

## Article

# Identifying Service Opportunities Based on Outcome-Driven Innovation Framework and Deep Learning: A Case Study of Hotel Service

Sunghyun Nam <sup>1</sup>, Sejun Yoon <sup>2</sup> , Nagarajan Raghavan <sup>3</sup>  and Hyunseok Park <sup>2,\*</sup><sup>1</sup> Samsung Research, Seoul 04763, Korea; shyun.nam@samsung.com<sup>2</sup> Department of Information Systems, Hanyang University, Seoul 04763, Korea; yoonsejun@hanyang.ac.kr<sup>3</sup> Engineering Product Development Pillar, Singapore University of Technology and Design (SUTD), Singapore 487372, Singapore; nagarajan@sutd.edu.sg

\* Correspondence: hp@hanyang.ac.kr; Tel.: +82-2-2220-2396

**Abstract:** This research proposes a data-driven systematic method to discover service opportunities in a specific service sector. Specifically, the method quantitatively identifies the important but unsatisfied customer needs by analyzing online review data. To represent customer needs in a structured form, the *job-to-be-done*-based customer outcomes are adopted from the outcome-driven innovation (ODI) framework. Therefore, *job-to-be-done* information is extracted from the review data and is transformed into customer outcomes. The outcomes having high service opportunities are selected by metrics for quantifying the importance and satisfaction score of the outcomes. This paper conducted an empirical study for hotel service using relevant review data. The results show that the method can identify customer needs in hotel service—e.g., maximizing safety to pay price/deposit, and maximizing possibility to avoid waiting at lobby—and objectively prioritize strategic directions for service innovation. Therefore, the proposed method can be used as an intelligent tool for the effective development of a business strategy.



**Citation:** Nam, S.; Yoon, S.; Raghavan, N.; Park, H. Identifying Service Opportunities Based on Outcome-Driven Innovation Framework and Deep Learning: A Case Study of Hotel Service. *Sustainability* **2021**, *13*, 391. <https://doi.org/10.3390/su13010391>

Received: 20 November 2020

Accepted: 30 December 2020

Published: 4 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** customer needs identification; *job-to-be-done*; service job map; subject-action-object (SAO); natural language processing (NLP); bidirectional encoder representations from transformers (BERT); attention network

## 1. Introduction

New business development is an essential task for sustainable growth in markets [1–3]. Even though customer needs are explicit directions for business development, identifying important but unmet needs is basically a difficult task [4,5]. In fact, even many successful companies often fail to find what their customers really want. To help the task, there has been much effort on a systematic approach to customer needs identification [6,7]. Outcome-driven innovation (ODI) as a *job-to-be-done*-based analytic framework to identify customer hidden needs [8] is one representative approach. ODI represents customer needs as what *outcomes* customers want to achieve from *job-to-be-done*, i.e., *desired outcomes* [9]. *Job-to-be-done* is generally defined as the fundamental purposes or problems that customers want to solve by products or services [9,10] and *outcomes* as a measure of how well the *job-to-be-done* are achieved [9]. In particular, since ODI decomposes the whole operational process of each *job-to-be-done* into multiple universal steps by using a *Universal job map* [5], ODI increases the chances to discover hidden service opportunities.

Regardless of the theoretical and methodological excellences, ODI has critical drawbacks. Since the diversity and robustness of a set of the defined *outcomes* directly affect the quality and reliability of analytic results, defining appropriate *job-to-be-done* and *outcomes* is the crucial task in an ODI-based approach [9]. ODI is basically an expert knowledge-based approach, but it is fundamentally difficult to define the complete set of *job-to-be-done* and *outcomes* by domain experts' subjective knowledge in the recent complex and converging

market environment. In addition, ODI determines some of the *outcomes* as service opportunities by customer interviews, but it is a very time-consuming and expensive approach. To overcome the limitations, this paper proposes an ODI-based quantitative method to identify customer needs in a specific service sector using deep learning techniques. Instead of customer interviews, the method utilizes online review data. Specifically, each customer review is tokenized into sentences, and each sentence is mapped onto one of the steps in *Service job map* by a multiclass text classification using BERT (bidirectional encoder representations from transformers) [11]. *Job-to-be-done* information represented as subject-action-object (SAO) is extracted by a syntactic analysis using a natural language processing (NLP) toolkit, such as Stanford CoreNLP [12] or spaCy [13]. *Job-to-be-done* is transformed into relevant *outcomes* by identifying the relevant measures using BERT-based semantic similarity and combining them to each *job-to-be-done*. *Outcomes* identified in the same step in a *Service job map* are clustered by using the pre-trained BERT-based semantic sentence similarities, and one representative *outcome* for each cluster is defined. The important, but unmet *outcomes*, i.e., customer needs, are identified based on the metrics for quantifying the importance and satisfaction of *outcomes*. As the first attempt to develop a data-driven ODI approach for service sectors, the proposed method has clear contributions. First, this research adopted *Service job map*, instead of *Universal job map*, to cover more complex service processes. Second, the performance of classifying *job-to-be-done* into one of *Service job map* steps is improved by using a BERT-based attention network. Third, the *outcome* statements are automatically generated from the review data. Lastly, the method can assess potential opportunities for the outcomes without any expensive interview process.

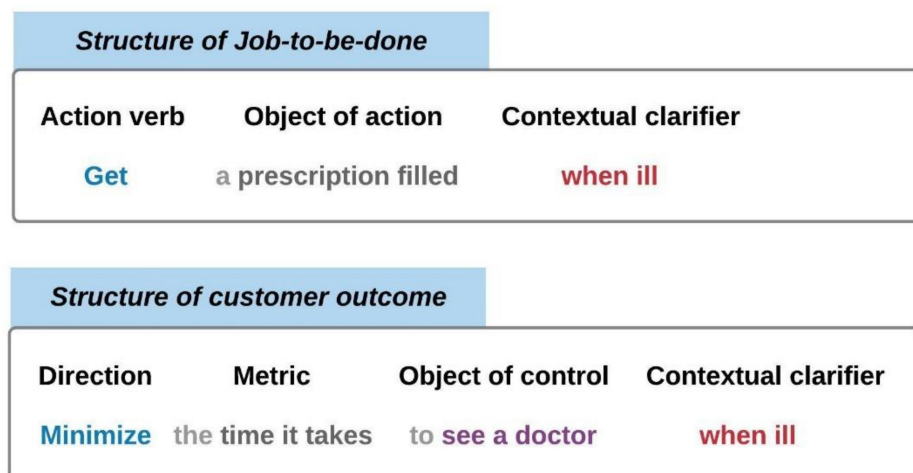
To test the proposed method, this paper conducted an empirical study on hotel service using the relevant online customer review data. The results show that the proposed method can discover customer needs, which are difficult to be identified by conventional approaches, and evaluate the service opportunity of customer needs in an objective manner. Since the result clearly prioritizes strategic directions, the proposed method can be an intelligent tool for service innovation.

## 2. Theoretical Background

### 2.1. Outcome-Driven Innovation

Customers buy products or services to get their *jobs* done [14]. When people buy a quarter-inch drill, what they really want is not a drill, but a quarter-inch hole [5]. In addition, unlike products or services, a *job-to-be-done* not only has no solution boundary but also is stable over time. In the music industry, for example, many products and services from tape players, compact disc players, MP3 players to streaming services have been developed and delivered over time, but their main *job-to-be-done* “listen to music” has not been changed. Therefore, *job-to-be-done* theory has been widely recognized as a framework for categorizing, defining, capturing, and organizing all customer needs [9,10,15], and new product/service development should start with identifying their customers’ fundamental jobs.

ODI is a systematic process to discover hidden segments of opportunities by analyzing from the customer’s *job-to-be-done* perspective [9]. ODI basically focuses on what customers are trying to execute, i.e., *job* when they are using a product or service and what measures of performance, i.e., *outcome*, customers use to evaluate the success during or after they execute it. Figure 1 shows an example of a *job-to-be-done* and *outcome*. The basic format of a *job-to-be-done* is verb-object + context, and the format of *outcome* is direction + metric + object of control, i.e., verb-object + context. In particular, since the structure of *outcome* allows only two directions, maximize or minimize, with measurable metrics, interviewees can give accurate scores without ambiguity.

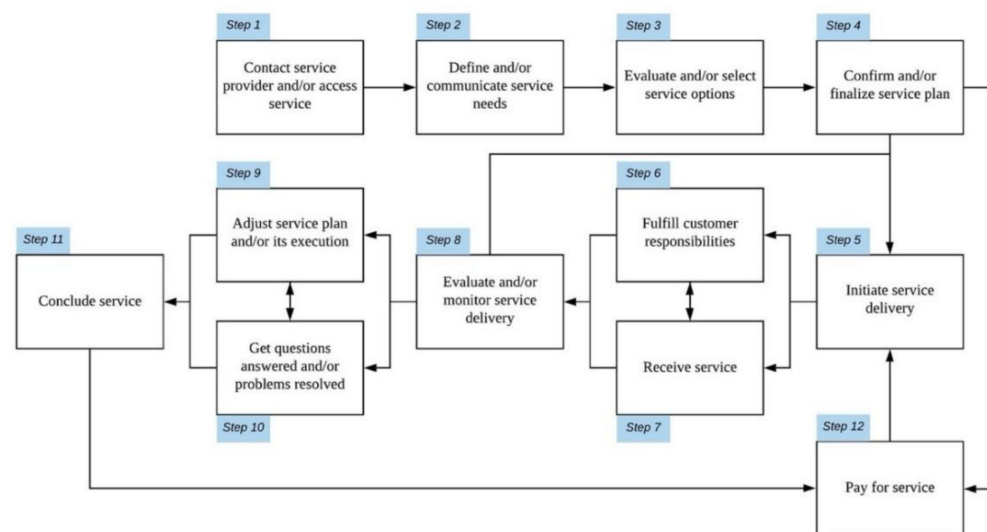


**Figure 1.** Example of *job-to-be-done* and *outcome*. Note: In the medical service process, if a patient, i.e., customer, is ill and visit the hospital, the reason why the patient gives personal information, sees a doctor, and describes symptoms is to get a prescription filled so that the patient can get take proper medicine and gets better. Therefore, the *job-to-be-done* that the customer is trying to achieve is to get a prescription filled. After having the prescription filled, the customer would evaluate the entire experience beginning from visiting the hospital to leaving the hospital. If the customer thinks that the time it takes to see a doctor is important and the time needs to be shortened, the *outcome* that the customer wants from the medical service is to minimize the time it takes to see a doctor. If the hospital wants to innovate their services for increasing customer values, reducing the time it takes to see a doctor would be a good strategic direction.

