Aspect Based Categorization

A Preprint

Anıl Özdemir
Department of Computer Science
Sabancı University
Istanbul, TR

May 21, 2023

Contents

1 Introduction

1.1 Aspect Based Polarity Analysis

Choosing convenient hotel based on client's need and affordability is a complex decision-making operation. With the widespread use of the Internet in every different industry and field in the world, the number of comments made by users about the services they use is increasing day by day on the internet.

1.2 Reputation System

Numerous studies have been conducted to improve the trustworthiness of online shopping malls by detecting abusers who have participated in the rating system for the sole purpose of manipulating the information provided to potential buyers (e.g., reputations of sellers and recommended items). Especially in the fields of multiagent and recommendation systems, various strategies have been proposed to handle abusers who attack the vulnerability of the system.

The reputation of a product plays an important role as a guide for potential buyers and significantly influences consumers' final purchasing decisions Grazioli and Jarvenpaa ((2000)) , Häubl and Trifts ((2000)) , Agarwal et al. ((2011)).

2 Literature Review

2.1 Text Categorization

In the literature, text classification corresponds to classifying documents under predetermined categories and identification of the offensive language in the particular domains is the sub-task of the text classification. The offensive messages consist of a combination of multiple categories under one roof, and in the past, example of studies have examined each of these subcategories include cyber-aggression, cyberbullying in Kumar et al. ((2018)) and hate speech in Greevy and Smeaton ((2004)). In the days before deep learning became a hot-topic, most of the works contains traditional machine learning methods to solve text classification.

These initial studies began at the beginning of the 21 century and include a wide range of approaches, such as regularized linear classification methods Zhang and Oles ((2001)), kNNs Soucy and Mineau ((2001)), neural networks Ruiz and Srinivasan ((2002)) , Hidden Markov Models Frasconi et al. ((2002)) and more traditional approaches.

During this study, recently applied methods were used rather than traditional methods.These methods mostly focused on CNNs, RNNs, Attention mechanisms, Transformers and Transfer Learning approaches, and therefore, studies involving these methods were utilized.CNN and RNN are undoubtedly very robust and successful models in all sub-branches of NLP, and many examples have been seen in the field of text classification. For instance, Yin et al. ((2017)) draws an analogy between the RNNs and CNNs on an extensive range of NLP tasks, such as textual entailment, relation-sentiment classification, part of speech tagging and a few more tasks. Authors come up with key conclusion that The CNNs and RNNs provide complementary utility for task classification-based problems and the importance level of semantic understanding of the whole text sequences determines the stronger model. Lee and Dernoncourt ((2016)) also confirmed the success of the RNNs and CNNs on the sequential short-text classification by exceeding the state-of-the-art model of that time on a dialog act prediction task. Another approach for text classification proposed by the Kowsari et al. ((2017)) and named as Hierarchical Deep Learning which is different from traditional methods. The Idea is classifying the document in terms of its specialized are, also in its overall field of that document.

The advancements in the NLP field continued with the attention mechanism and the development also had important results in text classification. One of the examples is the,Raffel and Ellis ((2015)) solved the Long-Term Memory problems with the simplistic version of attention that can be applied to feed-forward neural networks. Authors captured the temporal dependencies in the sequences with arbitrary length just as RNNs but without vanishing and exploding gradient problems.

Another example named hierarchical attention network (HAN) that applies the Attention mechanism proposed by the Yang et al. ((2016)). Authors came up with a model that has a hierarchical design that reflects the structure of documents by aggregating the representations of sentences to build representations of documentations

Subsequently, the transformer mechanism was introduced by Google in the famous paper known as "Attention is all you need" Vaswani et al. ((2017)). The Transformer is a model that mainly developed for the solve sequence-to-sequence tasks while capturing long-term relationships and sequence of words in sentences. In the architecture, the authors present a novel layer named "Self-Attention" which is attending the different positions of a single text-sequence to compute a representation of this sequence.Also, the importance of the self-attention on a sentiment analysis task confirmed by the Letarte et al. ((2018)) whose proposed a Self-Attention Network named SANET for text classification with providing more flexibility and interpretability.The self-attention layer provides allowing each word in the given sequence to pay attention to other words that in the same sequences without taking consideration of their position information. On the other hand, pre-trained deep contextualized word representations presented by the Peters et al. ((2018)) called Elmo (Embeddings from Language Models). It was a new type of word representation that models both complex characteristics of word use and how these use changes among the linguistic contexts. It was an advanced version of the traditional Word2vec proposed by the Mikolov et al. ((2013)). Elmo shown large improvements in a broad range of NLP tasks includes sentiment analysis. One of the examples can be shown in the multi-class classification task in the Stanford Sentiment Treebank Hashimoto et al. ((2016)) includes five labels (negative to positive) to describe a sentence from a movie review. Results demonstrate in a 1% absolute accuracy improvement over the state of the art.

