Evaluating Information Extraction Approaches in the Construction of a Real Estate Observatory

Luciana Tanevitch[1][0000−0002−5322−9314], Leandro Antonelli[1,3][0000−0003−1388−0337], and Diego Torres[1,2][0000−0001−7533−0133]

[1] LIFIA, CICPBA-Facultad de Informática, UNLP, Argentina
{name.surname}@lifia.info.unlp.edu.ar
[2] Departamento de Ciencia y Tecnología, UNQ, Argentina
[3] CAETI, Facultad de Tecnología Informática, UAI, Argentina

Abstract A real estate observatory plays a significant role in the aggregation and analysis of real estate market data. The information that lies in real estate advertisements can be leveraged to populate such an observatory. However, this data can present itself in both a structured and an unstructured manner. Unstructured data represents a problem to automatically process and extract information since it lacks a predefined structure. Thus, there's a need for techniques to give structure to unstructured data. Information Extraction (IE) is the process of structuring data from unstructured data. Natural Language Processing techniques enable machines to understand texts, making them particularly significant in the context of IE. This work evaluates both rule-based and machine-learning based IE approaches to extract features from real estate descriptions within advertisements. Those features are relevant in the context of real estate observatory construction. The performance of each approach is measured using precision, recall and f1-score metrics.

Keywords: Information Extraction · Natural Language Processing · Real Estate Observatory

1 Introduction

Unstructured data represents 80% of the information on the Web, in the form of text, photos, videos, etc. [11, 13]. This type of data lacks organization so analyzing it implies significant challenges, particularly due to machines' limited understanding of natural language. Structured data is more useful in process automation because of its standardized format, which defines the structure of the elements and how are they related. For example, efficient searching could be performed using contextual information, narrowing the universe of possible results [4].

Natural Language Processing (NLP) is a technique that allows machines to process natural language automatically, enabling human-machine understanding [8]. NLP tools can be used to extract information from unstructured texts.

Information Extraction (IE) is the process of analyzing text to extract concepts. Its goal is to discover structured information from unstructured or semi-structured text [10, 12]. IE can be divided in machine-learning based and non-machine-learning based.

The early IE systems were rule-based, where patterns based on linguistic features are defined [18]. Rule-based systems are robust if patterns are well-defined, but this requires knowing the data writing format to build effective patterns. Machine learning-based systems are growing exponentially, and they can achieve outstanding results. Named Entity Recognition (NER) is the process of identifying entities within a text. Those entities refer to relevant concepts existent in the real world, such as places, people, organizations, etc. [17]. Although several models are pre-trained to perform NER tasks, they are only able to recognize some limited entities. When the need for more entities arises, it is necessary to train the model to identify those new entities. NER models are usually trained over supervised learning, in which case it is required to have a big corpus of annotated data.

The leading machine learning-based approach to date is based on the transformers architecture [19]. Models based on transformers can perform NLP tasks to extract features from text. Transformers architecture have improved the performance of IE systems because of its attention mechanism. Within the IE context, Question Answering (QA) is a method that can be used to extract information by asking questions in which the answers are extracted from text. QA leverages transformers architecture such as BERT, having general-purpose pre-trained models to perform the task. Additionally, generative models can perform more advanced tasks, such as specifying the extraction format or adding instructions.

Several works describe the use of IE techniques to structure information. Ghani et. al. [9] use semi-supervised learning algorithms to extract explicit and implicit attribute-value pairs from product descriptions to augment databases of products. More [15] uses deep learning methods to extract attribute values from product titles to augment product metadata for e-commerce catalogs. Linková & Gurský [14] present algorithms to extract attributes and their values from product descriptions on e-shops, focusing on three data types: string, boolean and numeric. Those attributes can be useful in deduplication tasks, that is identifying identical resources. Blandón & Zapata [3] propose a rule-based framework that uses GATE-JAPE patterns to extract attribute values from text, to populate an ontology. Sabeh et. al. [16] develop a tool based on QA for attribute verification and enrichment on the e-commerce domain. Wang et. at. [21] propose an approach to extract attribute-value pairs using QA. Their model can classify unanswerable questions based on their context. State-of-art works describe methods and prompt designs to use Large-Language-Models [1, 5] to perform attribute-value extraction from descriptions, specifically, Brinkmann et. al. use ChatGPT to extract product attributes and brand from Amazon listings [6].