The strength of ODI as an analytic framework comes from the decomposition of a *job's* operational process. Although most *job-to-be-done*-based approaches only focus on a *job's* execution, there exist many cases that critical service opportunities were identified prior to or after the execution step. Apple's iTunes can be a good example. This successful service is more focused on the convenience in music preparation and organization than the execution of the *job-to-be-done* "listen to music". For this, ODI suggested *Service job map* (Figure 2) [16]. *Service job map* can effectively visualize and overall operational process of a specific *job*, and so it is a basic framework that enables ODI to discover desired outcomes from unexpected steps. *Service job map* consists of 12 steps and can fully consider the characteristic that service begins with defining service needs and accessing to service providers and ends with payment. Therefore, *Service job map* well represents service processes with great specificity. Since *Service job map* is designed for service innovation, it seems to be similar to a service blueprint. In fact, *Service job map* and service blueprint are basically similar in terms of the visualizing framework for a service process. However, the clear difference between them is the *job's* perspective. The goal of *Service job map* is not to capture significant service encounters but to represent what a customer must accomplish to obtain the service [16].

After defining *Jobs-to-be-done* for a specific service, how customers use the service from the *job-to-be-done* and *outcome* perspectives should be identified based on the *Service job map* framework. Each step describes what customers want to achieve; not what customers are doing. Domain experts usually define all *jobs* and *outcomes* based on their knowledge and insight. However, considering that some of the defined *outcomes* are selected as service opportunities, it seems to be risky to rely solely on experts' subjective insight for *job* and *outcome* definitions. If they look over some critical *outcomes*, some valuable service opportunities can never be discovered by ODI. In addition, ODI requires a large number of interviews, 100+ interviews for usual cases and 200+ for complex markets [9], and it costs many expenses and times. Therefore, this paper adopted *Service job map* as the analytic framework and mapped *job-to-be-done* extracted from review data on each step of the *Service job map*.

There exist several attempts to apply the ODI process in other fields. Lim et al. [17] proposed a semi-automatic method to construct a simple *Universal job map* for a specific product from patent data. By analyzing patent documents based on ODI's job classification, *job-to-be-done* were identified. However, since this research focuses on product opportunities using patent data, the analytic process structure and data are basically different from service-related research. Joung et al. [18] applied ODI to detect customer complaints in end-user products. This research extracts keywords and functional information from webpages and classifying them into *Universal job map* steps using semantic similarity analysis. Based on the clustered information in each step, ODI experts define the complaints of the product. Even though these studies tried to develop a data-driven ODI approach, the research purpose, method structure and data are fundamentally different from the proposed method and insufficient to apply to identify service opportunities.



**Figure 2.** Service job map, redrawn from Bettencourt [16].

## 2.2. Multi-Class Sentence Classification Using BERT

There have been many NLP approaches to classify a sentence into one of the defined classes. To run a machine-learning algorithm for NLP, textual information is usually converted to numerical feature vectors using representation models. Even though early representation models, such as a term frequency-based one-hot vector space, are simple and fast, they are basically difficult to reflect semantic similarities between sentences [19,20]. Deep learning techniques have been recently recognized to be a breakthrough for text classification and other NLP tasks. The powerful improvement neural networks made is incorporating context information.

Recurrent neural network (RNN) as the representative deep learning algorithm for NLP encodes information given until time step  $t-1$  into hidden internal state and computes it with new information given at time step  $t$  [21]. This incorporation of previous information enables RNN to conduct more sophisticated text classifications by considering semantic relationships. However, RNN inherently suffers from exploding or vanishing gradient problem [22]. Tokens must be transformed into a vector in order to be fed into a neural network. GloVe [23], Word2Vec [19] and FastText [24] are a pre-trained word embedding model that converts a token into a vector. Neural networks understand the meaning of a sentence by calculating the vectors in a variety of ways. Vanishing gradient problem occurs if the information on early tokens is diluted when the last token is calculated. If a sentence has  $n$  tokens, the first token's vector is multiplied with the same matrix at least  $n$  times. If the matrix has an element less than 1, which is what most matrixes are, the final value would be so small that the gradient vanishes. If the gradient is too small, the neural network becomes untrainable since feedforward neural networks such as RNN are

trained by a backpropagation algorithm [25]. The longer the sentence is, the bigger the problem of vanishing gradient is. The attention network is a network that is free from the vanishing gradient problem. It can incorporate context information efficiently since it allows modeling of dependencies without regard to their distance in the input or output sequences [26]. Using an attention network, neural networks can process complex semantic information. More recently, BERT, the pre-trained language model, was suggested based on the self-attention mechanism used for attention networks [11]. BERT trains a universal language model from the massive corpus and so can be utilized for many NLP tasks by fine-tuning. In particular, since BERT has achieved great performances in sentence-level NLP tasks, it is suitable for our purpose for sentence classifications.

Therefore, this paper classifies each review sentence to one of the steps in a *Service job map* by attention network-based multiclass sentence classifications using BERT.

### 3. Method

The overall procedure for the proposed method consists of six steps:

1. Collection of online review data;
2. Sentence tokenization and cleansing using NLTK (natural language toolkit);
3. Sentence classification to one of *Service job map* steps using BERT-based multiclass sentence classification;
4. *Job-to-be-done* extraction by using a syntactic analysis;
5. Transforming *jobs-to-be-done* to customer *outcomes* using BERT-based semantic similarities;
6. Service opportunity discovery based on importance and satisfaction of outcomes.

#### 3.1. Collection of Online Review Data

As the first step, customer online review data for a specific service is collected from websites, such as Trustpilot, Trip Advisor or Yelp. Web scraping is usually used to extract web data, such as title, review, date, rating, and user identity. To gather review data related to a specific service, a search query or category should be defined.

#### 3.2. Data Preprocessing

Review data needs to be refined for better textual analysis. Each review usually consists of multiple sentences, and they should be transformed into a line-sentence using a sentence-tokenization in NLTK. Then, stop words should be eliminated. Unlike usual text analysis, since customer behaviors are important information for this research, and so subjects related to people should be considered. To preserve information related to people, we excluded terms related to people, such as "I" or "them", from the stop word list.

#### 3.3. Sentence Classification to Service Job Map

Multilabel classifications commonly perform the problem transformation where multilabel problems were transformed into one or more single-label problems, such as a multiclass classification [27]. Binary relevance (BR) is the most common method when problem transformation is performed [28]. BR changes a multilabel problem into a series of binary problems that each binary problem predicts the relevance of one of the labels. When the dependency between labels is weak, BR can achieve higher quality [29]. BR neural network is not a popular method since each isolated binary network needs training while ignoring dependencies between labels [30]. When samples are assigned to a single label, the problem is called multiclass classification. In such a case, BR is not a good method to classify samples because BR can assign multiple labels at the same time. In the multiclass classification problem, the one-vs-all method is widely used [31]. The one-vs-all method makes the binary decision whether a sample belongs to one class or the others. By calculating the probability for all classes, the class with the highest probability is selected. In this way, only one class is assigned to a sample.

In the domain of neural networks, multiclass text classification can be divided into the two types: threshold-dependent neural network (TDNN) or binary relevance neural



network (BRNN) method using the BR method. TDNN trains every activation function at once, while RBNN trains one function at a time. It is known that the more labels are independent, the better quality BRNN can make. Based on the ODI theory, we considered that each step in *Service job map* is independent enough.

We also assumed that adding additional semantic descriptions on what customers must achieve at each step will help neural networks for better performance. Every step in *Service job map* is described with a short sentence (Figure 2). Then this short sentence was given to BERT as a piece of semantic information. BERT is trained to calculate the semantic similarity between a review sentence and the semantic descriptions (Figure 3). Among given semantic descriptions, one having the highest semantic similarity with the focal review sentence can represent the review sentence, and, therefore, the review sentence is assigned to the *Service job map* step related to the semantic description. From here, we call this approach as semantic similarity-binary relevance (SS-BR) method.



**Figure 3.** Binary relevance text classification using bidirectional encoder representations from transformers (BERT).

### 3.4. Extraction of Job-To-Be-Done

#### 3.4.1. Parsing Sentences into Clauses

*Job-to-be-done* is a function from the customer's perspective [5,14], and so it is basically the same format or expression with a general functionality. Function information in texts is usually expressed by verb-object (VO) [32] or subject-action-object (SAO) [33]. SAO structure as an extended expression for function information represents a clear relationship between subject and functional action [34]. In particular, since subjects for function information having the meaning of *job-to-be-done* for customers are basically customers, i.e., people, it is better to extract SAO structures.