Recently, attention-based Transformer Network and RNN variants like the multiplicative LSTM (mLSTM) language models trained on a 40-GB text dataset, then transfer those models to two text classification problems by Kant et al. ((2018)). Results show that approach matches the state of the art on the academic data-set and both Unsupervised pre-training and fine-tuning provides a powerful framework that is efficient in difficult text classification tasks. The Authors also noticed that transformer-based pre-trained models outperform the RNN based ones in the classification task. Afterward, Google researchers proposed a novel model named BERT (Bidirectional Encoder Representations from Transformer) based on the transformer architecture Devlin et al. ((2018)) and the model became a state of the art in a broad range of NLP tasks. One of the examples of BERT in the text classification field is demonstrated by Munikar et al. ((2019)). Authors reported that fine-tuned BERT model for the fine-grained sentiment classification over-performed the recursive, recurrent, and convolutional neural-based networks.

2.2 Sentiment Analysis

2.2.1 Aspect Based Sentiment Analysis

2.3 Sentiment Analysis in Tourism Industry

2.3.1 Aspect Based Sentiment Analysis

2.4 Reputation Systems

With the dramatic rise of digitalization and internet usage in recent years, comments made on products and services constitute a very high decision-making authority on the reputation of these products. One of the biggest challenges of a designing consistent reputation system is that we do not know about the characteristics, background, and behavior of the people who write comments. Reputation systems are started being used in many different domains including hospitality , commercial online applications 2007 , mobile ad-hoc Buchegger and Le Boudec ((2003)) and hospitality Ert and Fleischer ((2019)). Reputations help to lead users against advantageous opinions, and they also ensure a standard to measure ratings in computing overall scores for rated objects, leading to more confident scores. Reputation is actually a two-way concept, this reputation can be both the reviewer and the reputation of an object that the reviewer has commented on.However, but the reputation of reviewer is also referred to as "consistency" in some sources or is directly proportional to the consistency score @NEEDCITATION.

There is also a rapidly growing literature around trust and reputation systems, Chen and Singh ((2001)) proposes general framework for both raters and the objects in rating-based communities. Proposed design computes the quantitative confidence levels on both proficiency of raters and scores of objects while makes use of the method of *transfer of endorsement* Brin and Page ((1998)) which is prominent method that used in some search engines [1998 , 2011]. In the Chen and Singh ((2001)), first the reviewers are grouped under the different ratings they give for each object. The reputation of both the users and objects are then calculated as a result of an equation that includes their endorsement score between different groups and the ratings they give. In overall, proposed system provides the reputations scores for reviewers, the scores of objects with considering reputations of reviewers that has comment on that object and the confidence level of scores.

Oh et al. ((2015)) suffers from existing strategies [2010, 2005 ,2009] in reputation systems generally focus on detecting and eliminating abusers, however it's impossible for abusers to always be accurately identified. Subsequently, the ratings of standard users can be discarded and those of abusers can be included while calculating reputation. Proposed reputation framework evaluates the level of confidence for each rating and adjusts the reputation iteratively based on calculated confidences. The presented system calculates the confidence of the ratings and calculates 3 metrics for each user, namely *user activity*, *user consistency* and *user objectivity*. It calculates the final reputation value for the objects with the calculated confidences.

3 Aspect Based Categorization System

3.1 System Architecture

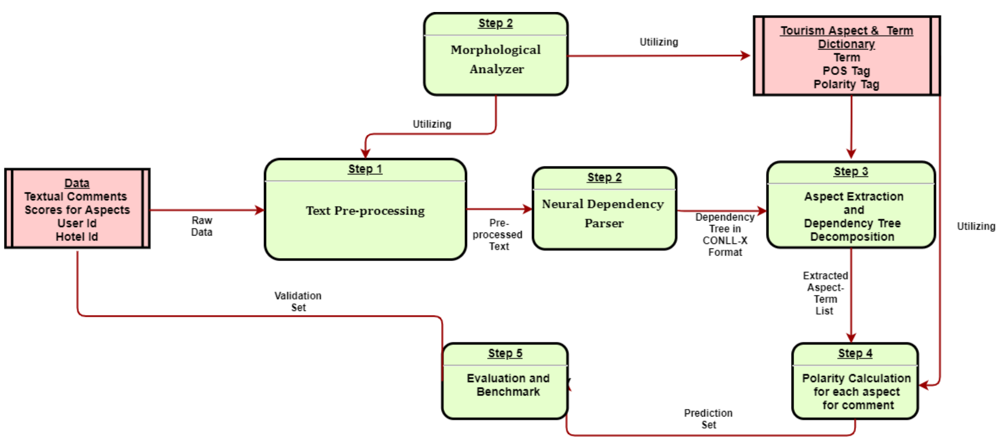

Figure 1: System Overview

3.2 Preliminary

Table 1: Notations

Name Comments	A Description
A	set of unique predefined aspects
R	represents collection of extracted aspects-terms relations
C	set of comments
c	A comment in **C**
r_{ij}	= non-empty set of aspect-term tuples t captured from c

3.2.1 Morphological Parser

One of the most essential steps in any system processing a natural language is morphological analysis. The main purpose of these systems using morphological analyzers is to decompose words into their functional components, applications such as spelling checker, machine translation, parsers can be given as examples of these systems.Morphological analysis is becoming a much more important issue for Turkish because of the fact that Turkish language has an agglutinative morphology structure. Such an structure able to become quite complicated on some of cases due to the large list of affix and abundant inflectional and derivational morphotactics of Turkish Oflazer ((1994)). Such a complex structure brings along the disambigation problem, because morphological parsers in complex languages such as Turkish may give more than one result. At the same time, one of the other consequences of such a complex structure is the problem of disambigation because morphological parsers in complex languages such as Turkish can yield more than one result.

