Natural Learning Processing based on Machine Learning Model for automatic analysis of Online Reviews related to Hotels and Resorts

Bogdan-Ştefan POSEDARU, Tiberiu-Marian GEORGESCU, Florin-Valeriu PANTELIMON
The Bucharest University of Economic Studies
posedarubogdan10@stud.ase.ro, tiberiugeorgescu@ase.ro, pantelimonflorin15@stud.ase.ro

This article describes the development and implementation of a natural language processing (NLP) model based on machine learning (ML) for automatic analysis of customers' reviews on hotels and resorts written in English. The model performs named entity recognition (NER), relation extraction (RE) as well as sentiment analysis (SA). The performance indicators validate the model, as we obtained an F1 score of 0.79 for ER and 0.61 for RE. Our results prove to be remarkable compared to other models that use similar techniques and technologies. Furthermore, we developed a web application which allows users to benefit from our model to automatically analyze customers' reviews about hotels and resorts.

Keywords: *Natural Language Processing, Machine Learning, Entity Recognition, Relation Extraction, Sentiment Analysis*

1 Introduction

The growth of internet in the last decade has had a massive impact in the hospitality industry, as in most industries. Nowadays, both customers and service providers invest plenty of time in reading and analyzing online information. When choosing to book a product or a service in the hospitality domain, a critical role is played by consumers' online reviews [1], [2]. Study [3] considers that more than 75% of people are taking online reviews into account when booking a hotel. Therefore, the companies from hospitality industry need to carefully monitor customers' reviews, as they are an important decision factor for their clients [4].

In this context, the volume of reviews increased drastically and it became more and more time consuming to analyze them manually. This study aims to automate the hotel review analysis, by using natural language processing (NLP) based on machine learning (ML). We developed a model which is able to perform named entity recognition (NER), relation extraction (RE) and sentiment analysis (SA) for online reviews regarding hotels and resorts written in English. Based on a list of reviews, the model identifies the main entities and relations, the sentiment for each entity and the general sentiment for the whole document analyzed. Furthermore, we created a web application available at https://hotelinsights.live/ which serves as a portal to the model. By this manner, users can benefit from the model's capabilities when choosing a hotel for their holiday.

Related work

The tremendous growth of data related to tourism led to the necessity of new tools to manage it. Many web mining techniques are used for automatically extracting useful insights from web content related to hospitality industry. In the 2000s, the main techniques used were business intelligence for structured and semi-structured content and web analytics for unstructured data [5]. The Web 2.0 brought to the table big volumes of data, especially user-generated content. The existing techniques weren't efficient in extracting insights from large volumes of data, therefore, new ones were developed, in particular NLP [6].

In the early ages of the NLP, rule-based techniques were used. Article [7] describes a rule-based NLP model for the analysis of online reviews, while paper [8] explores ways to implement rules-based NLP for documents about the hospitality industry.

However, such techniques are limited to predefined sets of rules. Since the statistical revolution, numerous algorithms were developed which are capable of discovering various patterns outside the predefined sets of rules. Subsequently, the ML paradigm has become increasingly popular, and in the last decade, ML-based NLP is considered by far the main approach. Many articles discuss NLP based on ML models for the hospitality industry, such as [9], [10].

Several studies that aim to automatically extract relevant information from tourism reviews have been identified. Article [11] uses NER on travel texts and the paper [12] studies the application of NER and RE to achieve the proposed goal. The article [13] discusses the SA techniques applied on hotel reviews.

Our work

The main purpose of our work consists in developing a NLP model which automatically analyses reviews related to hotels and resorts. In this regard, we studied related papers and identified the main methods, techniques and instruments used.

We started by developing a domain ontology specific to the hotel industry field. Section 2 describes in detail the process of developing the ontology. Subsequently, the ontology was used as the structure for the NLP based on ML model. Section 2 also presents the process of designing and training the ML model. We developed custom-made solutions to automatically download training data, consisting in online reviews from popular hotel booking platforms, such as Tripadvisor (https://www.tripadvisor.com) or Booking (https://www.booking.com/).

Section 3 presents the design and implementation of an application for automatic text analysis of hotel reviews. The application architecture is described in detail, the model described in the previous section being the main component. The application is available online at https://hotelinsights.live/, so that any interested individual can use it. Performance evaluation was conducted regularly, based on which we adjusted the model and the training process.

Section 4 describes a detailed performance analysis of the model. We calculated the values of the F1 score, precision and recall indicators and examined the correlation matrix. The remarkable results validate our model and prove its robustness. A comparison of our model's performances to the ones obtained by other projects was conducted in order to highlight our results in relation to other studies. Section 5 concludes our work and discusses the main contributions, future work as well as the limitations.

2 The development of the NLP based on ML model

This section describes the process of developing a specialized NLP based on ML model for the hotel industry. The model can perform the following three tasks: NER, RE and SA. Based on the documents received, the model identifies the relevant entities and the relations between them, the sentiment associated with each entity, as well as the general sentiment associated with each class of entities.

In order to build the model, the first step was to create a domain ontology. Once completed, it served as the structure of the model.

2.1. The ontology development process

An extensive study was done in order to understand the particularities of the documents of interest. In order to develop a robust ontology we applied two different approaches: (1) we studied the state-of-the-art related to similar ontologies and (2) we selected over 500 reviews available on platforms such as Tripadvisor or, from which we identified the tokens relevant for our purpose. The tokens, consisting in sequences of characters such as one word or more words combined, were considered potential entities for our ontology.

The study of relevant papers

Various articles discuss the process of building ontologies for the tourism field. Article [14] uses semantic web technologies to develop an ontology whose aim is to enable users to acquire useful information regarding their trip. Paper [15] describes a framework for tourists knowledge representation based on online reviews. Other studies focus more specifically and describe the development of ontologies for tourism in particular regions [16], [17]. Our study is concentrated particularly to the hotels and resorts, therefore we focused on the articles which describe ontologies designed for hotels services, such as [18] or [19]. Although paper [19] was partly written in a language unknown to us (Korean), we managed to comprehend its content by using translating applications. Since our work is designed to perform not only NER and RE, but also SA from online reviews, we studied various articles which discuss models suitable in this regard, such as precum [20] or [21].

Our ontology

A robust domain-ontology was essential in order to develop the NLP model. After we studied the related work, we explored online reviews and extracted the relevant tokens, which lately were implemented as entities in the model's dictionary. The ontology's structure was designed in such a manner to be suitable for the NLP based on ML model.

Our ontology consists of 6 classes and 14 subclasses. The 6 classes are connected to each other by 14 relationships, as shown in Figure 1. An extended representation of the ontology, which contains the subclasses as well, can be observed in Annex 1. In order to design the ontology, we used WebProtégé (https://webprotege.stanford.edu/). For the graphical representation, we used WebVOWL, a web application for visualization of ontologies available at (http://vowl.visualdataweb.org/webvowl.html).

In order to ensure the development of an optimal ontology, various strategies were used. In the initial stage, we decided that each author will develop its own version of the ontology and then compare them to each other. Out of the three ontologies, two were selected as candidates. We tested several versions of the ontology, in order to identify the optimal one for the NLP model described later in this paper. During this process, we eliminated the redundant classes or the ones with low relevance for our work. As an example, initially we considered the class *Hotel's Specifications*. However, we decided to include most of its entities into the class *Amenities* and to eliminate the others, as they were increasing the model's complexity. Also, several classes were merged. Instead of having two classes named *Incident* and *Vacation*, we decided to create a new one – *Experience* and included those two as subclasses of the class *Experience*.