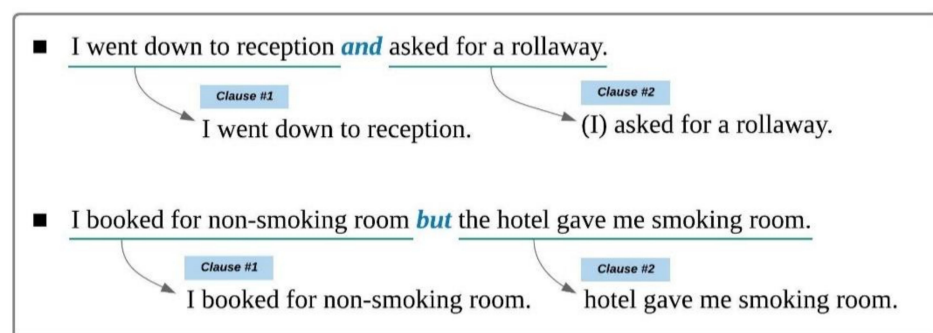
To extract SAO structures from a sentence, a syntactic analysis should be conducted. However, it is usually difficult to extract SAO structures from a complex sentence. Hence, a complex sentence needs to be decomposed into simple clauses for better extractions. Eight syntactic dependencies were used to describe core clausal relations (Table 1), and *markers* like the word introducing a finite clause subordinate to another clause is a dependent of the subordinate clause head. However, every clause does not start with a *marker* that fits our needs. For example, in the sentence "I tried to finish it", "to" is a *marker* and "finish it" is a clause. However, "finish it" does not contain any meaningful customer actions. Based on qualitative investigating a number of different types of reviews, we found that clauses linked by conjunction and that-clause are the two clauses having no exception. The dependency tag *marker* and *conjunction* are the two types we used for this research. For example, if token "that" has *marker* dependency, which links a clause, tokens that come after "that" tokens consist of a clause or multiple clauses. However, not all "that" token has

marker dependency. When “that” is used as demonstrative pronouns, such as in “I want that toy”, it will have *determiner* dependency.

**Table 1.** Core syntactic dependencies for clausal predicates.

Tag	Description
<i>nsubj</i>	Syntactic subject of a clause
<i>nsubjpass</i>	Syntactic subject of a passive clause
<i>dobj</i>	Object of the verb phrase
<i>iobj</i>	Indirect object of a verb phrase
<i>csubj</i>	Clausal subject of a clause
<i>csubjpass</i>	Clausal subject of a passive clause
<i>ccomp</i>	Clausal complement of a verb or adjective
<i>xcomp</i>	Open clausal complement of a verb or adjective

Sentences with conjunction also introduce additional clauses (Figure 4). When conjunctions are found in sentences, every possible SAO structure should be extracted. Conjunctions can be placed between sentences, clauses, noun phrases and verbs. A noun phrase is a phrase with a noun token as its head and adnominal tokens.



**Figure 4.** Parsing sentences with conjunction into clauses.

When conjunctions are placed between sentences and clauses, it can identify by using the same method. When conjunctions are placed between sentences or clauses, the token that comes after the conjunction is a noun phrase, and it has a parent token whose Part-of-speech (POS) is a verb. In such a case, the latter sentence or clause is detached with a conjunction. If a conjunction is attached to the former sentence or clause, it makes them an incomplete sentence, which could make the further analysis more complex. It is relatively easy to identify conjunctions between nouns. The sentence, such as “I asked for early check-in and extra roll-away,” is the case where a conjunction is placed between nouns. Suppose tokens that come before a conjunction are only nouns. If adnominal tokens and noun tokens that come after a conjunction and before a verb share the same verb, then the conjunction is introducing additional subjects. In such a case, the sentence is not parsed, and subjects are extracted as the subject in SAO structure as a whole, including conjunctions.

When conjunctions are placed between verbs, tokens that appear before and after conjunction have the possibility to be independent clauses. A sentence such as “He smiled and walked away” is the case. When conjunctions are placed between verbs, verbs that appear after the conjunction has the *xcomp* dependency and share the same *nsubj* or *nsubjpass* token in the previous clause as its child token. In such a case, the sentence is not parsed, and verbs are extracted as the action in SAO structure as a whole, including conjunctions.

Dependency tag *punct*—punctuation, such as comma—can work the same way as the conjunction works. It can be placed between sentences, clauses, noun chunks and verb chunks and share the same clause-parsing method with the conjunction. One unique characteristic of *punct* is that it can insert a clause or phrase in a sentence. A sentence such as “Mike, the defending champion, lost the race” or “Mike tried hard, but even he tried, he

lost the race” is the case. The former case has phrases between punctuations, and the latter case has clauses between punctuations. In both cases, tokens that lie between punctuations are extracted and treated as an independent clause. SAO structures are extracted from the clause. The extracted phrase itself does not contain any customer actions because it has no verb in it. Therefore, the extracted phrases are deleted.

### 3.4.2. Separating Phrases in SAO Structure

Pseudocode for extracting subject phrase, verb phrase and object phrase is as shown in Figure 5. A sentence that starts with a conjunction should be considered to extract noun phrases. In the previous issue, one sentence that multiple sentences are connected by conjunctions is parsed into independent sentences. In such a case, the latter sentence may share the same subject as the former sentence and hence, the subject token is missing. To solve this problem, we assumed the clause shares the same subject with the previous clause when the sentence starts with a conjunction, and the subject is missing.

#### Noun phrase extraction

```
for every token
  if pos == CCONJ then SUBJECT is unnecessary
  if (token.dep == nsubj or nsubjpass or head.token.dep == nsubj or nsubjpass) -> SUBJECT
  <repeat until head.token.dep == ROOT>
```

#### Verb phrase extraction

```
for every token
  if token.dep == ROOT or (token.pos == VERB or token.head.pos == ROOT)
  if token.dep == xcomp & pos == VERB
  if (token.dep == [aux, auxpass, neg, prep, acomp] & pos == [AUX, ADV, VERB, PART, ADP, ADJ]) & token.head.dep == ROOT or (token.head.pos == VERB and token.head.head.dep == ROOT)
  if token.dep == aux and token.pos == PART
  if OBJECT is found & token.dep == [compound, nsubj, pobj, amod, nummod, ccomp, xcomp, attr] & pos == [NOUN, ADJ, NUM, VERB, PROPEN]
```

#### Object phrase extraction

```
for every token
  if token.dep == [dobj, pobj, prep] or token.head.dep == [dobj, pobj]
  if token.dep == [attr, compound, npadvmod] & pos == [NOUN, PROPEN]
  if token.dep == [amod, ccomp, aux, attr, case, subtok] & pos == [ADJ, PART]
  if token.dep == advmod & pos == ADV
  if token.dep == acomp & head.dep == ROOT
  if token.dep == acl & pos == VERB
  if OBJECT is found & token.dep == [compound, nsubj, pobj, amod, nummod, ccomp, xcomp, attr] & pos == [NOUN, ADJ, NUM, VERB, PROPEN]
```

Figure 5. Pseudocode for extracting subject-action-object (SAO) structures from clause.

### 3.4.3. Filtering Out Irrelevant SAO Structures

After extraction of SAO structures, SAO structures irrelevant to a *Service job map* should be filtered out from the set. Since *job-to-be-done* are basically related to SAO structures for customer actions, SAO structures having no people/person as the subject can be considered irrelevant ones. For this, tokens referring to users as “I” and “we” were identified, and then they were replaced by token “user”. Therefore, SAO structures having no token “user” in the subject were filtered out from the set. To filter out SAO structures having no actions,



SAO structures whose action is be-verb, such as “am”, “is”, “was” and “were”, are usually about customer states or emotions.

### 3.5. Transforming Job-To-Be-Done to Customer Outcomes

After filtering out the irrelevant SAO structures, we represented the action in a single token to cluster similar actions. The original form of ROOT token can be used as the representative one. Verbs, such as gerund, past particle or passive voice, can all be represented in the same way. After verb phrases are represented in a single token, SAO structures are clustered by the pre-trained BERT-based semantic sentence similarities.

The verb-object (VO) structures in the clustered SAO structures can be considered as semantically similar *job-to-be-done*. The VO structures should be transformed into one representative customer *outcome*. The basic structure of *outcome* is the direction—metric—object or *job-to-be-done*, and thus, each SAO structure should be linked with relevant metrics and directions. There exist six types of metrics for service-related outcomes: frequency/possibility, energy/effort, cost, performance/accuracy/speed, safety, and reliability. Each metric is related to only one specific direction (Table 2).

**Table 2.** Types of metrics and their directions.

Metric	Direction
Frequency/possibility	Maximize
Energy/effort	Minimize
Cost	Minimize
Performance/accuracy/speed	Maximize
Safety	Maximize
Reliability	Maximize

To extend VO structures to the form of *outcomes*, each SAO structure should be linked with the relevant metrics. We identified the top two metrics close to the full sentence of the given SAO structure by using the pre-trained BERT-based semantic similarities. Based on the pairs of the metrics and VO structures, one representative customer *outcome* for each cluster is qualitatively defined.