In our system, in order to overcome these difficulties and to perform accurate morphological analysis on textual data we used morphological parser and disambiguation parser which proposed by Sak et al. ((2008)).Proposed Parser receives complete text as an input and adds the meaning of the words into the equation while analyzes and disambiguates the words.Since a word in the Turkish language has more than one morphological analysis output, the meanings of the words in the sentence directly affect these results. In this context, disambiguation is a very crucial step for morphology and the success of disambiguator proposed by Sak et al. ((2008)) has been reported as 96.45% on a disambiguated Turkish corpus.

3.2.2 Dependency Parser

In dependency-based syntactic parsing, the objective is to produce a syntactic structure depicted in Figure 2 and 4 given an input sentence by identifying the syntactic head of each token in the sentence. Such procedure, outputs a dependency graph, where the nodes implies words of the input sentence and the arcs corresponds the binary relations from head to dependent token Nivre et al. ((2007)). The first dependency parser studies started in a rule-based way and Oflazer is the first example in Turkish language and [2011, 2017 , 2011] are given as an examples in other languages.

With the developing technological equipment and the constantly increasing amount of data, the applications of deep learning applications have provided a dramatic increase in many branches in the literature and industry. Deep learning practice has also increased in the field of natural language processing and has been used specifically in dependency parsing systems and has been observed to improve performance [2015,2017]. Recently, a Stanford's LSTM-based dependency parser Dozat et al. was ranked first in 54 treebanks including the IMST-UD Treebank at the CoNLL'17 Shared Task Universal Dependency (UD) Parsing Zeman et al. ((2017)). Subsequently, advanced version of stanford parser proposed by Kanerva et al. presented at the CoNLL'18 Shared Task on UD Parsing Zeman et al. ((2018)) and currently hold the state-of-the-art performances on the dependency parsing of many languages. In this study, open source dependency parser proposed by Kanerva et al. is used due to its satisfying performance.

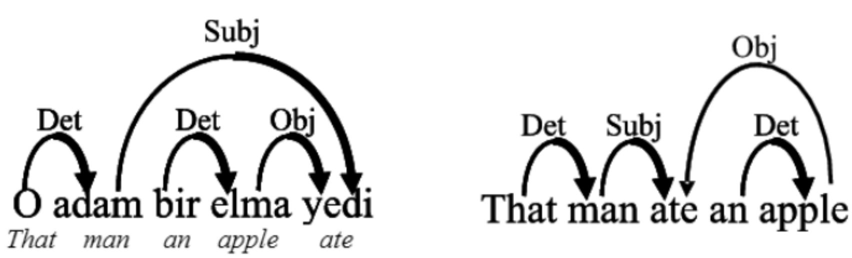

Figure 2: Dependency Relations for a Turkish and English sentence as Eryiğit and Oflazer argues.

Dependency Parsing in Turkish : Turkish is not a rich language in terms of natural language processing studies and this proposal is also valid for dependency parser. The first dependency parser study in Turkish was proposed by Oflazer , another work proposed by Eryiğit and Oflazer which a a word-based and two inflectional group-based first statistical dependency parser for Turkish language. They test their proposed system on 3,398 sentences of the Turkish Dependency Treebank Oflazer et al. ((2003)). Later studies includes , a data-driven dependency parser for Turkish 2008 , graph-based approach 2015.

Data Format : The systems proposed to the ConLL Tasks Zeman et al. ((2018)) take the raw texts as input and constructs dependency tree in CoNLL-U Format[1] which is a revised version of the CoNLL-X format ((2006)) as output. Detailed specification for output format for dependency parsers is the following :

[1]https://universaldependencies.org/format.html

Word lines containing the annotation of a word/token in 10 fields separated by single tab characters and sentences consist of one or more word lines, and word lines contain the following fields:

1. **Id :** Specific token or word index.
2. **Form :** Word form or punctiation symbol.
3. **Lemma :** Stem or lemma (depending on the particular language treebank).
4. **UPOS :** Universal part-of-speeh tag [2].
5. **XPOS :** Language-specific part-of-speech tag.
6. **FEATS :** Set of syntactic and morphological features (depending on the particular language treebank).
7. **HEAD :** Head of the current token, which is either a value of ID, or zero ('0').
8. **DEPREL :** : Dependency relation to the HEAD.
9. **DEPS :** Enhanced dependency graph in the form of a list of head-deprel pairs.
10. **MISC :** Other types of annotations.