Baur et. al. [2] evaluate several machine learning models to valuate real estate based on their textual descriptions.

A Real Estate Observatory (REO) is a tool for analysts to discover trends on real estate market. Hence, the absence of data is a problem because no database can be built without data. Since real estate listings contain a lot of information useful to valuate real estate, Web scrapers can be used as a solution to extract semi-structured features such as bulleted or tabular information. But Web scrappers typically struggle with unstructured information, so IE techniques can be performed to extract features that are present in text, such as from real estate descriptions provided by advertisers.

The contribution of this work is an analysis of different IE approaches to extract real estate features from unstructured text, specifically real estate descriptions on online listings. Approaches are evaluated using precision, recall and f1-score metrics with exact matching. The goal is to extract features that can be explicitly mentioned in the text, or inferred by other extracted features. For example, given the following text: *Lot for sale in an excellent location, with a structure practically ready for demolition. Located at Soler 700, Bahía Blanca. Ideal for construction companies. It boasts 8.66 meters of frontage, 23 meters on the west side, and 19.60 meters on the east side. Zoned as C2 with a FAR of 2.4.*, features able to extract are address (Soler 700), location (Bahía Blanca), FAR (2.4), and lot dimensions (8.66 x 23 x 19.60). Irregularity of lot is not mentioned explicitly but given lot dimensions, it can be inferred that the lot is irregular.

This paper is organized as follows. Section 2 describes different IE approaches both rule-based and machine-learning-based that are evaluated in this work, and the process to extract features from real estate descriptions using those approaches. Section 3 describes the evaluation method, the employed metrics and the ground truth construction to evaluate each approach. The results of the evaluation are described in Section 4. Section 5 summarizes the application and limitations of the approaches for the extraction of features. Finally, Section 6 outlines the paper's conclusions and suggests some possible future works.

2 Information Extraction Approaches for Real Estate Text Mining

The selection of target variables is determined by a team of domain experts, who assess their impact on the REO. This study focuses on identifying the most significant variables, although others may also be considered. Table 1 enumerates, describes and exemplifies each variable.

Different IE approaches are evaluated in this work, in order to assess their performance in extracting variables described in Table 1. The following subsections explore different approaches and their application in extracting attribute-value pairs in descriptions of real estate listings.

Variable	Description	Type	Example
Address	It describes the address where the property is located. This field can be useful to enable automatic geocodification.	String	Calle 7 n° 4534 Av. 32 y Av. 7 Independencia esq. San Martín Independencia e/ San Martín y Caseros
FOT	FOT is an indicator of the land's potential for vertical construction. It uses to be a single value, but sometimes multiple values can be mentioned.	Number	FOT 3 FOT 2.5 FOT residencial: 2, FOT comercial: 3
Irregular Lot	It indicates if the lot is not rectangular. Irregular lots typically have a lower price than regular ones because they complicate the constructive utilization of real estate products developed on the lot.	Boolean	Lote irregular Lote con forma de martillo Lote triangular
Lot dimensions	It shows the lot sizes.	String	10 x 40 10m x 40m 10mts x 40 mts 10 metros de frente x 40 metros de fondo
Corner	It indicates if the lot is located in a corner.	Boolean	Lote en esquina
Neighborhood	It is the name of the neighborhood.	String	Barrio Las Camelias Club de Campo Abril Barrio Privado San Sebastián
Facades	The number of facades of the property. It indicates that the lot has access to more than 1 street.	Number	2 frentes doble frente salida a dos calles
Pool	It indicates if the lot has swimming pool.	Boolean	...con pileta..

Table 1. Variables to be extracted

2.1 Rule-based Matching

Rule-based matching is a well-known and effective approach to extract information. In particular, using NLP, patterns could be defined to extract information according its linguistic features. Spacy[4] is a library that includes several IE tools including rule-based matching support. Hence, patterns are lists of dictionaries where each dictionary describes the linguistic features that each token should have to make a match. POS tag and dependency tags could be used to indicate which features tokens should satisfy.