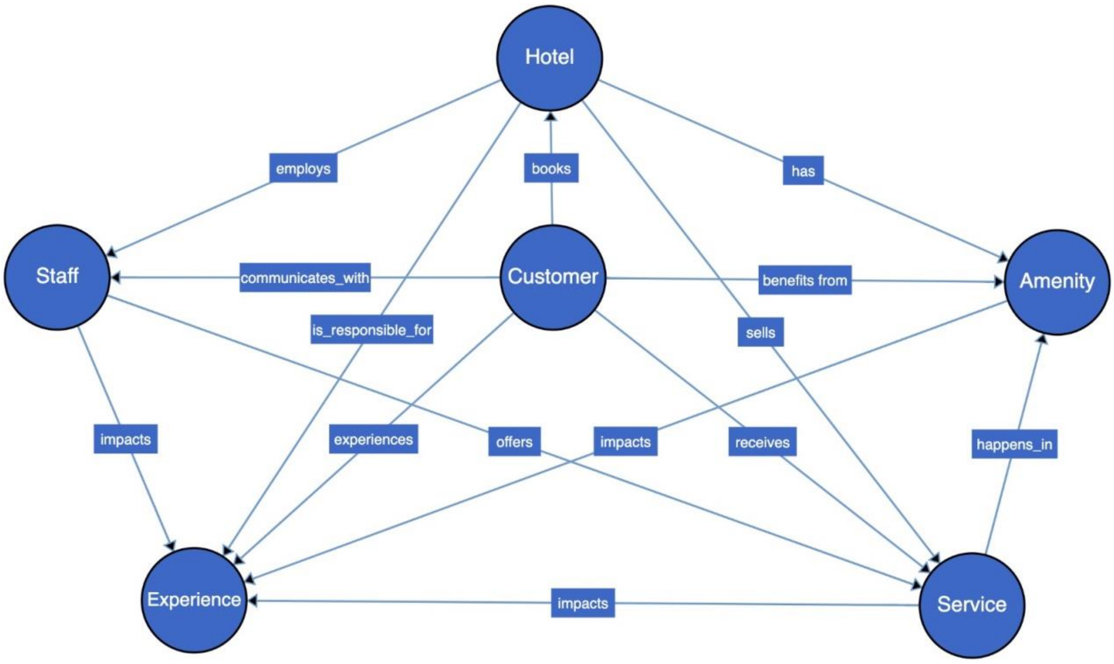


Fig. 1. The ontology specialized in the hotel field

Table 1 describes both the classes and their subclasses and table 2 illustrates the relationships between classes. Four of the classes are divided into subclasses. We considered that the complexity of the model based only on the main six classes was relatively low; therefore, it could bring results with better accuracy. However, by including the subclasses the model can provide more detailed information. After conducting various performance tests, we chose to keep the subclasses as part of our model's structure.

Table 1. The list of classes and subclasses

No.	Class	Subclass
1	Amenity	FoodDrinks_Amenity
		Leisure Amenity
		General_Amenity
		Room_Amenity
2	Experience	Incident
		Vacation
3	Staff	FoodDrinks_Staff
		General_Staff
		Leisure_Staff
		Room_Staff
4	Service	FoodDrinks_Service
		General_Service
		Leisure_Service
		Room_Service
5	Hotel	-
6	Customer	-

Table 2 illustrates the relations between classes. Once we implemented the NLP based on ML model, we noticed that its performance was very low for some of the relations; therefore we adjusted them to the current variation.

Table 2 The relations between classes

No.	Relation	Parent	Child
1	benefits_from	Customer	Amenity
2	books	Customer	Hotel
3	communicates_with	Customer	Staff
4	employs	Hotel	Staff
5	experiences	Customer	Experience
6	happens_in	Service	Amenity
7	has	Hotel	Amenity
8	impacts	Amenity	Experience
9	impacts	Staff	Experience
10	impacts	Service	Experience
11	is_responsible_for	Hotel	Experience
12	offers	Staff	Service
13	receives	Customer	Service
14	sells	Hotel	Service

2.2. Choosing the proper technologies and techniques

Paper [22] presents a detailed comparison of the main NLP solutions based on ML, and paper [23] describes the current state-of-the-art on NER based on ML. Considering the studies presented, as well as the research objectives of this paper, we chose to use the services Knowledge Studio(https://www.ibm.com/cloud/watson-knowledge-studio) and Natural Language Understanding (https://www.ibm.com/cloud/watson-natural-language-understanding) from IBM Watson (https://www.ibm.com/watson). Regarding the ML approach, supervised learning was preferred, as in recent years studies show that it provides the best performance for NLP [24].

Knowledge Studio was used to define the model's structure, to configure it according to the domain particularities and to train the model, while Natural Language Understanding was used to deploy the model and served as a gateway to the analysis results provided by our model.

We implemented the ontology in Knowledge Studio as follows: we defined the classes, their subtypes and the relations between classes. Each class contained a dictionary which consisted in a list of entities. Annex 2 illustrates the dictionary afferent for the class Amenity.

The dictionary was used as a rule-based model only in the training process, accelerating the document annotation process. It contains of approximately 1000 entities, plus their respective surface forms. For example, in annex 2, the entity "a la carte restaurant" has four possible surface forms.

2.3. Training the model

Supervised ML models require the provision of correct result sets on which the machine builds the *ground truth*. In the training process, we annotated the documents for two tasks NER and RE. For NER, the annotation process consisted in identifying and labeling the relevant tokens and assigning each one to the proper class. In order to automate the process, the rule-based model was used as it automatically identified the tokens defined in the dictionary. However, much attention from the annotator was required, as many of the tokens weren't correctly labeled by the dictionary. Figure 2 shows a screenshot taken during the training process for NER.