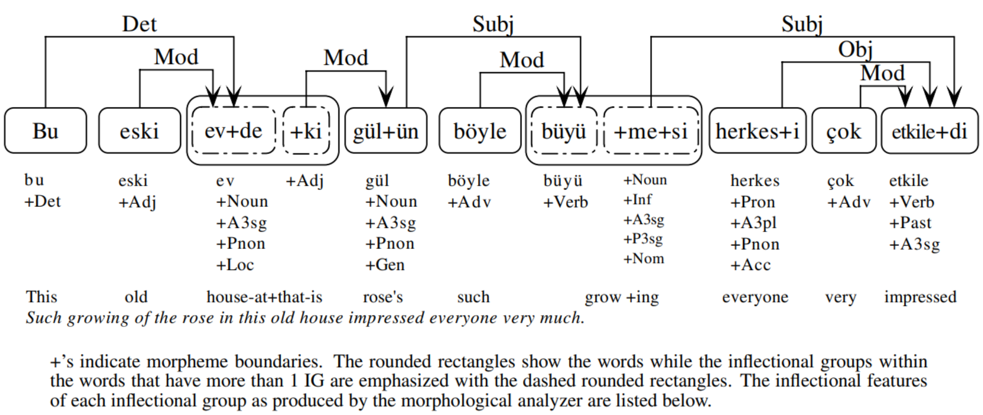

Figure 3: Dependency links in an example Turkish sentence as Eryiğit and Oflazer argues.

3.3 Create Tourism based Lexicon

In our conversations with experts in the field of tourism, we annotated around 450 words within the framework of different tourism categories by manual inspection with. During study, we referred categories as Aspects and extracted set of aspects **A** and examples set of words in that aspects are the following:

- Food - drink, breakfast...
- Hygen - clean , dirty, etc. .
- Room -
- Service -
- Activity -
- Price -
- Location -
- General -
- Unrelated -

The Following three polarity tag are assigned to particular word in the lexicon as a result discussions.

[2]https://universaldependencies.org/u/pos/index.html

- Positive (POS)
- Negative (NEG)
- Neutral (NET)

3.4 Aspect - Term Relation Extraction from Comments

It is not considered an adequate method to evaluate the interpretation made about an object completely in one direction. A comment can contain information about more than one category, which we refer to as an "aspect". Finding out which parts of the comments correspond to which "aspect" is discussed in this study. In other words, being able to extract the aspect-term relation list from the comments. After finding the aspects that a comment contains and the terms that the aspect contains, it is possible to calculate polarity for these aspects. Algorithm 1 , 2 , 3 demonstrates how we extract aspect-term relations from hotel reviews.

Given set of unique pre-defined Aspects **A** and comment c, we define aspect-term tuple t as follows :

$t = (word, AspectPolarity(word))$

Then, define aspect-term relations for single aspect as $r_{ij} \in \mathbf{R}$ where i denotes the unique aspect in **A** and j depends the number of total relations extracted from the dependency tree for that aspect and **R** represents collection of extracted aspects-terms relations.

r_{ij} = non-empty set of aspect-term tuples t captured from c

Example 1 : Given Comment : *otel aileler için uygun değil,gürültülü bir otel. denizi çok temiz,dalgasız bence denizi çok güzel. bardaki personel yetersiz. Yemekleri harika, çalışanlar çok kibar ama aktiviteleri çok az , fiyatına göre uygun , şehir merkezine yakın , odalarda interneti çalışmıyordu ve temiz oteldi..*

Intermediate Output : Intermediating output between preprocessing and post-processing , which result according to the Algorithm 1 .

$r_{11} == r_{food1} == [(temiz, Food, POS)]$
$r_{21} == r_{hygen1} == [(temiz, Hygen, POS), (oteldi, General, NET)]$
$r_{31} == r_{room1} == [(internet, Room, NET), (calismiyor, General, NEG)]$
$r_{41}, r_{42} == r_{swim1}, r_{swim2} == [(dalgasiz, Swim), (deniz, Swim, NET)], [(gzel, General, POS)(deniz, General, NET)]$
$r_{51} == r_{Service1} == [(kibar, General, POS), (calisanlar, Service, NET)]$
$r_{61} == r_{Activity1} == [(az, General, NET), (aktiviteler, Activity, NET)]$
$r_{71} == r_{Price1} == [(uygun, General, POS), (fiyat, Price, NET)]$
$r_{81} == r_{Location1} == [(merkeze, Location, NET), (yakßn, Location, POS)]$
$r_{91} == r_{General1} == [(gurultulu, General, NEG), (otel, General, NET)]$

Post Processed Output : Aspect-Term Relations list for each aspect we defined in Tourism lexicon (see Section 3.3).

- Food : temiz
- Hygen : temiz oteldi
- Room : internet çalışmıyor
- Swim : dalgasiz deniz , güzel deniz
- Service : kibar çalışanlar
- Activity : az aktiviteler
- Price : uygun fiyat
- Location : merkeze yakın değil
- General : gürültülü otel

3.5 Calculating Sentiment from Aspects Terms Relations of Comments

Main objective is the extracting a positivity, negativity and neutralness score for each aspect from the aspects-terms relations outputs.Then use these scores to model the aspect scores of the comments.Basicaly, idea here is to count the polarity tags of the words in the relations. Below you can see how sentiment scores are calculated for single aspect.

$$PolarityScores(c, r_{ij}) = \begin{cases} \sum_j pos^n \in r_{ij} \\ \sum_j neg^n \in r_{ij} \\ \sum_j net^n \in r_{ij} \end{cases} \quad (1)$$

where pos^n, neg^n, net^n denotes to the number of positive, negative and neutral tags in the relations captured through the comment for the corresponding aspect, respectively.