### 3.6. Service Opportunity Discovery

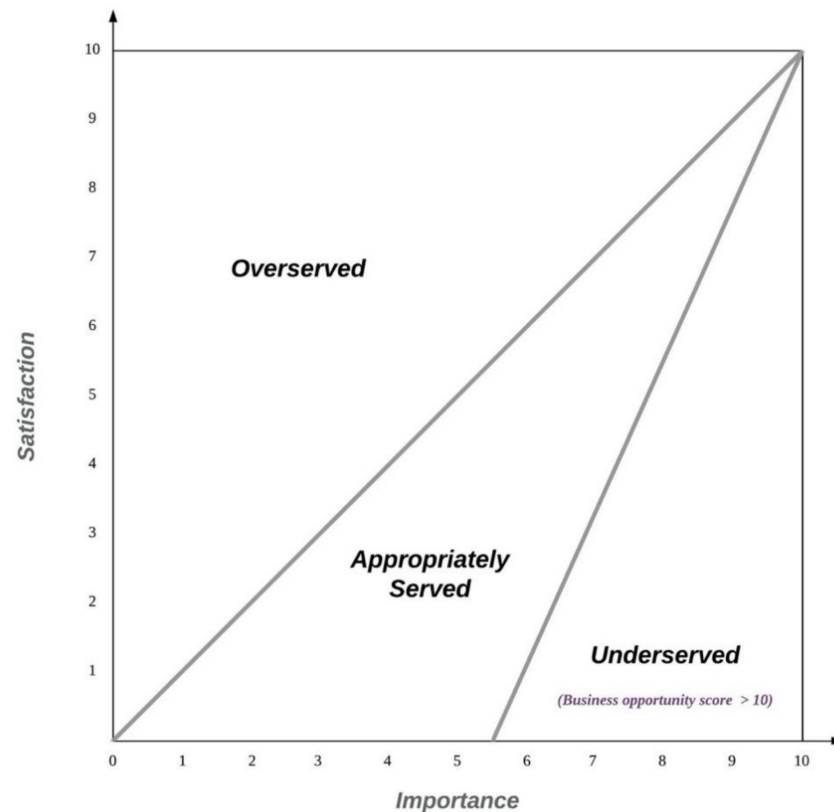
ODI framework defines business/service opportunities as desired *outcomes* that are important, but current satisfaction from the existing services is low (Figure 6). Based on this, ODI calculates the degree of service opportunity based on the gap between the importance and satisfaction score for the *outcome*. Specifically, ODI quantifies the importance and satisfaction of *outcomes* on a scale from 0 to 10. The formulation of service opportunities is as follows:

$$\text{Service opportunity score (SOS)} = \text{Importance} + \max(\text{Importance} - \text{Satisfaction}, 0)$$

This research quantifies the importance and satisfaction of the *outcome* as follows:

- The importance score can be calculated based on the number of SAO structures—, i.e., *job-to-be-done*—in each cluster for the customer *outcome*; frequently occurred *job-to-be-done* are, at least, important issues in a service. However, since the occurrence frequency distribution is usually skewed and so the size of one or some clusters can be far larger than the others, the normalization is difficult to transform the scores into a scale from 0 to 10. Therefore, we utilized *k*-means clustering, *k* = 11.
- The satisfaction score is calculated based on the difference between the number of SAO structures representing positive and negative opinions. The sentiment analysis for SAO structures was conducted by utilizing a BERT-based sentiment analysis [35]. To use contextual information around SAO structures, the full sentence for each SAO structure is analyzed. To train BERT for the sentiment analysis of review sentences in

a specific service sector, the training sentences clearly representing negative opinions were labeled as negative opinions, and the other sentences were labeled as positive opinions. The range of the satisfaction score also should be on a scale from 0 to 10. Therefore, we developed the equation as follows:  $Satisfaction = 5 * ((p - n)/T) + 5$ , where  $p$  is the number of positive SAO structures for a specific *outcome*,  $n$  is the number of negative SAO structures for a specific *outcome*,  $T$  is the total number of SAO structures for a specific *outcome*. The range of  $(p - n)/T$  is from -1.0 to 1.0, and so the range of *Satisfaction* is from 0 to 10.



**Figure 6.** Outcome-driven innovation (ODI) framework for service opportunity score, redrawn from Ulwick [5].

#### 4. Empirical Analysis: Hotel Service

This research conducted an empirical case study for hotel service. The hotel service, as one of the services with the longest history, has established service processes and systems as the basic framework, and so it is difficult to find the crucial points for service innovations. In addition, from the methodological perspective, customer review data related to hotel service is relatively easy to collect, and hotel service includes various additional services, such as facilities, restaurant, and room service and so there are many interactions with customers. Therefore, the hotel service is the case that service opportunities can be identified from various steps in the *Service job-map*, and so we deemed hotel service is a suitable case to be applied to the proposed method. Moreover, since previous studies found that online review data on hotel service is useful for a business intelligence approach [36,37], the hotel service can be a good case to test the proposed method.

We collected customer reviews from Trip Advisor ([www.tripadvisor.com](http://www.tripadvisor.com)). Since reviews with low ratings generally have rich information to find customer needs, reviews rated as one or two points/stars out of five were only considered as the data set. Three hundred eighty-five reviews on Royal Plaza Hotel from 2008 to 2018 were collected in such conditions. Four thousand six hundred fifty-four sentences were extracted, but 1361 sentences were selected as the final data set after cleaning noise sentences; 85% of the

sentences were used as the training set, and the rest 15% were used as the test set. Figure 7 shows the occurrence distribution of sentences for each step in *Service job map*. No sentence was assigned to step 1: “Contact the service provider and/or access service”. This step is about customers contacting the hotel for the first time, such as a phone call or website visit. Those actions were not found in customer reviews. Reviewers tend not to mention how they first contacted the hotel. The first interaction that reviewers often mention is “booked room”, which is step 4: “Confirm and/or finalize service plan”. Since steps 1–3 are done before they arrive at the hotel, there were little reviews related to the step 1–3.

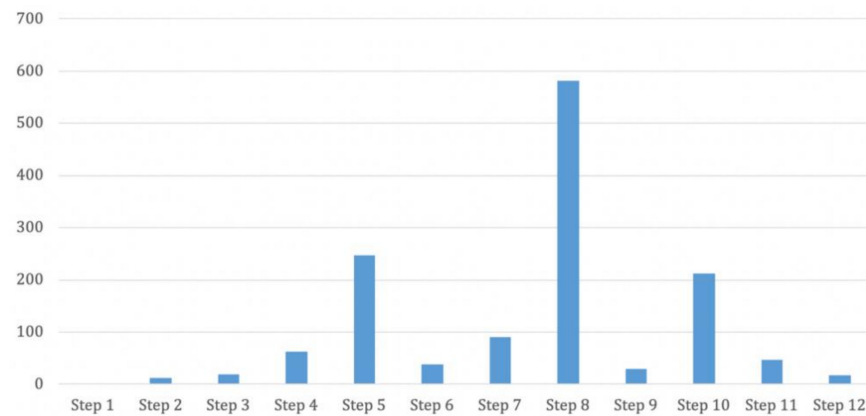


Figure 7. Sample sentence's distribution.

To test whether SS-BR performs better than TDNN using BERT, we used BERT with 12 transformer blocks, 768 hidden sizes and 12 self-attention heads and pre-train it with uncased English vocabulary. The structural difference between TDNN and SS-BR is shown in Figure 8. In TDNN, the first node of the last layer is connected with 13 activation functions by fully connected layers. Twelve activation functions correspond to 12 steps in the *Service job map*, and one activation function is used for sentences that do not belong to any of the 12 steps. The input to TDNN is a single review sentence, and the output is one of the labels between 0 and 12, where label 0 denotes noise.

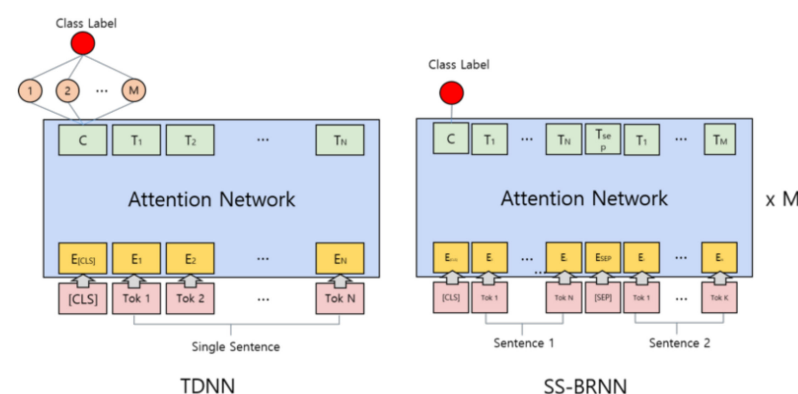


Figure 8. Structural difference between threshold-dependent neural network (TDNN) and semantic similarity-binary relevance (SS-BR), redrawn from [11].

In SS-BR, the first node in the last layer is connected with an activation function. The input to the SS-BR neural network consists of two sentences where the first sentence is a review sentence, and the second sentence is a semantic sentence that describes each step of *Service job map*. Since there are 12 steps, one review sentence must go through 13 different BERT with a different semantic sentence each time. The range of the output is (0,1), and it represents the semantic similarity. After calculating all 13 semantic similarities, a review sentence is assigned to the step with the highest similarity.