The address can be written in different formats, as shown in Table 1. After analyzing available data, three main format of addresses can be detected: `street name and number` (Calle 7 n° 4534), `street name and street name` (Av. 32 y Av. 7, Independencia esq. San Martín), `street name between streets` (Independencia e/ San Martín y Caseros). In addition, addresses in private neighborhoods use to be organized as 'lot number' or 'block number', and there are other formats that can not be generalized by a pattern. Keywords such as 'calle, avenida, diagonal' and their abbreviations can be present or not. That may be matched by the following sequence: {'LOWER':{'IN': ['calle', 'avenida', 'av', 'diagonal', 'diag']}, 'OP':'?' }. Since streets can be defined with number or name, both number and proper nouns should be considered: {'POS': {'IN': ['PROPN', 'NUM']}, 'OP': '+'}. Street number can be exact or not, for example, 'calle Independencia 4555' is an exact address, but 'calle Independencia al 4000' is not. This can be supported by making optional the preposition 'al': [{'LOWER': 'al', 'OP': '?'}, {'LIKE_NUM': True}]. Thus, by analyzing the structure of addresses and combining patterns, the different address structures can be covered.

Accordingly, a pattern to detect FOT value could be written as follows: [{'TEXT':'FOT'}, {'LIKE_NUM': True}]. Hence, model will match 'FOT 3' and 'FOT 2.5' but not 'f.o.t 3', 'fot 3', 'F.O.T 3'. Patterns should cover most common written formats, a more adequate pattern could be [{'LOWER':{'IN':['fot', 'f.o.t']}}, {'LIKE_NUM': True}], where 'fot', 'FOT', 'f.o.t', 'F.o.t', 'Fot', 'F.O.T' can match. Since FOT can take alternative values, optional tokens can be included to cover this cases: [{'LOWER':{'IN':['fot', 'f.o.t']}}, {'LOWER': {'IN': ['res', 'residencial', 'comercial', 'com', 'industrial']}, 'OP': '?'}, {'LIKE_NUM': True}].

To detect lot dimensions, a pattern that identifies the format `number x number` is defined (10 x 40). Unit of measure can be present or not (10 mts x 40 mts). Additionally, a pattern to find coincidences based on the structure `number de frente x number de fondo` is defined (10 mts de frente x 40 mts de fondo).

Facades number is usually specified as `number frentes`, but expressions referring to multiple street frontages are also common. So both cases are considered to define an adequate pattern.

Neighborhood name can be extracted by a pattern that searches the word 'Barrio' followed by a sequence of one or more proper nouns (Barrio Náutico).

[4] https://spacy.io/

As private neighborhoods are also considered, the pattern also covers keywords used to refer to them, such as 'Country', 'Estancia', 'Club'.

To detect boolean variables, an exact mention of the attribute is considered as a true value. In the case of the shape of the lot, irregularity can be detected by the presence of the word 'irregular' but also by an adjective modifying the word to refer to 'lot' or 'shape' (lote martillo, lote con forma triangular).

2.2 Named Entity Recognition

Named Entity Recognition is an NLP task for information extraction that involves automatically finding and classifying concepts according to defined categories [8]. Since trained NER models have standard categories such as location, person and organization, a new model should be trained to allow the model to use other categories. Training a new NER model requires annotated data, so the first step is to define tags to be used for the annotating process. One tag is defined for each variable on Table 1. Then, a training dataset is created using the field `description` of the REO knowledge base, which is the textual description on the real estate listing. Afterward, data should be labeled using defined tags. This requires an effort from a domain expert team to ensure that the process is homogeneous, i.e, all participants annotate data following the same criteria. For example, if dimensions are detected, include the unit measure if present. To make the process easier for people, NER Annotator Tool for Spacy is used to perform the task. This Web application allows users to load a text file and a tags file, and use a graphic interface to annotate each description. The generated file is used to train a Spacy model. In this work, `es_core_news_lg` (corresponding to large Spanish model) is used.

2.3 Transformers-based

Transformers is a novel architecture that uses self-attention mechanism to process data, making it suitable for tasks that require understanding of context and semantics [19]. Transformers are widely used in NLP tasks, having great results in this application.

Question-Answering Question-Answering (QA) is an approach that enables machines to extract answers to certain questions given a context. `Transformers` library available for Python, provides pre-trained models for QA task, which can be easily manipulated throw the `pipeline` API since it abstracts the implementation. BERT [20] is a state-of-the-art NLP model that is based on transformers architecture and can be used to perform the QA task. Three pre-trained Spanish models are selected to compare their performance. The difference lies in optimization techniques.