Algorithm 1 Aspect - Term Relation Decomposition

Input: Textual Comment **C**, Tourism Category Lexicon **L**, Empty Set of Dependency Graphs **G**
Output: Non-empty set of Collection of Aspects-Terms **R**
Initialize Empty Set of Colleciton Aspects-Terms **R**
for all sentence $s \in$ **S** in **C** **do**
  Compute and set of Dependency Graphs g (Set of Word Id I, Set of Word W, Set of Lemma L ,Set of Head H, Set of DepRel D);
  Insert g into **G**;
end for
for all g $\in$ **G** **do**
  Initialize Empty Set of Aspect-Term Relation Tuples r_{ij};
  for all i $\in$ I **do**
    if h is not Root **then**
      if u is ADJ or **AspectPolarity**(l) is not "UNK" **then**
        Upward(g,i,r);
      end if
      if u is $NOUN$ and **AspectPolarity**(l) is not "UNK" **then**
        Downward(g,i,r);
      end if
    end if
  end for
  insert r_{ij} into **R**;
end for
adjust **R** and remove duplicate aspects;
return **R**;

Algorithm 2 Upward

Input: Dependency Graph g, Word Id i , Set of Aspect-Term Relation Tuples r
Output: Updated Set of Aspect-Term Relation Tuples r^*
Find Head h, Word w for i in g;
Initialize $i^* \leftarrow h$;
Initialize $r^* \leftarrow r$;
while i^* is not Root **do**
  Find Word w^* and Head h^* for i^*;
  Find new Id i^{**} for h^*;
  Insert Tuple t (w, **AspectPolarity**(w),i) into r;
  Set $i^* \leftarrow h^*$;
end while
return $r*$;

Algorithm 3 Backward

Input: Dependency Graph g, Word Id i , Set of Aspect-Term Relation Tuples r
Output: Updated Set of Aspect-Term Relation Tuples r^*
Compute new Head h^*, new Word w^* , new Id i^* where $h^* \equiv i$
Insert Tuple (w^*,**AspectPolarity**(w^*),i^*) into r;
Return r;

Algorithm 4 AspectPolarity

Input: Word w , Tourism Based Lexicon L
Output: Tourism Domain Category of Input c_w, polarity p_w
Compute category of largest sub-word c_w and polarity p_w in w exist in L;
Return c_w ,p_w;

4 Sentiment Analysis

There are two reasons why we train the Sentiment model. First of all, we want to have a solid working sentiment model. The second is the need for a sentiment model when calculating the text consistency score while calculating the user reputation. Our sentiment model, which has been trained on a huge body of hotel reviews , and written as a micro-service. Thus, the user can choose different models or one of the feature transformation methods including TF-IDF and Bag of words and when using or inferring the model. These models include 5 different machine learning methods including Catboost ,Random Forest , Gradient Boosting Tree , Logistic Regression and SVM. Models are trained using Turkish Sentiment Analysis Dataset Ucan et al. ((2016)).

The procedures we follow and apply while preparing the model for deployment can be seen below.

- Optimize the hyper-parameters of the each models according to the macro-f1 score via cross-validation.
- Train each model using different feature transformations and using the best parameters for the selected model.
- Deploy the model.

5 Cross Reputation System

5.1 User Activity Frequency

In general, users who comment more and submit ratings should be considered as more active users, while those who comment less and submit ratings should be considered less active users.

$$a_u = \Psi(|\mathbf{R}^u|, \alpha, \mu) \tag{2}$$

$$\Psi(|\mathbf{R}^u|, \alpha, \mu) = \frac{1}{1 + e^{-\alpha(|\mathbf{R}^u|, \alpha, \mu)}}$$

(3)

To normalize user activity, we use Ψ function which is the sigmoid function for normalization and described in (3). Ψ function keeps the input value in the range of [0,1] and ensures that value not affected by the outliers. Thus, the users with user activity is 1 are the most active users. In (3), α is a parameter that determine the slope and μ corresponds the intercept of the curve of Ψ. Determining the appropriate values for the α and μ values has a direct effect on the reasonable and even distribution of $|\mathbf{R}^u|$ in the range of [0, 1].We determined these values by making pre-experiments by trial and error method.

5.2 Reputation

Reputation is computed based on ratings adjusted by confidence. The confidence of a rating is calculated based on two scores, user activity and user objectivity, and is then penalized based on its abnormality determined according to user consistency.

5.2.1 User Objectivity

$$o_r = |\frac{r - \overline{r}_m}{s_m}| \tag{4}$$

Where

- o_r: objectivity of rating r for item $m(r) \in \mathbf{R}_m$
- $\overline{r}_m$: mean of $\mathbf{R}_m$, reputation of item m
- s_m: standard deviation of $\mathbf{R}_m$
- $\mathbf{R}_m$: set of ratings on item m , $R_m \in \mathbf{R}$.
- $\mathbf{R}$: set of ratings (all)

If o_r closer to 0, the user is more objective according to the Oh et al. ((2015)).Then overall user objectivity, denoted by o_u, is calculated as the average of the objectivities of the ratings by that user. Definition of o_u as follows:

$$o_u = \frac{1}{\|\mathbf{R}^u\|} \sum_{r \in \mathbf{R}^u} o_r \tag{5}$$

Where

- R^u : Set of ratings by user u, $\mathbf{R}^u \in \mathbf{R}$

5.2.2 Text Consistency Score

Text consistency score was calculated in accordance with the following steps.

