2}r} - \widehat{f_{1/3}r}) \cdot (\widehat{f_{2/1}r} - \widehat{f_{2/3}r})}{\widehat{t_2r} - \widehat{f_{2/3}r}} - \widehat{f_{1/3}r}}
\end{aligned}
$$

(continued on next page)

$$P(2) = \frac{\dfrac{P(\hat{2}) - \dfrac{(\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot (P(\hat{1}) - \widehat{f_{1/3}r})}{\widehat{t_1 r} - \widehat{f_{1/3}r}} - \widehat{f_{2/3}r}}{\widehat{t_1 r} - \widehat{f_{1/3}r}}}{1 - \dfrac{\dfrac{(\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot (\widehat{f_{1/2}r} - \widehat{f_{1/3}r})}{\widehat{t_1 r} - \widehat{f_{1/3}r}}}{\widehat{t_2 r} - \widehat{f_{2/3}r}}} = \frac{P(\hat{2}) - \dfrac{(\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot (P(\hat{1}) - \widehat{f_{1/3}r})}{\widehat{t_1 r} - \widehat{f_{1/3}r}} - \widehat{f_{2/3}r}}{\widehat{t_2 r} - \dfrac{(\widehat{f_{2/1}r} - \widehat{f_{2/3}r}) \cdot (\widehat{f_{1/2}r} - \widehat{f_{1/3}r})}{\widehat{t_1 r} - \widehat{f_{1/3}r}} - \widehat{f_{2/3}r}}$$

$$P(3) = 1 - P(1) - P(2)$$