1. mrm8488/bert-base-spanish-wwm-cased-finetuned-spa-squad2-es
2. timpal0l/mdeberta-v3-base-squad2

3. rvargas93/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es

Since (1) is the base BERT Spanish model, (2) performs disentangled attention and enhanced mask decoder to improve the base model and (3) performs distillation process to compress the model to a smaller and more efficient version.

As it is mentioned above, `pipeline` API simplifies model manipulation, so it is used to evaluate QA performance of the different models. The input is a set of questions written to extract each attribute. For example, to detect the address a possible question could be "Which is the address of the property?" and the model should return the address from the property description if present and abstain if it isn't.

Generative Models Generative models, which are considered state-of-the-art, have the ability to generate outputs based on the specific task required in the input. These models are designed to understand the underlying patterns in the data and use this understanding to generate new data, which is useful in diverse applications such as information extraction. GPT-3 [7] is a huge generative pre-trained model based on transformers architecture and attention mechanism. This attention mechanism allows the model to focus on the most relevant parts of the input when making predictions, leading to more accurate and contextually aware outputs. GPT-3 can perform as a question-answering model, but also it can follows a task provided in the input. For example, instructions for data extraction and output formatting can be provided. Hence, prompt engineering has a lot to do with the result that the model retrieves. To extract real estate features, the provided instructions are related to the specific attributes that should be extracted. This includes the type of each variable whether it's a string, a number, or a boolean. The output format is also indicated in the instructions. It's important to note that the model is instructed not to add any information but to respond in a closed manner. Finally, it is specified to put null value if no attribute is detected in the text.

3 Evaluation Method

This section reports the evaluation of the performance of each model to automatically detect features in real estate descriptions. In particular, we want to know the capacity of each model to correctly detect variables detailed on 1. Based on extracted attributes, other attributes can be deduced. To evaluate this we use metrics described as follows.

3.1 Metrics

The most widely used metrics in machine learning are used to evaluate the performance of each approach: precision, recall and f1-score. Some concepts are tightly related to this metrics because they are used to calculate the results. True positive (TP) are pairs correctly identified by the model. Exact matching

is used to calculate TP which means that the predicted and expected pair should be exactly the same. False positive (FP) are those pairs raised by the model which are not expected by the ground truth. True negative (TN) occurs when the model doesn't recognize any pair but no pair is expected. Another case of TN is when a predicted and expected pair exists but they don't match (prediction is wrong). False negative (FN) occurs when the model doesn't recognize any pair, but a pair is expected to be recognized. These concepts play a crucial role in understanding the evaluation metrics. Precision measures the approach's ability to correctly identify the attribute-value pairs. Recall represents the approach's capability to detect all the existent pairs. F1-Score is a harmonic measure between precision and recall.

3.2 Ground Truth for Real Estate Data Analysis

A ground truth (GT) was developed to evaluate the IE approaches. Ground Truth is a dataset that contains real information. This is useful to evaluate results thrown by the different approaches and classify them as correct or incorrect. The GT is manually built and curated to ensure reliability. Real estate listings were manually selected from different portals. Only the description on each listing is considered to build the GT, considering those that includes the desired variables. Moreover, selected listings' descriptions should cover the variety of written formats of each variable. For example, in the case of lot dimensions, it represents to have listings where lot dimension is present as 20 x 10, 20 mts in the front and 10 mts in the back, front: 20 m back: 10 m, etc.

The normalization process removes information that is not relevant to the property (advertiser contact, information related to the real estate agency, legal disclaimers, auctioneers data), adding punctuation where necessary, and eliminating decorative characters. There are two considerations to be as faithful as possible to reality: not all variables are present in each listing, and variables have an uneven distribution, with varying quantities for each. Each row of GT contains the description, and extracted mentions for each variable of Table 1. Address, FOT and neighborhood name are set with exact mentions. Dimensions are set according to the format `number x number`, regardless of how it is mentioned in the description. Additionally, the analysis considered those variables that could be deduced from extracted ones to include them in the ground truth. For example, if lot dimensions are 20 x 10 x 30 x 40, it is assumed that the lot is irregular, even if it is not explicitly mentioned. The dataset contains 100 rows, corresponding to 100 real estate listing descriptions.