- Perform sentiment analysis to decide whether data contains positive or negative content along with a confidence value on a labeled sentiment dataset.
- Calculate the confidence , in other words probability of unlabeled comments being positive or negative using the pre-trained sentiment model. Then assign calculated probability as polarity confidence score (e.g. 99% confidence this comment is positive). Thus, confidence score give a good hint on uncertainties.
- Define sufficient number of segments of comments their polarity confidence scores for both positive and negative comments.
- The text consistency score is the Z-score over the distribution of the actual numerical ratings given by users for those comments, considered as a sample of the comments in the segment that a user is assigned using their polarity confidence score.

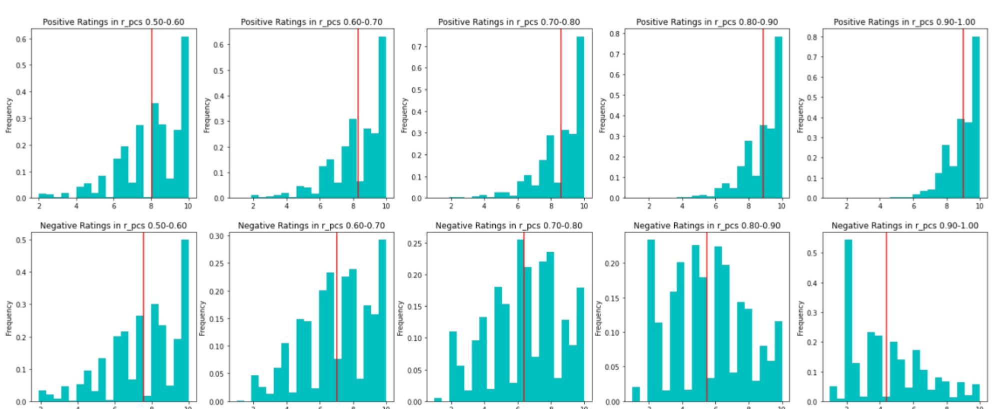

Figure 4: Distribution of Rating Scores according to the different segments determined by polarity confidence score levels

$$t_{cr} =, r \in \mathbf{R}^u \tag{6}$$

5.2.3 Trust Score of Rating

$$t_r = a_u \times o_u \times t_{cr} \tag{7}$$

Where

- t_{cr} : Text consistency score of rating r , where $r \in \mathbf{R}^u$
- t_r : Denotes confidence / trust score of rating r , where $r \in \mathbf{R}^u$
- o_u : Denotes the objectivity of user
- a_u : Activity Score of User

5.2.4 Reputation of User

It is calculated as the average of the trust rate scores of a post that users have commented on and given its rating score.

$$r_u = \overline{T_r}, r \in \mathbf{R}^u \tag{8}$$

Where

- T_r : Set of trust scores of ratings $\mathbf{R}^u$ scored by user.

5.2.5 Reputation of Items

$$R_i = \frac{\sum_{r \in \mathbf{R}_m}(r \times r_u)}{\sum_{r \in \mathbf{R}_m} r} \times a_i \tag{9}$$

Where

- a_i : Normalized activity score of an item.
- r_u : Reputation of the owner of the rating.

6 Experimental Setup Results

6.1 Dataset Description

6.1.1 Turkish Hotel Review Dataset

In order to test the performance of the proposed both aspect based categorization system and cross reputation system 400,000 annotated Turkish comments were collected from the site of hotelpuan.com, which is a special resource. Collected reviews were written for 800 different hotels with special user names and were scored using likert scale (0-10). in the following aspects. Scored aspects and descriptions of associated aspect can be shown below :

- **General Score :** Corresponds to the average of the scores given for the other aspects.
- **Food :** Evaluation of a hotel's food, cafeteria.
- **Room :** About the general comfort and features of the rooms.
- **Swim :** Issues related to the sea, pool and beach.
- **Service :** The general service of the hotel, the attitude of the employees.
- **Location :** The location of the hotel, Relative proximity to the city center.

6.1.2 Turkish Sentiment Analysis Dataset

To test and develop the sentiment model in the tourism domain texts, we used Turkish Sentiment Analysis Dataset Ucan et al. ((2016)) which has 18,478 reviews that extracted from 550 hotels, a balanced set of positive and negative reviews was selected. The average token length of all 11,600 selected positive and negative hotel reviews were 74.

6.2 Results for Aspect Based Categorization

Using the gold-labels in our data, we tried to measure whether the scores given by the users for the aspects could be predicted using the polarity scores we extracted for that aspect.