#### 4.1. Technical Result

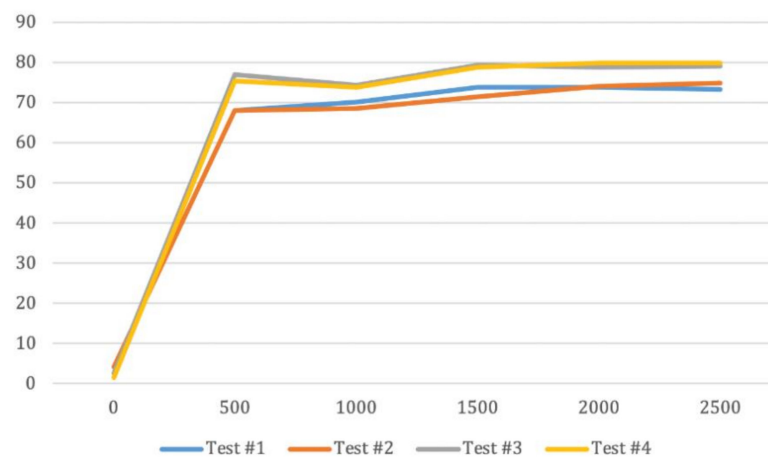
Accuracy was calculated under batch sizes 16 and 32, 30 epochs, and  $2 \times 10^{-5}$ ,  $5 \times 10^{-5}$  and  $7 \times 10^{-5}$  learning rate with the decaying rate of 0.01 on both TDNN and SS-BR. Since we used a pre-trained BERT model, each run shares an identical initial state. Training data were fed into the TDNN and SS-BR after being randomly shuffled. This shuffling is the only random process, and so we ran 5 different runs and calculated the average accuracy. Among 30 epochs, the best result is saved, and the trained model is saved for every 500 steps as a checkpoint. TensorFlow library in Python was used with Intel i7 3.0 GHz processor and Nvidia GeForce GTX 1080Ti graphic processing unit. Under these conditions, the result is as follows in Table 3. SS-BR was at least 4.7% point higher than TDNN and 2.8% point better on average. Based on the result, we concluded that each step of *Service job map* is independent, and thus SS-BR outperforms TDNN.

**Table 3.** Accuracy test of TDNN and SS-BR.

Test Number	TDNN (%)	SS-BR (%)
T1	73.58	78.05
T2	72.85	79.21
T3	74.16	79.83
T4	73.14	73.29
T5	68.03	74.89
Average	72.35	77.05

By calculating accuracy for every 500 steps, it is learned that the accuracy of SS-BR converges to its upper bound is less than 2 epochs (Figure 9). The accuracy gets lower as trained epochs get larger. This means that the training set and SS-BR method reaches the point where overfitting happens very quickly. Overfitting is a phenomenon that a neural network has learned from training sets so excessively that trained parameters in the neural network can memorize the training sets and shows very accurately while it predicts test samples in low accuracy. Because the neural network has not already seen the test samples while training, it cannot predict anything about test samples. To solve this problem, a neural network must be tested with samples not used for training. As mentioned earlier, we used 85% of samples for training and 15% for testing them. In general, 50~70% of total samples are used for training, 15~30% are used for validation, and 15~30% are used for tests. Validation is a process that human experts examine the output of a neural network and adjust the neural network structure itself or determines when training should stop to prevent overfitting. In our case study, neural network structure, such as a number of layers and input vector size, is already determined. In addition, by calculating accuracy for every 500 steps with the test sample, we could point out where the overfitting starts to happen.

We tagged syntactic dependencies and part-of-speech (POS) by using spaCy library in Python. Based on this tagging, meaningful clauses in a sentence can be extracted (Table 4). From 1299 sentences, we extracted 2100 clauses, and 31 of them were incomplete (Table 5). Since customer reviews are an informal and grammatically incomplete text, it is difficult to perfectly extract all clauses in a sentence. Most incomplete cases are caused by the omission of the subject. Reviewers sometimes omit subject, usually "I", when the context shows that the subject of a sentence or clause is clearly the reviewer. Since the proportion of incomplete clauses is very low, less than 2.6%, we did not focus on this issue and deleted them from the data set.



**Figure 9.** Accuracy of four tests for 500 training steps.

**Table 4.** Clause examples parsed from review sentences.

No.	Clause
1	The first few emails returned,
2	they offer me to upgrade to a Grand Plus
3	Speechless but it seems that
4	RPH wants everyone to upgrade to the best package
5	and all it is, is the same room with a different bed.
6	Couple of weeks later, I realize the hotel rate is lower
7	which was about HKD7000++
8	(I couldn't remember the exact rate)
9	which I find it acceptable,
10	although I'm paying a premium
11	the spread is still acceptable.
12	On 28 Sep I logon to the website again
13	and discovered that
14	the rate was much lower at HKD6968.50
15	where I was surprised.
16	No disclosure of any of this maintenance was given at time of booking or on arrival.
17	This information was not provided when
18	I made my reservation through Hotel Club 2 months ago.
19	Having checked the hotel website prior to booking,
20	which advertised
21	the deluxe room came with a king size bed or twin queen sized beds,
22	we tried to check prices on the website
23	but could only find prices for a king bed room.
24	We therefore assumed that
25	the prices for these rooms would be the same,
26	as all hotels in the UK for twin and double rooms are the same price.
27	We double checked the hotel's website-
28	whilst we could find a rate for a king size room,
29	nowhere could we find a rate for a twin room.
30	Nowhere on the website was it mentioned that
31	the rate for a twin was different from a king size room.
32	The lack of transparency of pricing, and lack of helpfulness at the reception desk left us feeling
33	we were being conned especially
34	as other hotels in Hong Kong charge the same price for a twin/king bed room.
35	I had a bad experience
36	when I had already book a family (plan to travel march 2013)
37	indicated for 4 adults
38	and it mentioned under notes below (in terms and condition)
39	that is free for 2 children under 12 years old
40	when using existing bed.



Table 5. Examples of incomplete clauses.

No.	Clause
1	Se swapped to British pounds
2	Hint get an express checkout
3	One downside, no cocktails in club lounge even though advertised as such
4	Security check at lift lobby during evening
5	in order to get another room.
6	Despite the long wait at the reception area when checking in,
7	Despite having the do not disturb light on
8	When asking to assist in contacting our travel agent.
9	and despite the late hour still had to wait nearly 20 min for check-in and a key.
10	and still not being informed on another room.
11	To our utter dismay and horror on our arrival home
12	Especially on the last day of stay
13	Ensure you check all the charges on your bill
14	Just nom nom with your money
15	post checking out and paying for our stay
16	Only after checking the CCTV
17	made a compliant to the front desk
18	had to wait until the following day and moved into a much better couple of rooms
19	After being told that
20	Was not sure what caused the delay was,
21	Called down to front dest & waited.
22	Requested for a change of room on the first day of our visit via the phone
23	Upgraded on arrival to Executive Floor, much nicer rooms and access to Lounge.
24	Check with the hotel concierge for airport express shuttel frequency + location (with a man in suit on 14/10/2014, approx. 10.45 p.m.)
25	but was given a strange face expression without even answering our question convincingly, and walk away from us after someone ask for him
26	And then again to be doubted by the staffs if we have a room there.
27	Arrived late a night
28	Waited at the lobby
29	and was able to check in slightly before 3 p.m.
30	Booked a family room with 2 double bed . . .
31	the STATEMET ABOVE REALLY BAD AND MISLEADING.

SAO structure was extracted from each clause. Some clauses whose verb is intransitive verb only have subject-action structures. Even though subject-action structures are not typical SAO structures, they are considered valid data set because the purpose of this research is to identify customers' actions. Next, we filtered out irrelevant SAO structures that do not include customer actions. As a result, 1057 SAO structures were retained after filtering out. To identify and cluster similar SAO structures, Actions are considered as a single token. SAO structures are clustered by semantic sentence similarities using the previously trained BERT. In addition, at the same time, each SAO structure is linked with the two most similar metric—direction pairs. To this end, the pre-trained BERT-based semantic similarity between the full sentence of SAO structures and metrics is conducted, and then, each AO structure is extended to a customer *outcome*.

Based on the set of the clustered *outcomes*, one representative customer *outcome* is qualitatively defined. The importance and satisfaction score of each customer *outcome* is quantified by the metrics in Section 3.6. Based on the importance and satisfaction scores, service opportunity scores (SOS) for customer outcomes are evaluated, and Table A1 shows the results.

#### 4.2. Service Opportunity Analysis

The ODI framework considers the *outcomes* whose service opportunity scores is larger than or equal to 10 as customer needs having high service opportunities. Our empirical analysis found seven customer needs in hotel service. Different customer needs were discovered from various steps in *Service job map*. Overall, by counting the number of *outcomes* for each step, a large number of *outcomes* were identified in step 5 and 10 (Table 6). Since step 5, usually about check-in process, is the initial phase of service delivery where what customers have reserved online was checked and delivered, many complaints and demands from customers happen. In particular, the most important customer needs ( $SOS \geq 14.0$ ) for hotel service innovations were mainly identified in step 5. About step 10, the main delivery of hotel service is to sell a room, and so changing the delivered service is basically difficult. Therefore, it usually takes a long time, or is difficult, to solve the service quality-related problems. Hence, many customer needs and complaints are identified in this step.