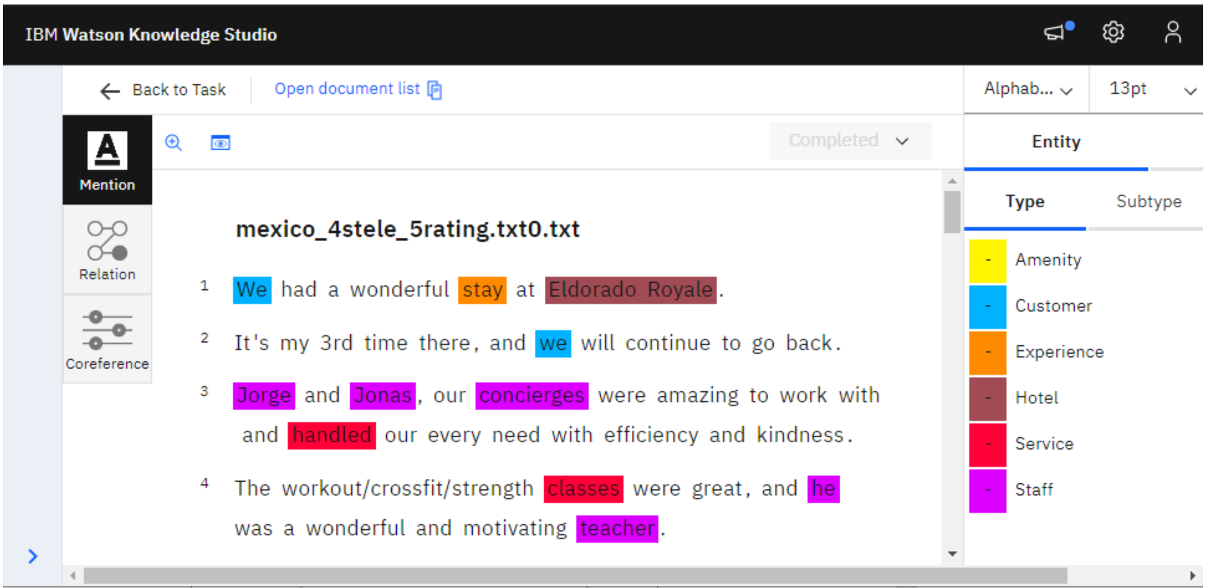


Fig. 2. Training the model for the NER task

As can be observed in figure above, in the first sentence the token *We* was annotated as part of the class *Customer*, the token *stay* as entity of the class *Experience* and the token *Eldorado Royale* as type *Hotel*. In addition to classes, subclasses were also annotated. Figure 3 illustrates a screenshot taken during the annotation process; as can be observed, the token *stay* was labeled as entity of the subclass *Vacation*.

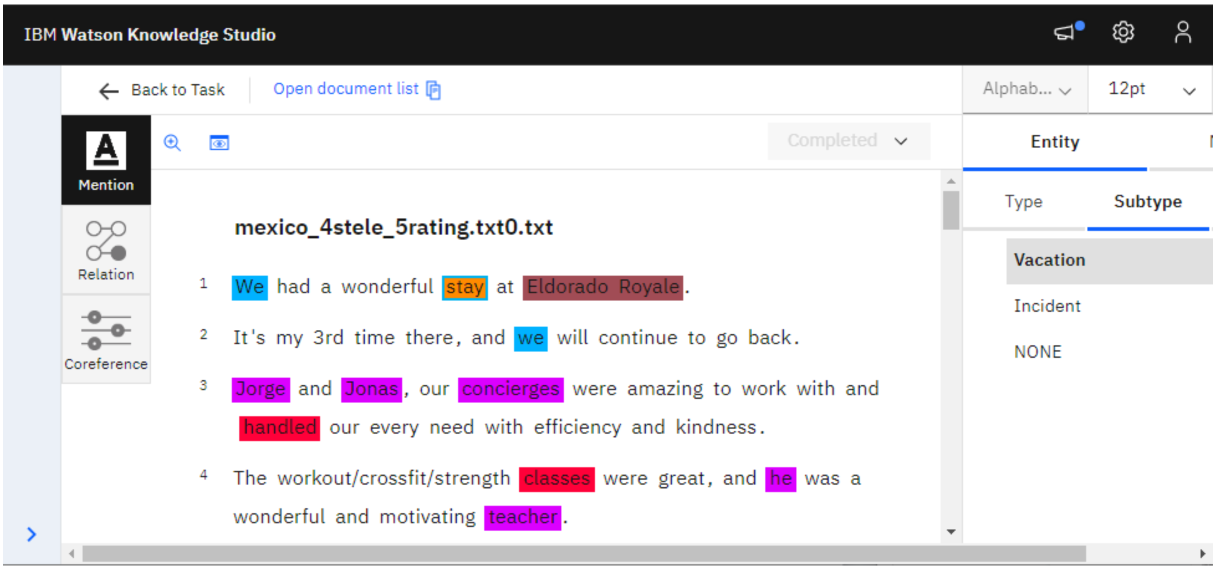


Fig. 3. The annotation of a subclass

The training of the RE functionality is performed by marking the relations between the identified tokens. Figure 4 is a screenshot taken during the annotation process. Within it, the annotated relations are: (1) *Customers experiences stay*, (2) *Customer books Eldorado Royale*, (3) *Eldorado Royale is responsible for stay*.

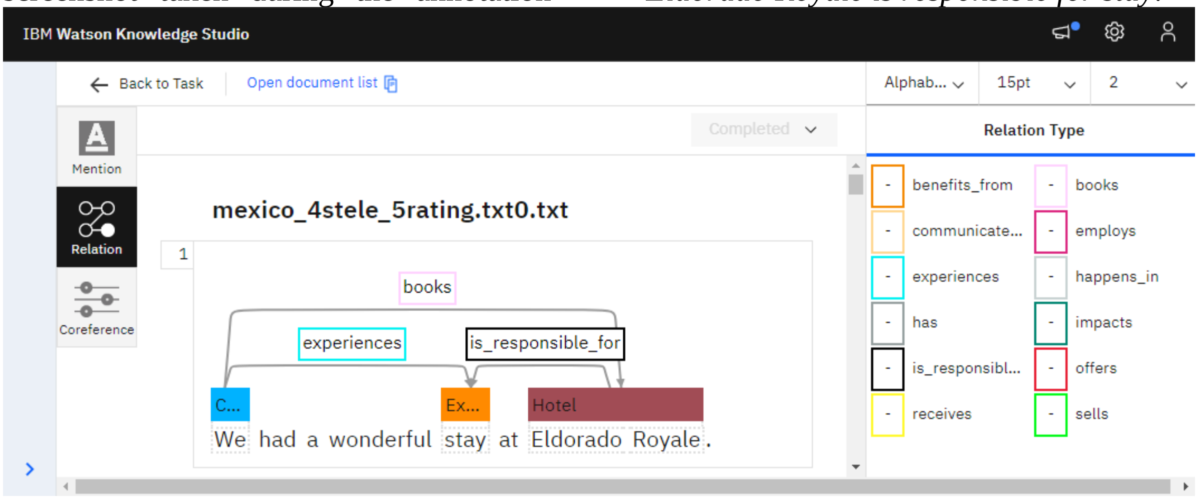


Fig. 4. Training the model for the RE task

For training, we downloaded reviews available online. We used documents totalizing approximately 100,000 words. In order to speed up the process, we developed custom-made scraping solutions to automatically download and structure the relevant data.

According to Watson's methodology (https://cloud.ibm.com/docs/watson-knowledge-studio-data?topic=watson-knowledge-studio-data-documents-for-annotation), for best performances it is recommended to use training documents of maximum 1000 words each. Therefore, we built a script that splits the training data in documents of at most 1000 words each.

The human annotation defined the model's *ground truth*. It was essential to make sure that the annotations were consistent. For our work, we decided that only one author (B.-Ş.P.) to perform annotation. Furthermore, the annotations were checked and amended by all three authors together before validating. During the process, periodic performance tests were conducted based on which we adjusted the annotations. A detailed description of the performance tests is presented in section 4.

2.4. Deploying the model

In order to deploy the model, we used IBM Watson Natural Language Understanding (NLU). Watson's Natural Language Understanding service is a collection of content analysis features that can extract semantic information from the desired input. The data can be either text, public URLs or HTML content, the service benefiting from sophisticated NLP techniques to quickly obtain features extracted from the content, such as: NER, keywords identification, RE, SA, emotion extraction, semantic roles extraction, etc.

In order to use the developed Model for Natural Language Processing analysis, it must be deployed from Knowledge Studio (where the model was trained) to Natural Language Understanding Service, communicating through a RESTful API.

Following various configurations, a private access key (API key) is offered, which requires secure storage and which must be attached to every API call to the service. In order to ensure the security of such sensitive data, a robust solution was designed, its implementation being presented in greater detail in subsection 3.1 of the present paper.