In the beginning, we had comments and these comments had scores according to aspects (food,room,swim, service and location). Our hypothesis is to estimate the user's score for that aspect with the aspect scores produced by our aspect-based-categorization algorithm. However, instead of estimating the users' scores numerically, we grouped them as positive and negative (positive if 7 and higher than 7, otherwise negative, this limit was determined according to the distribution).Thus, we turned the problem into a classification problem and tried to solve it that way. For each aspect, 4 different machine learning models including bagging-boosting tree algorithms and logistic regression were trained and compared their performance with our baseline scores (Dummy models that labeled all comments as positive or negative). We trained the models by reducing the positive comments to have balanced data and as a result we compared the macro f1 scores of the models with the baseline. The performances of our models, both during the cross validation and on the test data, yields much higher scores than the baselines we determined. Specifically, the Catboost model outperformed other models.

Model	Target Aspect	Training Size	Test Size	Neg / Pos	Macro F1	Acc
Catboost	General	51596	9100	1.14	0.800	0.80
Random Forest	General	51596	9100	1.14	0.790	0.780
Gradient Boosting	General	51596	9100	1.14	0.797	0.796
Logistic Regression	General	51596	9100	1.14	0.780	0.780
ALL POS	General	51596	9100	1.14	0.318	0.467
ALL NEG	General	51596	9100	1.14	0.348	0.533
Catboost	Food	51596	9100	0.592	0.659	0.606
Random Forest	Food	51596	9100	0.592	0.653	0.607
Gradient Boosting	Food	51596	9100	0.592	0.656	0.602
Logistic Regression	Food	51596	9100	0.592	0.64	0.51
ALL POS	Food	51596	9100	0.592	0.386	0.628
ALL NEG	Food	51596	9100	0.592	0.271	0.372
Catboost	Room	51596	9100	0.591	0.66	0.61
Random Forest	Room	51596	9100	0.591	0.651	0.605
Gradient Boosting	Room	51596	9100	0.591	0.656	0.602
Logistic Regression	Room	51596	9100	0.591	0.64	0.51
ALL POS	Room	51596	9100	0.591	0.386	0.628
ALL NEG	Room	51596	9100	0.591	0.271	0.372
Catboost	Swim	51596	9100	0.704	0.688	0.677
Random Forest	Swim	51596	9100	0.704	0.673	0.664
Gradient Boosting	Swim	51596	9100	0.704	0.677	0.667
Logistic Regression	Swim	51596	9100	0.704	0.650	0.600
ALL POS	Swim	51596	9100	0.704	0.370	0.587
ALL NEG	Swim	51596	9100	0.704	0.293	0.413
Catboost	Service	51596	9100	0.632	0.683	0.658
Random Forest	Service	51596	9100	0.632	0.672	0.650
Gradient Boosting	Service	51596	9100	0.632	0.680	0.657
Logistic Regression	Service	51596	9100	0.632	0.650	0.570
ALL POS	Service	51596	9100	0.632	0.380	0.613
ALL NEG	Service	51596	9100	0.632	0.279	0.387
Catboost	Location	51596	9100	0.709	0.705	0.697
Random Forest	Location	51596	9100	0.709	0.704	0.696
Gradient Boosting	Location	51596	9100	0.709	0.704	0.696
Logistic Regression	Location	51596	9100	0.709	0.670	0.630
ALL POS	Location	51596	9100	0.709	0.369	0.585
ALL NEG	Location	51596	9100	0.709	0.293	0.415

Table 2: Evaluation of different Models among Aspects

Model	Target Aspect	CV = 5 Mean F1	CV=5 F1 std	Test Macro F1	Test Acc
Catboost	General	0.79	0.00275	0.800	0.800
Catboost	Food	0.601	0.00322	0.659	0.606
Catboost	Room	0.599	0.00339	0.66	0.61
Catboost	Swim	0.67	0.002	0.688	0.677
Catboost	Service	0.653	0.0016	0.683	0.658
Catboost	Location	0.697	0.003	0.705	0.697

Table 3: Cross Validation Result of Best Models for each Target Aspect

References

A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. J. Passonneau. Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)*, pages 30–38, 2011.

I. Boguslavsky, L. Iomdin, V. Sizov, L. Tsinman, and V. Petrochenkov. Rule-based dependency parser refined by empirical and corpus statistics. In *Proceedings of the International Conference on Dependency Linguistics*, pages 318–327. Citeseer, 2011.

M. Brennan, S. Wrazien, and R. Greenstadt. Using machine learning to augment collaborative filtering of community discussions. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1569–1570, 2010.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

S. Buchegger and J.-Y. Le Boudec. A robust reputation system for mobile ad-hoc networks. Technical report, 2003.

S. Buchholz and E. Marsi. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning (CoNLL-X)*, pages 149–164, 2006.

M. Chen and J. P. Singh. Computing and using reputations for internet ratings. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 154–162, 2001.

P.-A. Chirita, W. Nejdl, and C. Zamfir. Preventing shilling attacks in online recommender systems. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 67–74, 2005.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

T. Dozat, P. Qi, and C. D. Manning. Stanford's graph-based neural dependency parser at the conll 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, 2017.

C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*, 2015.

E. Ert and A. Fleischer. The evolution of trust in airbnb: A case of home rental. *Annals of Tourism Research*, 75: 279–287, 2019.

G. Eryiğit and K. Oflazer. Statistical dependency parsing for turkish. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

G. Eryiğit, J. Nivre, and K. Oflazer. Dependency parsing of turkish. *Computational Linguistics*, 34(3):357–389, 2008.

P. Frasconi, G. Soda, and A. Vullo. Hidden markov models for text categorization in multi-page documents. *Journal of Intelligent Information Systems*, 18(2-3):195–217, 2002.