**Table 6.** Occurrence frequency of customer *outcomes* at each step.

Step #	Definition	# of Representative Outcomes	# of Identified Outcomes
2	Define and/or communicate service needs	4	10
3	Evaluate and/or select service options	5	12
4	Confirm and/or finalize service plan	6	50
5	Initial service delivery	28	120
6	Fulfill customer responsibilities	8	9
7	Receive service	11	19
8	Evaluate and/or monitor service delivery	9	17
9	Adjust service plan and/or its execution	3	3
10	Get questions answered and/or problems resolved	34	64
11	Conclude service	4	14
12	Pay for service	4	16

In particular, the following results show specific customer needs for service innovations:

- “Minimize cost (to) check in the hotel early” (CO #17, SOS = 14.0) is the *outcome* whose SOS is the highest in our results but seems to be very common, and every hotel service provider already knew most customers want this *outcome*. However, few know how much relatively important this *outcome* is. Our results quantitatively show that CO #17 is very important, but its current satisfaction is low. Therefore, the strategic directions for hotel service innovation should focus on CO #17 to maximize customer values. To deal with this issue, various service options for early check-in can be developed. At the same time, other relevant issues for service implementation, e.g., fast room cleaning after checkout, should be considered.
- The outcome “Maximize safety (to) pay price/deposit” (CO #26, SOS = 14.0) is caused by negative customer experiences related to online payment. Recent online payment systems usually assure high reliability for online monetary transactions. However, given that price, time and memory are the most significant factors in traveling, online payment problems can waste customer time and good memory in the trip. Therefore, many reviewers mentioned this issue in online reviews. Hotel service providers need to prepare solutions to minimize risks caused by relevant issues.
- The outcome “Maximize possibility (to) avoid waiting at the lobby for check-in, or other purposes” (CO #30, SOS = 12.5) is a very common customer need. However, our result shows customers feel this issue is very important, but most hotel services do not fulfill it.

- The outcome “Maximize possibility (to) assign to a preferred room” (CO #96, SOS = 12.0) is evaluated as one of the important business opportunities. From the relevant reviews, we found that this outcome is usually related to the two expectations: free room upgrade and assigning to the right room booked by a customer. If a customer is assigned to an unexpected room, his/her trip may be ruined. In particular, the smoking issue is one critical example. Some reviews were about a free upgrade and mentioned that they had a great memory at the hotel due to the free upgrade service.
- “Maximize possibility (to) avoid billing mistake, due to overcharging, ignored payment by cash, overcharging tips, etc.” (CO #110, SOS = 12.0) sounds similar to CO#26, but the difference is that CO#110 is caused at the hotels by internal problems, and CO#26 is caused by online payment problems. Most employees may not seriously think about these mistakes. However, the result clearly shows that customers highly matter this *outcome*. Therefore, a strict guideline or education program should be developed to reduce the possibility of internal billing mistakes.
- The outcome “Minimize effort (to) check prices” (CO #5, SOS = 10) is closely related to hotel searching websites. Based on the dynamic pricing strategies, the price for the same room can be changed over time. Every customer wants to buy at a lower cost with a minimum effort. Even though the strategic scope for this outcome is not directly inside hotel service, hotel service providers should find ways to fulfill this *outcome*. Some creative service options may be a solution to deal with this *outcome*. To offer a lower price than booking websites to the registered existing customers can be one example.
- The outcome “Minimize cost (to) book hotel” (CO #11, SOS = 10.5) is about the basic customer need in any service. All customers are sensitive to price, so it makes sense that this *outcome* gets a relatively high score. Recently, it is inevitable for Hotel service to compete with a new type of accommodation services, such as Airbnb. Therefore, according to the results and current social, business, and technology trends, we can think that the traditional hotel business is forced to be changed, or innovated, for sustainable competitive advantages. Our results provide some strategic directions for business innovation in hotel service.

## 5. Conclusions

This research aims to develop a quantitative method to identify customer needs in a specific service sector by analyzing online review data based on the ODI framework using BERT-based deep learning techniques. The major contribution of the proposed method is as follows: First, this paper is the first attempt to develop a data-driven ODI approach for service sectors. More specifically, previous studies using a data-driven ODI approach focus on identifying product opportunities, and so they are based on *Universal job map*, which is the eight-step job map for universal purpose, so not well applicable for service sectors. However, this paper adopted *Service job map* instead of *Universal job map*, and so most of the analytic processes were redesigned. Second, the performance for matching/classifying the extracted SAOs to one of *Service job map* steps is improved by using a BERT-based attention network. Specifically, each sentence in review data is assigned into one of the steps in *Service job map* using BERT-based multiclass sentence classification. Since this process can directly affect the characteristics of the defined outcomes, this process is critical for the proposed method because this process can directly affect the characteristics of the defined *outcomes*, and this paper, in particular, adopted a more complex *job map*, i.e., *Service job map*. Third, the proposed method automatically generates the *outcome* statements for every step in *Service job map*. Each *job-to-be-done* is extended to the form of customer *outcome*, and the service opportunity score (SOS) for each outcome is evaluated based on the importance and satisfaction scores. Generating various but reliable outcomes from different job map steps with minimizing cost is one of the most important demands in the ODI framework. Even though previous studies [17,18] could not deal with this issue, this paper developed the method that fulfills the demand. Fourth, since the proposed

method can automatically generate outcome statements, the method can assess potential opportunities for the outcomes without any expensive interview process. Therefore, this research is the first attempt to identify service opportunities based on a data-driven ODI approach without an interview process.

To test the proposed method, this paper conducted an empirical analysis of the hotel service sector. The method found significant *outcomes* to develop further service strategies in the hotel service sector. In particular, the practical advantages of the proposed method are to identify unrecognized or hidden customer needs and to quantitatively denote what *outcomes*, even though some of the sounds common, should be priorly focused on to develop customer values. Prior research on analyzing hotel service using online customer review [38] have calculated keyword frequencies and divided reviews into four groups. Other studies [39,40] also use keyword frequencies to analyze a large amount of text data. Considering that “billing mistake” caused the hotel’s internal process problems and online payment problems, keyword-frequency-based analysis has limitations in identifying such context. Based on the results, new value propositions should be developed to better fulfill the discovered service opportunities.

However, some issues should be considered in further work. First, the large size of review data should be secured to improve the quality of results. The proposed method basically adopts many machine learning, specifically deep learning, techniques, and so the size of the training set is one important factor in assuring reliability. Second, even though a neural network performs better than previous networks, its accuracy is not yet sufficient to capture all meaningful sentences. In addition, when new customer action is introduced, and customer reviews on that action are little, neural networks may fail to classify that action as relevant. Third, this research transforms the clustered outcomes into one representative outcome based on qualitative knowledge. In fact, there exist some approaches for automatic sentence inferencing, but our toy-test in sentence inferencing for this research, or hotel service case, provided an insufficient performance. Even though we adopted an expert-based representation to increase the quality and reliability, an automatic sentence representation can provide a huge advantage for this research.

Our further work will focus on neural network structure to perform better with less training datasets. Neural network techniques, such as transfer learning, are being considered. By collecting data from various service industries, the neural network could be trained to classify general *jobs*, and by further train task-specific data, the model is tuned to produce a desirable outcome. To utilize transfer learning, services that share a common structure for the process must be clustered beforehand, which is also another future research topic. To overcome the limitation regarding the outcome cluster issue, GANs (generative adversarial networks)-based sentence generation [41] is a carefully considered possible approach.

**Author Contributions:** Conceptualization, S.N., N.R. and H.P.; methodology, S.N. and S.Y.; project administration, H.P.; supervision, H.P.; writing—original draft, S.N. and H.P.; writing—review and editing, S.N., N.R. and H.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by Hanyang University (Grant number: HY-2019).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Data are not publicly available, though the data may be made available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

Table A1. Service opportunity scores of customer outcomes.

Label	Step	Outcome		Occurrence Frequency	# of Positive Outcomes	# of Negative Outcomes	Importance Score	Satisfaction Score	Service Opportunity Score (SOS)
		Direction	Metric (Measure) Job-To-Be-Done						
CO(Customer Outcome)#1	2	Minimize	Energy/Effort (To) ask for bed	3	3	0	2	10	2
CO#2	2	Minimize	Cost (To) check price on website	2	0	2	0	0	0
CO#3	2	Minimize	Energy/Effort (To) request service	2	2	0	0	10	0
CO#4	2	Minimize	Energy/Effort (To) communicate with the hotel	3	2	1	2	6.66666667	2
CO#5	3	Minimize	Energy/Effort (To) check prices	7	0	7	5	0	10 *
CO#6	3	Minimize	Energy/Effort (To) logon website	1	1	0	0	10	0
CO#7	3	Minimize	Energy/Effort (To) make reservation	1	1	0	0	10	0
CO#8	3	Maximize	Safety (To) remember price	1	0	1	0	0	0
CO#9	3	Minimize	Energy/Effort (To) pay price	2	1	1	0	5	0
CO#10	4	Minimize	Energy/Effort (To) ask for amenities	1	0	1	0	0	0
CO#11	4	Minimize	Cost (To) book hotel	43	41	2	10	9.53488372	10.46511628 *
CO#12	4	Minimize	Energy/Effort (To) give rating	1	1	0	0	10	0
CO#13	4	Minimize	Energy/Effort (To) communicate with the hotel	3	3	0	2	10	2
CO#14	4	Minimize	Energy/Effort (To) get a room	1	1	0	0	10	0
CO#15	4	Minimize	Energy/Effort (To) travel in a group	1	1	0	0	10	0
CO#16	5	Maximize	Performance/ Accuracy/Speed (To) be assigned to right rooms	18	11	7	8	6.11111111	9.88888889
CO#17	5	Minimize	Cost (To) check in hotel early	30	12	18	9	4	14 *
CO#18	5	Minimize	Cost (To) communicate with the hotel	5	2	3	4	4	4
CO#19	5	Minimize	Energy/Effort (To) ask hotel policy	1	0	1	0	0	0
CO#20	5	Minimize	Energy/Effort (To) ask for sofa	1	1	0	0	10	0
CO#21	5	Minimize	Cost (To) get extra bed	2	0	2	1	0	2
CO#22	5	Minimize	Cost (To) ensure free WIFI	1	1	0	0	10	0
CO#23	5	Minimize	Energy/Effort (To) explain room specification	2	1	1	1	5	1
CO#24	5	Minimize	Energy/Effort (To) send booking notes to the hotel	1	0	1	0	0	0



Table A1. Cont.