The Natural Language Understanding API receives all requests and sends all responses using the JSON (JavaScript Object Notation) standard, which facilitates data manipulation at all the levels of our web application, described in section 3 of the present paper.

Following the analysis of the content, the NER task of the NLU service sends a response consisting of a list of identified entities and their respective properties (figure 5).

```
entities: [{
 type: "Amenity",
 text: "Spa",
 mentions: [
   {
     text: "Spa",
     location: [332, 335],
     confidence: 0.998892,
   },
 ],
 disambiguation: {
   subtype:
["Leisure_Amenity"],
 },
 confidence: 0.998892,
}]
```

Fig. 5. Watson NLU service response for the NER functionality

We can notice from the above code snippet many properties of the identified entity, such as:

- the class and subclass to which it belongs;
- the entity text;
- location of the entity in the text (both starting and ending indices are provided);
- the associated confidence score: the closer the score is to 1, the higher the confidence label.

The RE functionality, which is closely related to the above described process of NER, provides valuable insights in relation to the present connections at a sentence level.

The IBM Watson Natural Language Understanding service sends a response consisting of a complete list of identified relationships, with their associated properties, such as linked entities, relations types, confidence scores, etc.

Regarding the SA functionality, the NLU service assesses the sentiment for the document in its entirety and also performs an individual sentiment evaluation for every entity identified. An example of the response provided by the relationship extraction functionality can be consulted in figure 6.

```
{
 type: "Amenity",
 text: "Spa",
 sentiment: {
   score: 0.972212,
   label: "positive",
 },
 disambiguation: {
   subtype:
["Leisure_Amenity"],
 },
 count: 1,
}
```

Fig. 6. Watson NLU service response for the SA functionality

From the above code snippet, it is noticeable that the *Amenity* identified in the given text as *Spa*, belonging to *Leisure Amenity* subclass, had attributed a positive sentiment, with an almost perfect score of 0.97.
The sentiment score provided by the service can take any value between -1 and 1, where a score of 0 represents a completely neutral sentiment towards the subject. The closer a value gets to -1, the more negative the sentiment it denotes, and the closer the value is to 1, the stronger the positive sentiment related to the entity.

3. Implementation of an application for automatic text analysis of hotel reviews

In order to demonstrate the capabilities of the NLP model based on ML, a web application was designed for the cognitive analysis of documents specific to the hospitality industry.

3.1. The Application's Architecture

During the documentation and development phases, the question arose as to whether the web application should also include a server component, or whether a direct Client-NLP Model communication should suffice. It was finally decided to implement the stand-alone server component, acting as a bridge between the interface and the IBM Watson API Service, for the following reasons:

- The stand-alone NodeJS Server allows to securely store sensitive data, such as API keys, by using environment variables stored in special *.env* files that are inaccessible to the external environment;
- The stand-alone NodeJS Server allows to execute preprocessing functions on input data, such as extracting text content from compatible files. This is an important aspect in terms of IBM Cloud usage savings;

Taking the above into account, the complete architecture of the web application can be consulted in the figure below:

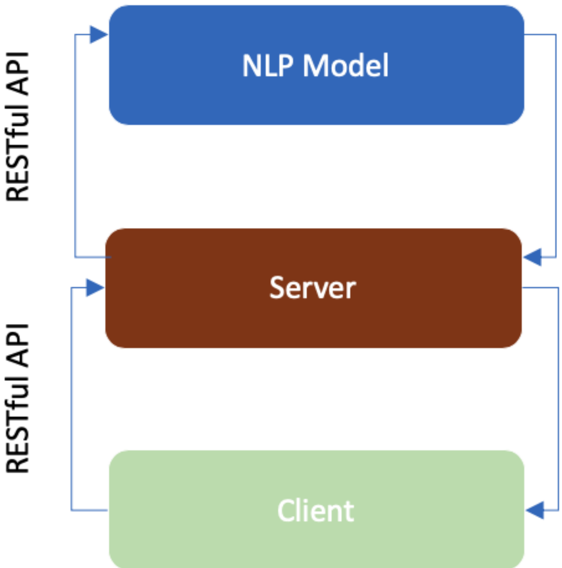


Fig. 7. Web Application Architecture

3.2. Technologies used

Considering the fact that one of the main requirements of the application was for it to be readily available for any potential user, we decided to develop it as a web application, published and accessible from a public domain (http://hotelinsights.live).
In addition to satisfying the basic functionalities of the app, many other factors were taken into account when

deciding on the technology stack, such as general speed and performance, Client-Server communication stability or session persistence.

In order to develop the web application, a solution was chosen that consists entirely of Javascript libraries and frameworks, both for the client-side (ReactJS) and for the server-side (NodeJS). Developing both components using the same programming language ensures the possibility of fast services integration or code reusability. The chosen technology stack presents many advantages, such as full compliance with all the functional requirements of the application or the high availability and reduced costs related to deploying solutions.

The complete list of technologies used includes: Javascript, NodeJS, ExpressJs, ReactJS, Create-React-App CLI, Jest, Axios, GoJS, Textract, Dotenv, Git, Heroku and Postman.

Analyzing the stack of technologies presented above, we observe a complete synergy between all the application's components, which is possible only due to the use of a single programming language for all architectural layers.

3.3. An overview of an application scenario

This subsection describes a typical use case scenario, with all the associated steps from loading the document to seeing the analysis results, structured in an intuitive way.

Accessing and using the web application is very simple both from a laptop and a mobile device, the only requirement being a modern web browser, such as Google Chrome or Mozilla Firefox.

When the application loads, it presents the user with a welcome screen and displays a series of three buttons representing options of entering data, by uploading a document from the user's personal computer, using sample hotel reviews documents or entering text directly in the web app. In the following paragraphs, we describe the file upload method of entering data, component that also allows drag-and-drop technique, improving the general user experience. After the desired document is selected, the user confirms the operation, the data is sent to the stand-alone server and a *loading* component is presented, acting as a buffer screen between the input and the output components.

In case the stand-alone server sends back an error-free response, the user is redirected to the results page, from where he can visualize the analysis done by the NLP based on ML model. The results are structured into three tabs, one for NER task, one for RE and the final for SA.

For visualizing the NER task results, the application highlights the recognized entities in the text, using custom colors according to its corresponding class.

Results from the RE task are presented as a Force Directed Graph consisting of nodes – representing entities, and links – representing identified relations, where the user can rearrange these components, for a better overview of complex sentences, with a high number of elements identified (figure 8).