4 Results

This section reports the evaluation results of each IE approach in the task of detecting defined variables.

4.1 Rule-based matching

Address field was measured considering all possible formats at once. 25 address mentions were correctly found over 53 mentioned addresses. This approach obtained precision: 0.33, recall: 0.89 and f1-score: 0.49. This suggests that the model assigned addresses to a significant portion of the found pairs, but many of these assignments were incorrect, which showed that the model's ability is not optimal to accurately discern addresses from other text elements. **FOT** obtained 0.94 in precision, 0.92 in recall and 0.93 in f1-score. Since 40 FOT mentions were available in text, 35 were correctly detected. This results show a good ability to extract FOT variables. To detect **dimensions** variable, the model obtained precision: 0.84, recall: 0.56 and f1-score: 0.67. 39 pairs were correctly identified over a total of 74 mentions. Model got precision: 0.61, recall: 0.33 and f1-score: 0.43 on **neighborhood** detection. Only 8 of 28 pairs were correctly extracted. This means that the model performed badly to detect this variable. Detecting the **irregular** shape of the lot obtained precision: 1.0, recall: 0.64 and f1: 0.78. 20 of 31 pairs were detected. Results show that since the model is good at detecting this feature, not all pairs are being detected. 28 over 40 available **facades** mentions were detected, with precision: 0.93, recall: 0.71 and f1-score: 0.81. Since **swimming pool** and **corner** occurrences appear mentioned in text when they are present in the property, results were good, having f1-score of 1.0 in both cases. Results are summarized on Table 2

Variable	Precision	Recall	F1 Score
address	0.33	0.89	0.49
fot	0.94	0.92	0.93
irregular	1.0	0.64	0.78
dimensions	0.84	0.5655	0.67
corner	1.0	1.0	1.0
neighborhood	0.61	0.33	0.43
facades	0.93	0.71	0.81
pool	1.0	1.0	1.0

Table 2. Rule based matching performance by variable

4.2 Named Entity Recognition

Address field was measured considering all possible formats at once. This approach identified 29 over 53 mentions, with precision: 0.74, recall: 0.61 and f1-score: 0.67. 33 of 40 **FOT** pairs were extracted, giving precision: 1.0, recall: 0.82 and f1-score: 0.9. 41 pairs were correctly detected from a total of 74 in **dimensions** variable. Results were precision: 0.82, recall: 0.6 and f1-score: 0.69. Finding **neighborhood** name obtained 0.53 in precision, 0.32 in recall and 0.4 in f1-score. 8 over 40 pairs were recognized. 18 over 31 pairs were correctly identified for lot **irregularity**, with precision: 1.0, recall: 0.58 and f1-score: 0.73. 26

pairs which referring **facade** amount were detected over 40, with precision: 0.92, recall: 0.66 and f1-score 0.77. To detect **swimming pool** feature, precision was 1.0, recall 0.73 and f1-score 0.85, counting 17 over 23 identified pairs. **Corner** location obtained 1.0 in precision, 0.95 in recall and 0.97 in f1-score, with 19 of 20 pairs extracted.

Table 3 summarizes obtained results.

Variable	Precision	Recall	F1 Score
address	0.74	0.61	0.67
fot	1.0	0.82	0.9
irregular	1.0	0.58	0.73
dimensions	0.82	0.6	0.69
corner	1.0	0.95	0.97
neighborhood	0.53	0.32	0.4
facades	0.92	0.66	0.77
pool	1.0	0.73	0.85

Table 3. NER performance by variable

4.3 Transformers-based

Table 4 shows results obtained by each QA model, for each variable. Color pink highlights the model with the best results for each variable. mDeBERTa model outperforms on most variables compared to the other QA models. In detecting **addresses**, **dimensions** and **neighborhoods** all models had bad results. Analyzing the cause of failure, it is associated to a wrong result in most of them. Even modifying the syntax on the question used to extract those variables, performance didn't improve.