Label	Step	Outcome		Occurrence Frequency	# of Positive Outcomes	# of Negative Outcomes	Importance Score	Satisfaction Score	Service Opportunity Score (SOS)
		Direction	Metric (Measure) Job-To-Be-Done						
CO#25	5	Minimize	Energy/Effort (To) get email confirmation	1	0	1	0	0	0
CO#26	5	Maximize	Safety (To) pay price/deposit	12	0	12	7	0	14 *
CO#27	5	Maximize	Safety (To) go shopping	4	2	2	3	5	3
CO#28	5	Maximize	Performance/ Accuracy/Speed (To) get confirmation	2	0	2	1	0	2
CO#29	5	Maximize	Frequency/ Possibility (To) be notified of cancellations	2	0	2	1	0	2
CO#30	5	Maximize	Frequency/ Possibility (To) avoid waiting at lobby for check-in (or other purposes)	13	2	11	7	1.53846154	12.46153846 *
CO#31	5	Maximize	Frequency/ Possibility (To) get a prepaid voucher	1	1	0	0	10	0
CO#32	5	Minimize	Cost (To) have amenities	1	1	0	0	10	0
CO#33	5	Maximize	Frequency/ Possibility (To) be notified of cancellations	1	0	1	0	0	0
CO#34	5	Maximize	Frequency/ Possibility (To) get back left/missing luggage at hotel	3	2	1	2	6.66666667	2
CO#35	5	Maximize	Frequency/ Possibility (To) make sure non-smoking room	1	1	0	0	10	0
CO#36	5	Minimize	Energy/Effort (To) move to new room	1	0	1	0	0	0
CO#37	5	Minimize	Energy/Effort (To) print boarding pass (To) receive reservation	1	1	0	0	10	0
CO#38	5	Minimize	Energy/Effort numbers and confirmation emails (To) request refreshing of room next morning	1	0	1	0	0	0
CO#40	5	Minimize	Cost (To) have enough space	1	0	1	0	0	0
CO#41	5	Minimize	Cost (To) upgrade room	8	5	3	5	6.25	5
CO#42	5	Maximize	Frequency/ Possibility (To) be notified of room readiness	4	0	4	3	0	6
CO#43	5	Minimize	Energy/Effort (To) sign paperwork	1	0	1	0	0	0
CO#44	6	Minimize	Energy/Effort (To) access to the room	1	1	0	0	10	0

Table A1. Cont.

Label	Step	Outcome		Occurrence Frequency	# of Positive Outcomes	# of Negative Outcomes	Importance Score	Satisfaction Score	Service Opportunity Score (SOS)
		Direction	Metric (Measure) Job-To-Be-Done						
CO#45	6	Minimize	Energy/Effort (To) carry luggage	2	0	2	1	0	2
CO#46	6	Minimize	Energy/Effort (To) get invoice	1	0	1	0	0	0
CO#47	6	Minimize	Energy/Effort (To) receive access card	1	0	1	0	0	0
CO#48	6	Minimize	Energy/Effort (To) put do not disturb card	1	0	1	0	0	0
CO#49	6	Minimize	Energy/Effort (To) register as guest	1	0	1	0	0	0
CO#50	6	Minimize	Energy/Effort (To) request cleaning	1	1	1	0	5	0
CO#51	6	Maximize	Frequency/Possibility (To) confirm breakfast at discount rate	1	0	1	0	0	0
CO#52	7	Minimize	Cost (To) buy amenities	1	0	1	0	0	0
CO#53	7	Minimize	Energy/Effort (To) order room service	1	1	0	0	10	0
CO#54	7	Minimize	Cost (To) receive executive lounge service when upgraded	1	0	1	0	0	0
CO#55	7	Minimize	Energy/Effort (To) inquire free shuttle	1	0	1	0	0	0
CO#56	7	Minimize	Energy/Effort (To) get a room cleaning	1	0	1	0	0	0
CO#57	7	Minimize	Energy/Effort (To) use coupon	1	0	1	0	0	0
CO#58	7	Minimize	Cost (To) pay for service fee (e.g., Internet, excluded service, )	6	0	6	4	0	8
CO#59	7	Minimize	Cost (To) register at the hotel	2	0	2	1	0	2
CO#60	7	Minimize	Energy/Effort (To) sign bill	1	0	1	0	0	0
CO#61	7	Minimize	Energy/Effort (To) inquire for the Internet rate	1	0	1	0	0	0
CO#62	7	Minimize	Energy/Effort (To) communicate to concierge	3	1	2	2	3.33333333	2
CO#63	8	Maximize	Frequency/Possibility (To) arrive hotel late	2	0	2	1	0	2
CO#64	8	Maximize	Frequency/Possibility (To) change rooms (due to reasons)	3	0	3	2	0	4
CO#65	8	Maximize	Frequency/Possibility (To) avoid mold	1	0	1	0	0	0
CO#66	8	Maximize	Frequency/Possibility (To) avoid signboard outside	1	0	1	0	0	0

Table A1. Cont.

Label	Step	Outcome		Occurrence Frequency	# of Positive Outcomes	# of Negative Outcomes	Importance Score	Satisfaction Score	Service Opportunity Score (SOS)
		Direction	Metric (Measure) Job-To-Be-Done						
CO#67	8	Maximize	Frequency/ Possibility (To) avoid sewage smell	1	0	1	0	0	0
CO#68	8	Maximize	Frequency/ Possibility (To) avoid ventilation system problem	1	0	1	0	0	0
CO#69	8	Maximize	Frequency/ Possibility (To) avoid insect/bed bug issue	2	0	2	1	0	2
CO#70	8	Maximize	Frequency/ Possibility (To) avoid toilet/shower issue	1	0	1	0	0	0
CO#71	8	Maximize	Frequency/ Possibility (To) avoid odor of cigarette smoke	5	0	5	4	0	8
CO#72	9	Minimize	Energy/Effort (To) approach front desk officer	1	0	1	0	0	0
CO#73	9	Maximize	Frequency/ Possibility (To) check in early	1	1	0	0	10	0
CO#74	9	Maximize	Frequency/ Possibility (To) request change rooms	1	0	1	0	0	0
CO#75	10	Minimize	Energy/Effort (To) accept user's food in hotel's fridge	1	1	0	0	10	0
CO#76	10	Maximize	Frequency/ Possibility (To) arrive hotel late	2	1	1	1	5	1
CO#77	10	Minimize	Energy/Effort (To) ask for extra towels	1	1	0	0	10	0
CO#78	10	Minimize	Energy/Effort (To) ask club lounge member	1	1	0	0	10	0
CO#79	10	Minimize	Energy/Effort (To) ask to concierge for direction	1	1	0	0	10	0
CO#80	10	Minimize	Cost (To) change room	8	1	7	5	1.25	8.75
CO#81	10	Minimize	Cost (To) change bedding and sheets	1	0	1	0	0	0
CO#82	10	Minimize	Energy/Effort (To) chase missing call	1	1	0	0	10	0
CO#83	10	Maximize	Frequency/ Possibility (To) ask to see a doctor	1	1	0	0	10	0
CO#84	10	Maximize	Frequency/ Possibility (To) accept specific room service request	1	0	1	0	0	0
CO#85	10	Maximize	Frequency/ Possibility (To) book a room	1	0	1	0	0	0

Table A1. Cont.

Label	Step	Outcome		Occurrence Frequency	# of Positive Outcomes	# of Negative Outcomes	Importance Score	Satisfaction Score	Service Opportunity Score (SOS)
		Direction	Metric (Measure) Job-To-Be-Done						
CO#86	10	Maximize	Frequency/ Possibility (To) connect (properly) to front desk	8	3	5	5	3.75	6.25
CO#87	10	Minimize	Energy/Effort (To) check rooms	1	0	1	0	0	0
CO#88	10	Maximize	Frequency/ Possibility (To) check in late	1	0	1	0	0	0
CO#89	10	Maximize	Frequency/ Possibility (To) assign to connecting rooms	1	0	1	0	0	0
CO#90	10	Maximize	Frequency/ Possibility (To) avoid from noise	1	0	1	0	0	0
CO#91	10	Maximize	Frequency/ Possibility (To) be notified of renovations	4	0	4	3	0	6
CO#92	10	Maximize	Frequency/ Possibility (To) assign to non-smoking room	1	0	1	0	0	0
CO#93	10	Maximize	Frequency/ Possibility (To) avoid card key problem	1	0	1	0	0	0
CO#94	10	Maximize	Frequency/ Possibility (To) avoid mold problem	1	0	1	0	0	0
CO#95	10	Maximize	Frequency/ Possibility (To) avoid odor of cigarette smoke	4	0	4	3	0	6
CO#96	10	Maximize	Frequency/ Possibility (To) assign to preferred room	9	0	9	6	0	12 *
CO#97	10	Maximize	Frequency/ Possibility (To) avoid sewage smell	1	0	1	0	0	0
CO#98	10	Minimize	Cost (To) pay for another day	1	0	1	0	0	0
CO#99	10	Maximize	Frequency/ Possibility (To) avoid insect issue	1	0	1	0	0	0
CO#100	10	Minimize	Energy/Effort (To) request room refreshing next morning	1	0	1	0	0	0
CO#101	10	Maximize	Frequency/ Possibility (To) avoid lift problem	1	0	1	0	0	0
CO#102	10	Maximize	Frequency/ Possibility (To) sleep comfortable	1	0	1	0	0	0
CO#103	10	Maximize	Frequency/ Possibility (To) avoid mold issue	1	0	1	0	0	0
CO#104	10	Minimize	Energy/Effort (To) get dehumidifier	1	0	1	0	0	0

Table A1. Cont.