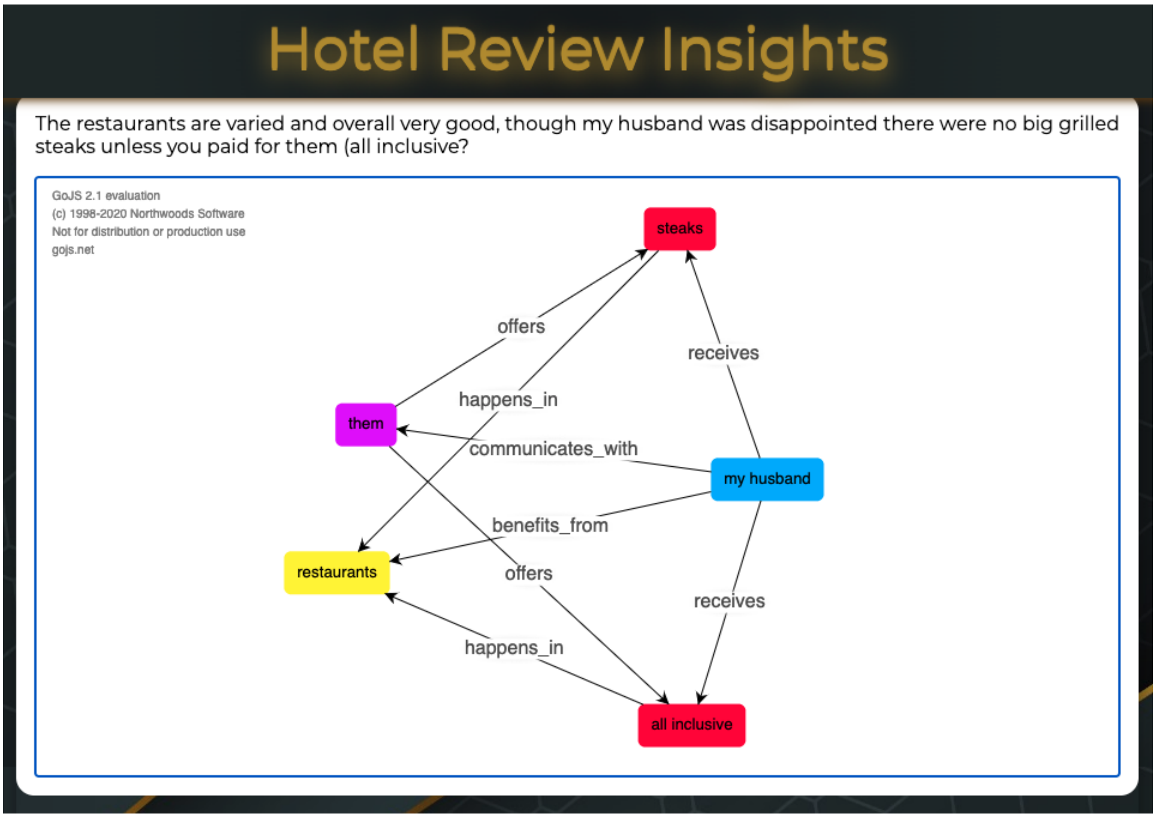


Fig. 8. The graphical representation of the RE results

Regarding the SA results, the web application presents an entity-level analysis, aggregated by class, as follows: *Experience*, *Amenity*, *Customer*, *Hotel*, *Service* and *Staff*. Each class' sentiment score is graphically represented by the help of a red circle that is partially or completely covered with a green trail: the longer the green trail, the better the sentiment score, as can be seen in figure 9. The user can also get a more detailed look on one class' sentiment score by clicking on the corresponding button and consulting the table of identified entities and associated sentiment score.

Fig. 9. The graphical representation of the SA results

A special role in the analysis is played by the *Incident* subclass, belonging to the *Experience* class. The web application builds an *Incident Report* based on recognized entities belonging to this subclass and assigns a score representing the probability of an incident being described in the provided document.

Taking into account the previously described functionalities, the user can benefit from the NLP model analysis capabilities in an intuitive web application, and can extract invaluable insights from documents related to the hospitality industry.

4 Results

This section describes the performances evaluation process and the model's results. F1 score was calculated for the NER and RE functionalities and the results validate the model. We describe the values of the indicators, for each entity type and relation type as well as overall. Besides these, the confusion matrices are presented and analyzed, in order to better understand the model's performances and identify new strategies for improvement. Also, we compare our results with other similar projects and discuss our model's advantages as well as its limitations.

4.1. Methodology

In order to evaluate the performance of the model, the methodology developed by IBM, specific for NLP based on ML models, was used (https://cloud.ibm.com/docs/watson-knowledge-studio?topic=watson-knowledge-studio-evaluate-ml).

The documents were split into three categories: training sets, test sets and blind sets. The training sets which were used to define the *ground truth* represented 70% of the total documents. The test sets were documents annotated by human, which weren't included in the *ground truth*, but were kept aside to be used in the performance evaluation process. The test sets represented 23%. For the blind sets, we kept 7% of the documents aside not only from the *ground truth*, but also from the annotator (B.-Ş.P). They were managed by the other two authors of this paper. Knowing all the documents, the annotator could have influenced the model so that it would have adapted particularly for the test sets and not in general, as desired. By this manner, we reduced the bias tendency of the annotator as much as possible.

We took into account the F1 score, precision and recall, which are established indicators used for the supervised ML evaluation process [25] The value of each indicator is between [0,1], the bigger the value, the better the result.

The precision indicator illustrates how accurate are the model's annotations in comparison to the human's annotations, considered as *ground truth*. The precision's formula is:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}},$$

where *true positive* consists in tokens annotated as relevant in a correct manner, contrary to *false positive*, which represent tokens incorrectly labeled as relevant. The worst value for the precision indicator is 0 and the best is 1. If the value of precision is 1, it means that the model marked correctly all the selected tokens.

Recall measures how many mentions were actually annotated with the proper labels.

$$\text{(2) Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}},$$

where *false negative* consists in the mentions which the model fails to recognize. The highest value (of 1) indicates that the model identifies all the relevant mentions, on the contrary, a recall of 0 means that the model wasn't able to identify any entity.

The F1 score is calculated based on precision and recall, as a harmonic mean of the two indicators, showed in the formula below. In order to validate the model, the F1 score value has to be higher than 0.5.

$$\text{(3)} \qquad \text{F1 score} = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}.$$

Usually, a low precision indicates that the machine generates incorrect annotations (https://cloud.ibm.com/docs/watson-knowledge-studio?topic=watson-knowledge-studio-evaluate-ml#evaluate-mllowp). A low recall illustrates that the machine is not able to identify and annotate the right mentions (https://cloud.ibm.com/docs/watson-knowledge-studio?topic=watson-knowledge-studio-evaluate-ml#evaluate-mllowr).

4.2. The performances of the model

The F1 score of the current version of the model is 0.79 for NER and 0.61 for RE. Figure 10 illustrate the evolution of the model, as well as the values for the Precision and Recall for both NER and RE.

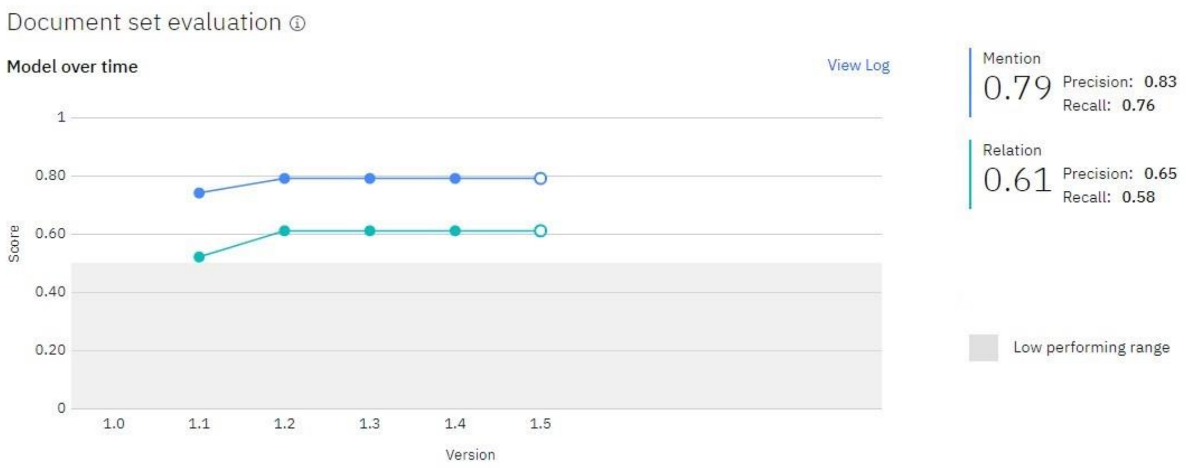


Fig. 10. The evolution of the model's performances for NER and RE

As can be observed, the F1 score has known a significant improvement from version 1.1 to version 1.2 both for NER and RE. For version 1.1 documents totaling around 50,000 words were annotated, compared to version 1.2 where their number was approximately 100,000. The performance tests corresponding to 1.3, 1.4 and 1.5 are performed with the same corpus as in version 1.2, but with different test sets. Since all the last 4 versions have very similar F1 score, we consider that the results weren't casual and that the tests present sufficient consistency.