model	Address	FOT	Irregular	Dimensions	Corner	Neighborhood	Facades	Pool
BETO	p: 0.13 r: 0.73 f1: 0.22	p: 0.32 r: 0.65 f1: 0.43	p: 1.0 r: 0.35 f1: 0.52	p: 0.34 r: 0.75 f1: 0.47	p: 1.0 r: 0.15 f1: 0.26	p: 0.41 r: 0.17 f1:0.25	p: 0.6 r: 0.35 f1: 0.45	p: 1.0 r: 0.82 f1: 0.9
mDeBERTa	p: 0.29 r: 0.85 f1: 0.43	p: 0.77 r: 0.8 f1: 0.78	p: 0.9 r: 0.58 f1: 0.7	p: 0.42 r: 0.9 f1: 0.57	p: 1.0 r: 0.55 f1: 0.7	p: 0.52 r: 0.46 f1: 0.48	p: 0.82 r: 0.61 f1: 0.7	p: 1.0 r: 0.82 f1: 0.9
BETO + distilled	p: 0.21 r: 0.85 f1: 0.33	p: 0.2 r: 0.6 f1: 0.3	p: 0.89 r: 0.54 f1: 0.67	p: 0.34 r: 0.76 f1: 0.48	p: 1.0 r: 0.4 f1: 0.57	p: 0.41 r: 0.29 f1: 0.34	p: 0.68 r: 0.51 f1: 0.58	p: 1.0 r: 0.52 f1: 0.68

Table 4. Transformers: QA performance by variable

GPT-3 had great results in almost all variables. Most of the observed missings are due to the output format. Since the ground truth dataset is constructed from

literal extractions in the descriptions and evaluated with exact matching, pairs containing correct information but not in the format specified by the ground truth may be classified as false. For example, GPT may output 'Melo 836, Lomas del Mirador' while the ground truth states 'Melo 836'. This contains the correct address but includes Location data, resulting in it being computed as a false positive. All results are shown in Table 5.

Variable	Precision	Recall	F1 Score
address	0.45	0.84	0.59
fot	0.92	0.97	0.94
irregular	0.9	0.96	0.93
dimensions	0.86	0.98	0.92
corner	0.76	1.0	0.86
neighborhood	0.66	0.92	0.77
facades	1.0	1.0	1.0
pool	1.0	1.0	1.0

Table 5. GPT-3 performance by variable

General results Table 6 summarizes all results obtained, highlighting the model with the best performance for each variable. As observed, the best scores were obtained with GPT-3 in most of variables, while rule-based and NER achieved acceptable results.

Approach	Address	FOT	Irregular	Dimensions	Corner	Neighborhood	Facades	Pool
Rule based matching	P: 0.33 R: 0.89 F1: 0.49	P: 0.94 R: 0.92 F1: 0.93	P: 1.0 R: 0.64 F1: 0.78	P: 0.84 R: 0.56 F1: 0.67	P: 1.0 R: 1.0 F1: 1.0	P: 0.61 R: 0.33 F1: 0.43	P: 0.93 R: 0.71 F1: 0.81	P: 1.0 R: 1.0 F1: 1.0
NER	P: 0.74 R: 0.61 F1: 0.67	P: 1.0 R: 0.82 F1: 0.9	P: 1.0 R: 0.58 F1: 0.73	P: 0.82 R: 0.6 F1: 0.69	P: 1.0 R: 0.95 F1: 0.97	P:0.53 R: 0.32 F1: 0.4	P: 0.92 R: 0.66 F1: 0.67	P: 1.0 R: 0.73 F1: 0.85
Transformers (BETO)	P: 0.13 R: 0.73 F1: 0.22	P: 0.32 R: 0.65 F1: 0.43	P: 1.0 R: 0.35 F1: 0.52	P: 0.34 R: 0.75 F1: 0.47	P: 1.0 R: 0.15 F1: 0.26	P: 0.41 R: 0.17 F1: 0.25	P: 0.6 R: 0.35 F1: 0.45	P: 1.0 R: 0.82 F1: 0.9
Transformers (mDeBERTa)	P: 0.29 R: 0.85 F1: 0.43	P: 0.77 R: 0.8 F1: 0.78	P: 0.9 R: 0.58 F1: 0.7	P: 0.42 R: 0.9 F1: 0.57	P: 1.0 R: 0.55 F1: 0.7	P: 0.52 R: 0.46 F1: 0.48	P: 0.82 R: 0.61 F1: 0.7	P: 1.0 R: 0.82 F1: 0.9
Transformers (BETO + distilled)	P: 0.21 R: 0.85 F1: 0.33	P: 0.2 R: 0.6 F1: 0.3	P: 0.89 R: 0.54 F1: 0.67	P: 0.34 R: 0.76 F1: 0.48	P: 1.0 R: 0.4 F1: 0.57	P: 0.41 R: 0.29 F1: 0.34	P: 0.68 R: 0.51 F1: 0.58	P: 1.0 R: 0.5 F1: 0.68
Transformers (GPT-3)	P: 0.45 R: 0.84 F1: 0.59	P: 0.92 R: 0.97 F1: 0.94	P: 0.9 R: 0.96 F1: 0.93	P: 0.86 R: 0.98 F1: 0.92	P: 0.76 R: 1.0 F1: 0.86	P: 0.66 R: 0.92 F1: 0.77	P: 1.0 R: 1.0 F1: 1.0	P: 1.0 R: 1.0 F1: 1.0