S. Grazioli and S. L. Jarvenpaa. Perils of internet fraud: An empirical investigation of deception and trust with experienced internet consumers. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(4):395–410, 2000.

E. Greevy and A. F. Smeaton. Classifying racist texts using a support vector machine. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 468–469. ACM, 2004.

K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016.

G. Häubl and V. Trifts. Consumer decision making in online shopping environments: The effects of interactive decision aids. *Marketing science*, 19(1):4–21, 2000.

N. Hurley, Z. Cheng, and M. Zhang. Statistical attack detection. In *Proceedings of the third ACM conference on Recommender systems*, pages 149–156, 2009.

A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

J. Kanerva, F. Ginter, N. Miekka, A. Leino, and T. Salakoski. Turku neural parser pipeline: An end-to-end system for the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 133–142, 2018.

N. Kant, R. Puri, N. Yakovenko, and B. Catanzaro. Practical text classification with large pre-trained language models. *arXiv preprint arXiv:1812.01207*, 2018.

J. M. Kleinberg, M. Newman, A.-L. Barabási, and D. J. Watts. *Authoritative sources in a hyperlinked environment*. Princeton University Press, 2011.

M. Korzeniowski and J. Mazurkiewicz. Rule based dependency parser for polish language. In *International Conference on Artificial Intelligence and Soft Computing*, pages 498–508. Springer, 2017.

K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 364–371. IEEE, 2017.

J. Krivanek and D. Meurers. Comparing rule-based and data-driven dependency parsing of learner language. 2011.

R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics.

J. Y. Lee and F. Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*, 2016.

G. Letarte, F. Paradis, P. Giguère, and F. Laviolette. Importance of self-attention for sentiment analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 267–275, 2018.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

M. Munikar, S. Shakya, and A. Shrestha. Fine-grained sentiment classification using bert. *arXiv preprint arXiv:1910.03474*, 2019.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. The conll 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932, 2007.

K. Oflazer. Two-level description of turkish morphology. *Literary and linguistic computing*, 9(2):137–148, 1994.

K. Oflazer. Dependency parsing with an extended finite-state approach. *Computational Linguistics*, 29(4):515–544, 2003.

K. Oflazer, B. Say, D. Z. Hakkani-Tür, and G. Tür. Building a turkish treebank. In *Treebanks*, pages 261–277. Springer, 2003.

H.-K. Oh, S.-W. Kim, S. Park, and M. Zhou. Can you trust online ratings? a mutual reinforcement model for trustworthy online rating systems. *IEEE Transactions on systems, man, and cybernetics: systems*, 45(12):1564–1576, 2015.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

C. Raffel and D. P. Ellis. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*, 2015.

M. E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1): 87–118, 2002.

H. Sak, T. Güngör, and M. Saraçlar. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *International Conference on Natural Language Processing*, pages 417–427. Springer, 2008.

W. Seeker and Ö. Çetinoğlu. A graph-based lattice dependency parser for joint morphological segmentation and syntactic analysis. *Transactions of the Association for Computational Linguistics*, 3:359–373, 2015.

P. Soucy and G. W. Mineau. A simple knn algorithm for text categorization. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 647–648. IEEE, 2001.

A. Ucan, B. Naderalvojoud, E. A. Sezer, and H. Sever. Sentiwordnet for new language: Automatic translation approach. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 308–315. IEEE, 2016.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

W. Yin, K. Kann, M. Yu, and H. Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

D. Zeman, M. Popel, M. Straka, J. Hajič, J. Nivre, F. Ginter, J. Luotolahti, S. Pyysalo, S. Petrov, M. Potthast, F. Tyers, E. Badmaeva, M. Gokirmak, A. Nedoluzhko, S. Cinková, J. Hajič jr., J. Hlaváčová, V. Kettnerová, Z. Urešová, J. Kanerva, S. Ojala, A. Missilä, C. D. Manning, S. Schuster, S. Reddy, D. Taji, N. Habash, H. Leung, M.-C. de Marneffe, M. Sanguinetti, M. Simi, H. Kanayama, V. de Paiva, K. Droganova, H. Martínez Alonso, Ç. Çöltekin, U. Sulubacak, H. Uszkoreit, V. Macketanz, A. Burchardt, K. Harris, K. Marheinecke, G. Rehm, T. Kayadelen, M. Attia, A. Elkahky, Z. Yu, E. Pitler, S. Lertpradit, M. Mandl, J. Kirchner, H. F. Alcalde, J. Strnadová, E. Banerjee, R. Manurung, A. Stella, A. Shimada, S. Kwak, G. Mendonça, T. Lando, R. Nitisaroj, and J. Li. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-3001. URL https://www.aclweb.org/anthology/K17-3001.

D. Zeman, J. Hajic, M. Popel, M. Potthast, M. Straka, F. Ginter, J. Nivre, and S. Petrov. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21, 2018.

T. Zhang and F. J. Oles. Text categorization based on regularized linear classification methods. *Information retrieval*, 4 (1):5–31, 2001.