The model's performances for NER

The F1 score for NER of 0.79 indicates good performances. The precision indicator is higher than the recall, therefore we plan to perform further training session which may improve the model. Moreover, as can be observed in figure 11, the recall is lower than the precision for each entity class, which accentuates our assumption that further training may raise the performances.

Entity Types	F1	Precision	Recall	% of Total Annotations
Amenity	0.74	0.8	0.7	30% (633/2114)
Customer	0.89	0.9	0.87	18% (378/2114)
Experience	0.69	0.71	0.67	7% (139/2114)
Hotel	0.75	0.89	0.66	7% (156/2114)
Service	0.68	0.72	0.64	28% (619/2114)
Staff	0.7	0.75	0.66	9% (189/2114)

Fig. 11. The performance indicators for each class

The F1 score is over 0.68 for each class type. The best performances are obtained for the class Customer (0.89) and the biggest percent of annotation for the classes Amenity (30%) and Service (28%). An essential tool for improving the model is the confusion matrix. Based on it, we can get a better understanding about the model's flaws.

Based on it, we can get a better understanding of the classes for which the model has trouble annotating properly, as it offers details about the reasons for the wrong labels. The analysis of the matrix can be done from two perspectives: (1) the confusion between each type of class and the others and (2) the confusion between each type of class and tokens which are not relevant for our model marked in table 3 with the label *Other tokens*. The table below illustrates the confusion matrix for NER in absolute values.

Table 3. The confusion matrix for NER

Entity Types	**Amenity**	**Customer**	**Experience**	**Hotel**	**Other tokens**	**Service**	**Staff**	**Total**
Amenity	462	0	1	3	137	21	9	633
Customer	0	329	0	0	47	2	0	378
Experience	2	0	88	2	45	2	0	139
Hotel	4	0	0	92	40	11	9	156
Other tokens	15	30	15	0	7424	41	17	7542
Service	13	0	7	1	169	423	6	619
Staff	3	0	0	0	66	3	117	189
Total	**499**	**359**	**111**	**98**	**7928**	**503**	**158**	**9656**

The model's performances for RE

The task of RE proves to be more complex, since our model contains a relatively high number of relation types (14). The F1 score of 0.61 validates the model with a precision of 0.65 and a recall of 0.58. As in the case of NER, the recall is lower than the precision, thus further annotations could improve the performances. Figure 12 illustrates the indicators for each relation type.

	Relation Types	F1	Precision	Recall	% of Total Annotations
	benefits_from	0.64	0.7	0.59	16% (743/4579)
	books	0.64	0.77	0.54	6% (254/4579)
	communicates_...	0.58	0.63	0.54	6% (261/4579)
	employs	0.57	0.74	0.47	2% (75/4579)
	experiences	0.55	0.56	0.54	6% (271/4579)
⚠	**happens_in**	**0.4**	**0.39**	**0.42**	**16% (747/4579)**
	has	0.51	0.67	0.41	5% (216/4579)
⚠	**impacts**	**0.49**	**0.52**	**0.47**	**9% (401/4579)**
⚠	**is_responsible_f...**	**0.46**	**0.56**	**0.39**	**2% (108/4579)**
⚠	**offers**	**0.41**	**0.46**	**0.37**	**11% (522/4579)**
	receives	0.59	0.64	0.55	18% (809/4579)
	sells	0.64	0.72	0.57	4% (172/4579)

Fig. 12. The performance indicators for each relation type

As can be observed, eight of the relation types have the F1 score over 0.5 and the others a little below the threshold. All the relation types have a better precision than recall, except for "happens_in", which is also one of the most frequent annotations (16%), alongside "benefits_from" (16%) and „receives" (18%). On the other hand, the relation types „is_reponsible_for", "employs" and "sells" are rarely met in the training documents, therefore they may be not as relevant for the model as the others. The confusion matrix for RE (table 4) shows no confusions between any relation type and the others. This was expected, since any relation is defined by a parent and a child and there are no relations with the same parent-child combination. However, there is a relatively high percentage of confusions between correctly identified and false positive types of relations.

Table 4. The confusion matrix for RE

Relation Types	**benefits_from**	**books**	**communicates_with**	**employs**	**experiences**	**happens_in**	**has**	**impacts**	**is_responsible_for**	**offers**	**receives**	**sells**	**Total**
benefits_from	176	0	0	0	0	0	0	0	0	0	0	0	300
books	0	56	0	0	0	0	0	0	0	0	0	0	103
communicates_with	0	0	64	0	0	0	0	0	0	0	0	0	118
employs	0	0	0	14	0	0	0	0	0	0	0	0	30
experiences	0	0	0	0	57	0	0	0	0	0	0	0	105

happens_in	0	0	0	0	0	110	0	0	0	0	0	0	265
has	0	0	0	0	0	0	33	0	0	0	0	0	80
impacts	0	0	0	0	0	0	0	64	0	0	0	0	136
is_responsible_for	0	0	0	0	0	0	0	0	14	0	0	0	36
offers	0	0	0	0	0	0	0	0	0	67	0	0	182
receives	0	0	0	0	0	0	0	0	0	0	172	0	311
sells	0	0	0	0	0	0	0	0	0	0	0	39	68
Total	**252**	**73**	**101**	**19**	**101**	**283**	**49**	**124**	**25**	**147**	**269**	**54**	**2365**

4.3. Results compared to other projects

Below the performances of the model described in the paper are compared to those of other models that use similar techniques and technologies for NLP. Six studies have been identified that use similar technologies for both model development and performance evaluation. This is a fact of great importance, because not having the same methodological approach, the comparison of indicators presented in the studies would not have been valid. Table 5 shows a comparison between these studies and the model described in the present paper.