Table 6. General results by variable

5 Discussion

Some limitations are being detected in the specific variable extraction along the different evaluation steps. The nature of each variable and the analyzed approach deserve a discussion on use and configuration. In the following, a detailed discussion of the approaches applied to the most relevant variables is presented.

Address detection can enable georeferencing. It is desirable to determine the format in which this feature is presented. Due to the high variability in writing formats, it is difficult to create patterns that cover all situations. On the other hand, training a NER to detect each type of address requires a lot of annotated data. GPT-3 tends to include the location in the address, resulting in false positives according to the exact matching proposed. A balanced alternative would be to combine GPT-3 with rule-based matching to extract an answer and then classify it in different formats.

Extracting the dimensions of the lot requires high precision because, as this variable is used to deduce lot irregularity, it must be ensured that the detected value is accurate.

Detecting a neighborhood name is a complex task for machines since it is also difficult for humans. As humans, we expect some keywords to easily detect a name as a neighborhood (Barrio, Country, Club de Campo, Estancia). But if none of those keywords are present, we can't know if the name is a neighborhood, a city, a locality or another place.

A central issue found with QA is interpreting answers when expecting boolean values. One possible approach could be to consider the response generated by the QA system as positive, regardless of its content. This may work when working with datasets that only mention a feature when it is present, which are most of the cases. Given that instances where the feature is mentioned negatively are exceedingly rare but not impossible, identifying certain keywords that indicate the presence of the variable in the obtained responses can enable the interpretation of the truth value (positive or negative) of that variable. In cases where the feature is not mentioned, the model should refrain from responding.

NER encounters similar challenges since it processes input as a sequence labeling problem. Therefore, this approach should also be applied in the presence of negative occurrences. For GPT-3, an adequate prompt offers enough guidance to entrust it with the task of determining the boolean value (or null).

6 Conclusions

This work evaluated different IE approaches to extract features from text, in the context of the construction of a real estate observatory. Both rule-based and machine-learning-based approach was evaluated over real estate descriptions, extracted and manually curated from online advertisements.

GPT-3 obtained the best results in this evaluation. GPT-3 can compute results automatically, which for other approaches would require defining additional rules to derive values based on the extracted information. For instance, if the text

mentions that the dimensions of a lot are 10x30x25x40, GPT-3 could identify that the lot is irregular. While it would be possible to define rules for NER-based approaches and rule-based approaches to set the number of sides to 2 if the extraction detects that the lot is in a corner and the facades amount is not specified (but not vice versa), it may also be possible to establish a rule to classify a lot as irregular based on the extracted dimensions (if more than 2 dimensions are specified and it's not rectangular, it's irregular). Another interesting approach is NER, which performs good results even with minimal training. These results could be further improved by training the model on a larger dataset.

Considering that exact match is the most restrictive evaluation, using partial matching could improve general performance in some variables. For certain variables such as address, dimensions, and neighborhood name, exact matching can be overly restrictive since if the model provides the correct data in a different format, it will be classified as false. Instead, partial matching can be useful to avoid such situations, thereby setting a threshold that is beneficial for obtaining the correct information without losses.