Label	Step	Outcome			Occurrence Frequency	# of Positive Outcomes	# of Negative Outcomes	Importance Score	Satisfaction Score	Service Opportunity Score (SOS)
		Direction	Metric (Measure)	Job-To-Be-Done						
CO#105	10	Maximize	Frequency/ Possibility	(To) avoid malfunction of alarm	1	0	1	0	0	0
CO#106	10	Maximize	Frequency/ Possibility	(To) assign to downgraded room	1	0	1	0	0	0
CO#107	10	Minimize	Energy/Effort	(To) check out early	1	0	1	0	0	0
CO#108	10	Maximize	Frequency/ Possibility	(To) avoid odor issue	2	0	2	1	0	2
CO#109	11	Minimize	Energy/Effort	(To) enquiry to manager	1	0	1	0	0	0
CO#110	11	Maximize	Frequency/ Possibility	(To) avoid billing mistake (due to overcharging, ignored payment by cash, overcharging tips)	9	0	9	6	0	12 *
CO#111	11	Maximize	Frequency/ Possibility	(To) be notified for late check-out	3	0	3	2	0	4
CO#112	11	Minimize	Cost	(To) pay for early check-in	1	0	1	0	0	0
CO#113	12	Maximize	Frequency/ Possibility	(To) get a proper service at front desk	2	1	1	1	5	1
CO#114	12	Maximize	Frequency/ Possibility	(To) avoid billing mistake (due to exchange rate, calculation mistake, ... )	6	2	4	4	3.33333333	4.666666667
CO#115	12	Maximize	Frequency/ Possibility	(To) return deposit properly	7	3	4	5	4.28571429	5.714285714
CO#116	12	Minimize	Energy/Effort	(To) communicate with manager	1	1	0	0	10	0

\* : outcomes having high business opportunities.



## References

1. Fritsch, M.; Mueller, P. Effects of new business formation on regional development over time. *Reg. Stud.* **2004**, *38*, 961–975. [[CrossRef](#)]
2. Aaker, D.A.; Adler, D.A. *Developing Business Strategies*; John Wiley & Sons: New York, NY, USA, 2001.
3. Mun, C.; Kim, Y.; Yoo, D.; Yoon, S.; Hyun, H.; Raghavan, N.; Park, H. Discovering business diversification opportunities using patent information and open innovation cases. *Technol. Forecast. Soc. Chang.* **2019**, *139*, 144–154. [[CrossRef](#)]
4. Park, H.; Yoon, J. A chance discovery-based approach for new product-service system (PSS) concepts. *Serv. Bus.* **2015**, *9*, 115–135. [[CrossRef](#)]
5. Ulwick, A. *What Customers Want*; McGraw-Hill Professional Publishing: New York, NY, USA, 2005.
6. Bigorra, A.M.; Isaksson, O.; Karlberg, M. Semi-autonomous methodology to validate and update customer needs database through text data analytics. *Int. J. Inf. Manag.* **2020**, *52*, 102073.
7. Xu, X.; Wang, X.; Li, Y.; Haghghi, M. Business intelligence in online customer textual reviews: Understanding consumer perceptions and influential factors. *Int. J. Inf. Manag.* **2017**, *37*, 673–683. [[CrossRef](#)]
8. Ulwick, A.W. *What Is Outcome-Driven Innovation® (ODI)? White Paper*; Strategyn Inc.: Denver, CO, USA, 2009.
9. Ulwick, A.W. *Jobs to Be Done: Theory to Practice*; Idea Bite Press: New York, NY, USA, 2016.
10. Christensen, C.M.; Hall, T.; Dillon, K.; Duncan, D.S. Know your customers' jobs to be done. *Harv. Bus. Rev.* **2016**, *94*, 54–62.
11. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
12. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. *The Stanford CoreNLP Natural Language Processing Toolkit, Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 22–27 June 2014*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 55–60.
13. Honnibal, M.; Montani, I. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To Appear* **2017**, *7*.
14. Christensen, C.M.; Anthony, S.D.; Berstell, G.; Nitterhouse, D.J.M.S.M.R. Finding the right job for your product. *MIT Sloan Manag. Rev.* **2007**, *48*, 38.
15. Johnson, M.W.; Christensen, C.M.; Kagermann, H. Reinventing your business model. *Harv. Bus. Rev.* **2008**, *86*, 57–68.
16. Bettencourt, L. *Service Innovation: How to go from Customer Needs to Breakthrough Services*; McGraw Hill Professional: New York, NY, USA, 2010.
17. Lim, J.; Choi, S.; Lim, C.; Kim, K. SAO-based semantic mining of patents for semi-automatic construction of a customer job map. *Sustainability* **2017**, *9*, 1386. [[CrossRef](#)]
18. Joung, J.; Jung, K.; Ko, S.; Kim, K. Customer complaints analysis using text mining and outcome-driven innovation method for market-oriented product development. *Sustainability* **2019**, *11*, 40. [[CrossRef](#)]
19. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J.J. Efficient estimation of word representations in vector space. In Proceedings of the International Conference on Learning Representations, Scottsdale, AZ, USA, 2–4 May 2013.
20. Park, H.; Yoon, J.; Kim, K. Identifying patent infringement using SAO based semantic technological similarities. *Scientometrics* **2012**, *90*, 515–529. [[CrossRef](#)]
21. Elman, J. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [[CrossRef](#)]
22. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **1998**, *6*, 107–116. [[CrossRef](#)]
23. Pennington, J.; Socher, R.; Manning, C. *Glove: Global Vectors for Word Representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1532–1543.
24. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T.J.T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [[CrossRef](#)]
25. LeCun, Y.A.; Bottou, L.; Orr, G.B.; Müller, K.-R. Efficient backprop. In *Neural networks: Tricks of the Trade*; Springer Link: Berlin, Germany, 2012; pp. 9–48.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems, Proceedings of The Thirty-First Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Biosonomy: Kassel, Germany, 2017; pp. 5998–6008.
27. Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier chains for multi-label classification. *J. Mach. Learn.* **2011**, *85*, 333. [[CrossRef](#)]
28. Zhang, M.-L.; Zhou, Z.-H.J.G. A k-nearest neighbor based algorithm for multi-label classification. In Proceedings of the 2005 IEEE International Conference on Granular Computing, Beijing, China, 25–27 July 2005; Volume 5, pp. 718–721.
29. Luaces, O.; Díez, J.; Barranquero, J.; del Coz, J.J.; Bahamonde, A. Binary relevance efficacy for multilabel classification. *Prog. Artif. Intell.* **2012**, *1*, 303–313. [[CrossRef](#)]
30. Nam, J.; Kim, J.; Mencía, E.L.; Gurevych, I.; Fürnkranz, J. *Large-Scale Multi-Label Text Classification—Revisiting Neural Networks, Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, France, 14–18 September 2014*; Springer: Berlin, Germany, 2014; pp. 437–452.
31. Rifkin, R.; Klautau, A. In defense of one-vs-all classification. *J. Mach. Learn. Res.* **2004**, *5*, 101–141.

32. Park, H.; Yoon, J.; Kim, K. Using function-based patent analysis to identify potential application areas of technology for technology transfer. *Exp. Syst. Appl.* **2013**, *40*, 5260–5265. [[CrossRef](#)]
33. Park, H.; Ree, J.J.; Kim, K. Identification of promising patents for technology transfers using TRIZ evolution trends. *Exp. Syst. Appl.* **2013**, *40*, 736–743. [[CrossRef](#)]
34. Park, H.; Yoon, J.; Kim, K. Identification and evaluation of corporations for merger and acquisition strategies using patent information and text mining. *Scientometrics* **2013**, *97*, 883–909. [[CrossRef](#)]
35. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. *How to Fine-Tune BERT for Text Classification? Proceedings of the China National Conference on Chinese Computational Linguistics, Kunming, China, 18–20 October 2019*; Springer: Berlin, Germany, 2019; pp. 194–206. [[CrossRef](#)]
36. Hu, Y.-H.; Chen, K. Predicting hotel review helpfulness: The impact of review visibility, and interaction between hotel stars and review ratings. *Int. J. Inf. Manag.* **2016**, *36*, 929–944. [[CrossRef](#)]
37. Neirotti, P.; Raguseo, E.; Paolucci, E. Are customers' reviews creating value in the hospitality industry? Exploring the moderating effects of market positioning. *Int. J. Inf. Manag.* **2016**, *36*, 1133–1143. [[CrossRef](#)]
38. Ban, H.-J.; Choi, H.; Choi, E.-K.; Lee, S.; Kim, H.-S. Investigating key attributes in experience and satisfaction of hotel customer using online review data. *Sustainability* **2019**, *11*, 6570. [[CrossRef](#)]
39. Kim, N.; Lee, S.; Jo, S.; Kim, J. Multi-dimensional keyword search and analysis of hotel review data using multi-dimensional text cubes. *J. Inf. Technol. Archit.* **2014**, *11*, 63–73.
40. Hwang, S.-Y.; Lai, C.-Y.; Jiang, J.-J.; Chang, S. The identification of noteworthy hotel reviews for hotel management. *Pac. Asia J. Assoc. Inf. Syst.* **2014**, *6*, 1. [[CrossRef](#)]
41. Yuan, C.; Huang, Y.-C. Personalized sentence generation using generative adversarial networks with author-specific word usage. *arXiv* **2019**, arXiv:1904.09442. [[CrossRef](#)]