Table 5. Comparison of the performance of our model in relation to other studies

No.	Study	NLP based on ML tool	Domain	F1 Score for NER	F1 score for RE	Number of classes	Number of relations
1	Our model	Watson Knowledge Studio	Hotel	0.79	0.61	6	14
2	[26]	Watson Knowledge Studio	Cybersecurity	0.81	0.58	18	33
3	[27]	Watson Knowledge Studio	IoT security	0.68	0.46	17	30
4	[28]	Watson Knowledge Studio	Medical field	0.49	0.19	5	N/A
5	[29]	Watson Knowledge Studio	Shipping	0.67	0.55	4	2
6	[30]	Watson Knowledge Studio	Medical field	0.73	N/A	5	N/A
7	[31]	Stanford Named Entity Recognizer	Cybersecurity	0.80	N/A	10	N/A

N/A = Not available

Similar comparisons are presented in papers [26] and [27]. In addition to the F1 scores for NER and RE, table 5 also takes into account the number of classes for each

of the two functionalities, as well as the software tool that was used to implement the models. All identified projects use the Watson Knowledge Studio service, except for the study conducted by [31] in which Stanford Named Entity Recognition was preferred.
The fields in which the models were applied vary: three projects with applications in cybersecurity, two in medical field, one in shipping and one in the hotel field (the model described in this paper).
The projects [26], [31], as well as our model present the highest F1 scores for NER, having very close performance indicators values. The highest value for the precision was obtained by [26] - 0.88, followed by [31] and our model, with the same precision of 0.83. All three studies have the recall lower than the precision.
Regarding the RE task, the model presented in this paper achieves the best performances (0.61), followed by [26] with the F1 score equal to 0.58. Once again, both projects have higher value for precision in comparison with recall, by 0.07 in the case of our model and by 0.10 in the case of the project [26].
Table 5 also illustrates the number of classes and the number of relations for each study. The more classes the NLP model is based on, the better the level of understanding of the field. On the other hand, the complexity of the model increases in direct proportion to the number of classes. In order to obtain a more detailed representation of the hotel industry, while maintaining the complexity of the model at an acceptable level, we chose to include a relatively small number of classes (6). However, we decided to divide four of them into subclasses, totaling 14 such subtypes, as presented in section 2.
The developed model presents high performances in relation to the studies identified in the specialized literature, aspect that highlights the relevance of the research.

5. Conclusions and future work

In the context of the ever-expanding volume of the Internet, automated data processing solutions are becoming increasingly necessary. In this paper, we study the current issues related to the processing of natural language with the help of artificial intelligence.
The paper describes detail the development and implementation processes of a NLP based on ML model specialized in the hotel field. The methods, techniques and technologies used are described. The model performs three types of tasks: NER, RE and SA. The performance scores were presented, along with a description of the testing methodology, the values of the indicators and how they were obtained. High values of F1 scores for both entity recognition and relationship extraction demonstrate the validity of the model. Also, it shows good performances in comparison with other similar projects.
In order for the model to be usable by any interested individual, a web application was developed. This allows visitors to automatically analyze hotel reviews. The application also contains a feedback form, where users can offer valuable insights and help improve the solution.
In addition to the scientific achievements presented, a limitation of our work was also identified. This is related to the use of commercial software tools (Watson Knowledge Studio and Watson Natural Language Understanding) as part of the solution. In the future we want to develop similar models, but using only free and open-source libraries and frameworks.
For future research, we would like to test new techniques, methods and technologies specific to the NLP field. It is also desired to develop new ML models for other fields, apart from the hospitality industry, as well as developing solutions for NLP in the Romanian language.

References

[1] Giacomo Del Chiappa, Carlota Lorenzo-Romero , and Maria-del-Carmen Alarcon-del-Amo, "Profiling tourists based on their perceptions of the trustworthiness of different types of peer-to-peer applications," *Current Issues in Tourism, 21.3*, pp. 259-276, 2018.

[2] Markus Schuckert, Xianwei Liu, and Rob Law, "Hospitality and tourism online reviews: Recent trends and future directions," *Journal of Travel & Tourism Marketing, 32.5*, pp. 608-621, 2015.

[3] K. L. Xie, C. Chen, and S Wu, "Online Consumer Review Factors Affecting Offline Hotel Popularity: Evidence from Tripadvisor," *Journal of Travel & Tourism Marketing. Advance online publication. doi:10.1080/10548408.2015.1050538*, 2015.

[4] David D'Acunto, Annamaria Tuan, and Daniele Dalli, "Are Online Reviews Helpful for Consumers?: Big Data Evidence From Services Industry," *Exploring the Power of Electronic Word-of-Mouth in the Services Industry. IGI Global* , pp. 198-216, 2020.

[5] Fernando Iafrate, *From big data to smart data. Vol. 1.:* John Wiley & Sons, 2015.

[6] Aitor García-Pablos, Montse Cuadros, and Maria Teresa Linaza, "Automatic analysis of textual hotel reviews," *Information Technology & Tourism 16.1*, pp. 45-69, 2016.

[7] Yin Kang and Lina Zhou, "RubE: Rule-based methods for extracting product features from online consumer reviews," *Information & Management, 54.2*, pp. 166-176, 2017.

[8] Roshan Fernandes and Rio D'Souza GL., "Semantic analysis of reviews provided by mobile web services using rule based and supervised machine learning techniques," *International Journal of Applied Engineering Research 12.22*, pp. 12637-12644, 2017.

[9] P. S. Bhargav, G. N. Reddy, R. R. Chand, K. Pujitha, and A. Mathur, "Sentiment Analysis for Hotel Rating using Machine Learning Algorithms," *International Journal of Innovative Technology and Exploring Engineering (IJITEE) 8.6*, 2019.

[10] C. H. Ku, Y. C. Chang, Y. Wang, C. H. Chen, and S. H. Hsiao, "Artificial Intelligence and Visual Analytics: A Deep-Learning Approach to Analyze Hotel Reviews & Responses," in *In Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[11] B. Cowan et al., "Named entity recognition in travel-related search queries," in *In Twenty-Seventh IAAI Conference*, 2015.

[12] C. Chantrapornchai and A. Tunsakul, "Information Extraction based on Named Entity for Tourism Corpus," in *16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2019, pp. 187-192.

[13] K. Zvarevashe and O. O. Olugbara, "A framework for sentiment analysis with opinion mining of hotel reviews," in *Conference on Information Communications Technology and Society (ICTAS)*, 2018.

[14] C. I. Lee, T. C. Hsia, H. C. Hsu, and J. Y. Lin, "Ontology-based tourism recommendation system," in *4th International Conference on Industrial Engineering and Applications (ICIEA)*, 2017, pp. 376-379.

[15] M. Y. Pai, D. C. Wang, T. H. Hsu, G. Y. Lin, and C. C. Chen, "On Ontology-Based Tourist Knowledge Representation and Recommendation," *Applied Sciences, 9(23)*, p. 5097, 2019.

[16] L. Afuan and N. Hidayat, "Ontology model for tourism information in Banyumas," in *AIP Conference Proceedings 2094, (1)*, 2019.

[17] C. Lohvynenko and D. Nedbal, "Usage of Semantic Web in Austrian Regional Tourism Organizations," in *International Conference on Semantic Systems*, 2019, pp. 3-18.

[18] M. Chaves and C. Trojahn, "Towards a multilingual ontology for

ontology-driven content mining in social web sites," in *Proceedings of the ISWC 2010 Workshops, Volume I, 1st International Workshop on Cross-Cultural and Cross-Lingual Aspects of the Semantic Web*, 2010.

[19] D. H. Yoo and Y. M. Suh, "An Ontology-based Hotel Search System Using Semantic Web Technologies," *The Journal of Society for e-Business Studies, 13(4)*, pp. 71-92, 2008.

[20] A. R. Alaei, S. Becken, and B. Stantic, "Sentiment analysis in tourism: capitalizing on big data," *Journal of Travel Research, 58(2)*, pp. 175-191, 2019.

[21] S. Banerjee and A. Y. Chua, "Trust in online hotel reviews across review polarity and hotel category," *Computers in Human Behavior, (90)*, pp. 265-275, 2019.