Future works relate to evaluating the performance of the approaches on a massive dataset, considering data augmentation techniques. Moreover, the extracted features should be aligned to existing ones in the real estate observatory, requiring techniques to detect debilities in available data.

References

1. Baumann, N., Brinkmann, A., Bizer, C.: Using llms for the extraction and normalization of product attribute values. arXiv preprint arXiv:2403.02130 (2024)
2. Baur, K., Rosenfelder, M., Lutz, B.: Automated real estate valuation with machine learning models using property descriptions. Expert Systems with Applications **213**, 119147 (Mar 2023). https://doi.org/10.1016/j.eswa.2022.119147, https://www.sciencedirect.com/science/article/pii/S0957417422021650
3. Blandón Andrade, J.C., Zapata Jaramillo, C.M.: Gate-Based Rules for Extracting Attribute Values. Computación y Sistemas **25**(4) (Feb 2021). https://doi.org/10.13053/cys-25-4-3493, https://cys.cic.ipn.mx/ojs/index.php/CyS/article/view/3493, number: 4
4. Blumberg, R., Atre, S.: The Problem with Unstructured Data
5. Brinkmann, A., Shraga, R., Bizer, C.: Product attribute value extraction using large language models. ArXiv **abs/2310.12537** (2023), https://api.semanticscholar.org/CorpusID:264305709
6. Brinkmann, A., Shraga, R., Der, R.C., Bizer, C.: Product information extraction using chatgpt (2023), https://api.semanticscholar.org/CorpusID:259262489
7. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language Models are Few-Shot Learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

8. Chowdhury, G.G.: Natural Language Processing
9. Ghani, R., Probst, K., Liu, Y., Krema, M., Fano, A.: Text mining for product attribute extraction. SIGKDD Explor. Newsl. **8**(1), 41–48 (jun 2006). https://doi.org/10.1145/1147234.1147241, https://doi.org/10.1145/1147234.1147241
10. Grishman, R.: We often refer to text as unstructured data. However, in reality, the text has a lot of structure—it's just that most of it isn't explicit, mak-. IEEE INTELLIGENT SYSTEMS (2015)
11. Inmon, W.H., Linstedt, D., Levins, M.: Chapter 4.4 - Unstructured Data. In: Inmon, W.H., Linstedt, D., Levins, M. (eds.) Data Architecture (Second Edition), pp. 89–97. Academic Press, second edition edn. (2019). https://doi.org/https://doi.org/10.1016/B978-0-12-816916-2.00013-9, https://www.sciencedirect.com/science/article/pii/B9780128169162000139
12. Jiang, J.: Information Extraction from Text. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, pp. 11–41. Springer US, Boston, MA (2012). https://doi.org/10.1007/978-1-4614-3223-4$_2$, https://link.springer.com/10.1007/978-1-4614-3223-4_2
13. Kulkarni, A., Shivananda, A.: Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python. Apress, Berkeley, CA (2019). https://doi.org/10.1007/978-1-4842-4267-4, http://link.springer.com/10.1007/978-1-4842-4267-4
14. Linková, M., Gurskỳ, P.: Attributes extraction from product descriptions on e-shops. In: ITAT. pp. 23–26 (2017)
15. More, A.: Attribute extraction from product titles in ecommerce. CoRR **abs/1608.04670** (2016), http://arxiv.org/abs/1608.04670
16. Sabeh, K., Kacimi, M., Gamper, J.: CAVE: Correcting Attribute Values in E-commerce Profiles. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4965–4969. ACM, Atlanta GA USA (Oct 2022). https://doi.org/10.1145/3511808.3557161, https://dl.acm.org/doi/10.1145/3511808.3557161
17. Sharnagat, R.: Named Entity Recognition: A Literature Survey
18. Small, S., Medsker, L.: Review of information extraction technologies and applications. Neural Computing and Applications (12 2013). https://doi.org/10.1007/s00521-013-1516-6
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, , Polosukhin, I.: Attention is All you Need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, , Polosukhin, I.: Attention is All you Need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
21. Wang, Q., Yang, L., Kanagal, B., Sanghai, S., Sivakumar, D., Shu, B., Yu, Z., Elsas, J.: Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 47–55. ACM, Virtual Event CA USA (Aug 2020). https://doi.org/10.1145/3394486.3403047, https://dl.acm.org/doi/10.1145/3394486.3403047