[22] Tiberiu-Marian Georgescu, Modelarea Bazată pe Volume Mari de Date. Securitate Cibernetică în Contextul Big Data (Eng. "Modelling Based on Large Volumes of Data. Cybersecurity in the Context of Big Data"). Ph.D. Thesis, The Bucharest University of Economic Studies, Bucharest, Romania, 2019

[23] Tiberiu-Marian Georgescu et al., "A Survey on Named Entity Recognition Solutions Applied for Cybersecurity-Related Text Processing," in *International Congress on Information and Communication Technology*, London, 2020.

[24] G. Acampora, V. Loia, C.S. Lee, and M. H Wang, *On the Power of Fuzzy Markup Language.*: Springer, 2013.

[25] CoNLL. (2018) Universal Dependencies. [Online]. HYPERLINK "https://universaldependencies.org/conll18/evaluation.html" https://universaldependencies.org/conll18/evaluation.html

[26] Tiberiu-Marian Georgescu, "Natural Language Processing Model for Automatic Analysis of Cybersecurity-Related Documents," Symmetry 12.3, 2020.

[27] T. M. Georgescu, B. Iancu, and M. Zurini, "Named-Entity-Recognition-Based Automated System for Diagnosing Cybersecurity Situations in IoT Networks," *Sensors, 19(15)*, 2019.

[28] Timothy, et al. NeCamp. (2017) Data Science For Social Good, University of Chicago. [Online]. "https://dssg.uchicago.edu/wp-content/uploads/2017/09/necamp.pdf"

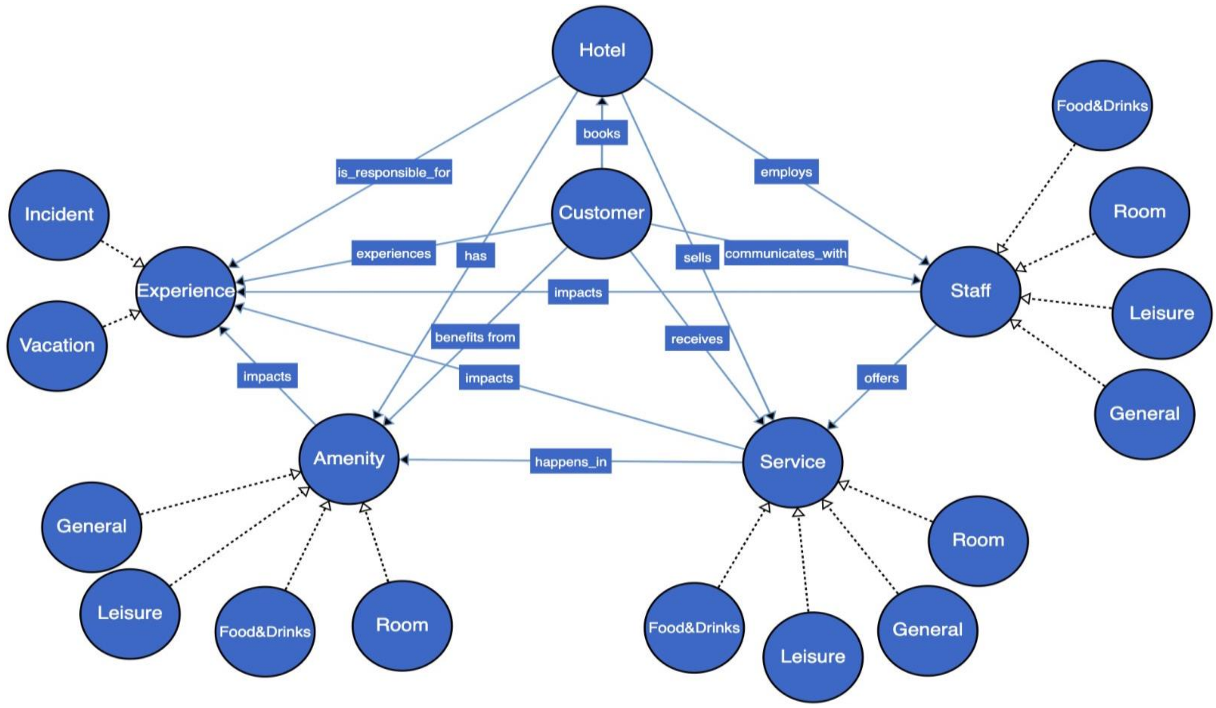
[29] J. E. H. Fritzner, "Automated Information Extraction in Natural Language (Master's thesis, NTNU)," Norwegian University of Science and Tehnology, Master's thesis 2017.

[30] L. Tonin. (2017) Digitala Vetenskapliga Arkivet. [Online]. "http://www.diva-portal.org/smash/get/diva2:1087619/FULLTEXT01.pdf"

[31] A. Joshi, R. Lal, T. Finin, and A. Joshi, "Extracting cybersecurity related linked data from text.," in IEEE Seventh International Conference on Semantic Computing (pp. 252-259), 2013, pp. 252-259.

Annexes

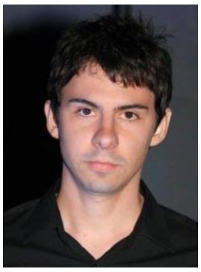
Annex 1. The extended version of the ontology, including the subclasses



Annex 2. The entities of the class Amenity

Entity	Part of speech	Lemmas			
a la carte restaurant	noun	a la carte restaurant	a la carte restaurants	a-la-carte restaurant	a-la-carte restaurants
air conditioning	noun	air conditioning	air-conditioning		
ambiance	noun	ambiance			
architecture	noun	architecture			
arrival area	noun	arrival area	arrival areas		
atmosphere	noun	atmosphere			
balcony	noun	balcony	balconies		
bali bed	noun	bali bed	bali beds	balinese bed	balinese beds
bar	noun	bar	bars		
bathroom	noun	bathroom	bath	bath room	
beach	noun	beach	beaches		
beach bed	noun	beach bed	beach bed	beachbeds	beach beds
beach chair	noun	beach chair	beach chairs		
beach grill	noun	beach grill			
beachfront suite	noun	beachfront suite	beachfront suites	beach front suite	beach front suites
bed	noun	bed	beds	bedding	
bedding	noun	bedding			
blanket	noun	blanket	blankets		
buffet restaurant	noun	buffet restaurant	buffet restaurants		

Bogdan-Ştefan POSEDARU graduated from the Faculty of Business and Tourism in 2012. He is currently a student at the Informatics Systems for the Management of Economic Resources Master program at the Bucharest University of Economic Studies. He worked as a freelancer in the IT field for 3 years. He was a member in various research projects, and since September 2018 he is one of the three co-founders of the start-up Chess Coders (https://chesscoders.com/). His main fields of interest are web technologies and natural language processing.



Tiberiu-Marian GEORGESCU graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2012. In 2015 he graduated the Informatics Systems for the Management of Economic Resources Master program. He completed his PhD program in Economic Informatics in September 2019 at the Bucharest University of Economic Studies. Currently, he is working as a Research Assistant in the Department of Economic Informatics and Cybernetics at the Bucharest University of Economic Studies. His main fields of interest are cybersecurity, machine learning and natural language processing.

Florin-Valeriu PANTELIMON graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2015. He is currently a student at the Informatics Systems for the Management of Economic Resources Master program at the Bucharest University of Economic Studies. He has been working as a software developer for several companies ranging from consulting companies, healthcare and clinical trials to international stock exchanges and game development companies. His main fields of interest are entrepreneurship, cloud computing and web development.