Attribute-Value Extraction: the case of a Real Estate Observatory

Luciana Tanevitch[1 [0000−0002−5322−9314]], Alejandro Fernández[1 [0000−0002−7968−6871]], Juan Pablo Del Río[2 [0000−0002−4031−3007]], and Diego Torres[1,3 [0000−0001−7533−0133]]

[1] LIFIA-CICPBA. Facultad de Informática, UNLP. La Plata, Argentina
name.surname@lifia.info.unlp.edu.ar
[2] LINTA-CICPBA CONICET. FaHCE, UNLP. La Plata, Argentina
[3] Departamento de Ciencia y Tecnología, UNQ. Bernal, Argentina

Abstract. Structured information is a valuable resource in information systems construction. The process of structuring unstructured data can be automated, but since machines can't directly process natural language texts, NLP techniques are required. This work aims to evaluate different approaches to perform attribute-value extraction in real estate descriptions, in the context of the construction of a real estate observatory for the Province of Buenos Aires. The performance of each model is measured using precision, recall and F1-score with a partial matching approach.

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

1 Introduction

E-commerce is an industry that has been growing rapidly over the last years. Every day people around the world publish products, search for products and order products online. Structured data is well-organized and can be easily analyzed [19], being fundamental to build recommendation engines, categorization systems, and product comparison platforms. Most of the information available on the Web is published in natural language, which makes it challenging for machines to automatically process that data and make inferences from it. Ontologies enables data to be organized and structured in a machine-readable format [25].

Real estate websites are not exempt from this characteristic, since they usually don't include metadata to automatically extract the information. Advertisements are usually written by users using templates, where they fill out a table with some predefined features, and free text descriptions. This work unfolds within the scope of the project named "Observatorio de Valores del Suelo e Instrumentos de Financiamiento del Desarrollo Urbano". The OVS is a collaborative initiative between the scientific-technical community and the provincial public sector, with the goal of systematically collecting georeferenced information on real estate values. Furthermore, it aims to foster the development of tools

for urban land management and public participation in real estate valuation [2]. OVS is structured by a real estate ontology and data is extracted from real estate advertisements [13].

Real estate advertisements include both tabular and unstructured information. Tabular data can be automatically extracted by agents, but descriptions entail challenges due to the machines' lack of comprehension of the natural language. Since valuable information can be present in real estate descriptions, it is desired to process them. Natural Language Processing (NLP) allows machines to understand texts written in human language [17]. Early attribute extraction systems were rules-based. Then, machine learning advances allowed researchers to discover new strategies to automatically extract data [12]. Atribute-value extraction is the task to identify a feature that describes a real object, and its correspondent value.

The goal of this work is to find an approach to extract attribute-value pairs from descriptions in real estate listings written in natural language in Spanish, in order to enrich and validate OVS data. For example, given an attribute-value pair stored in the OVS such as {`address: Buenos Aires al 4500`}, and a description in natural language about that real estate, the objective is to extract the value of the address from the description. This extraction aims to verify existing information or to add new information if it weren't present. For example, if the extracted description was 'Buenos Aires al 4500', the previous information can be validated, while if it was 'Buenos Aires n° 4565' more detail about the real address can be added.

This work evaluates three NLP-based approaches for attribute-value pair extraction in the real estate domain: Rule-based matching, Named Entity Recognition and models based on the Transformers architecture (in particular, Question-answering models and GPT-3). The evaluation is done using the precision, recall and F1-score metrics.

This article is organized as follows. Section 2 provides a review of related works in the context of Natural Language Processing (NLP) for attribute extraction within the e-commerce domain. Section 3 describes the OVS and defines the problem of detecting variables in real estate descriptions. Section 4 describes how different approaches are applied to perform the task of attribute-value extraction. Section 5 describes the metrics and the data used to evaluate the extraction approaches. Section 6 reports the results obtained for each approach to extract the desired features. Finally, section 7 summarizes the work and presents some future lines.

2 Related works

Baur et. al. [6] evaluate several machine learning models to valuate real estate based on their textual descriptions. Different techniques based on NLP are used to extract attribute-value pairs in natural language texts. Anantharangachar et. al. [5] extract features from text in the form of semantic triples to populate an ontology. They use NLP and pattern matching to extract features from a de-

scription, so they can build triples based on the subject-predicate-object tags but also discover values of the features present in text. Linková & Gurský [18] suggest methods to extract attributes and their values from product descriptions on e-shops. They propose algorithms to deal with three data types: string, boolean and number. Boolean attributes are detected as true if a mention is found in the text. Numeric attributes are detected using patterns to extract the attribute name, its value and the unit of measurement. To extract string data they apply exact matching and then search for the value based on known values for that attribute, but they mention that NLP could improve the performance. Al Amoudi et. al. [3] use rule-based matching to extract metadata of a book from the PDF file. Ghani et. al. [14] present a system based on deep learning algorithms with the capability of inferring implicit and explicit attribute-value pairs from product descriptions, to augment products' databases.

Zheng et. al. [27] present OpenTag, a tool to discover and replace missing values of certain attributes on product listings from unstructured text (such as title, description and bullets). They apply a deep learning architecture with an attention mechanism to detect the values. Sabeh et. al. [22] develop CAVE, a tool for attribute correction and enrichment on the e-commerce domain. CAVE is based on the Question-Answering paradigm, wherein each attribute is extracted through a question. Wang et. at. [26] propose AVEQA, a BERT-based model which can classify unanswerable questions based on its context. Probst et. al. [20] train a model to extract attribute-value pairs in product descriptions, based on the co-EM algorithm with Naïve Bayes to classify the identified concepts into attribute or value. They use a dependency parser to relate the attribute to its corresponding value. Finally, human intervention allows for correcting results returned by the model. IDEALO [1] is a product price comparison software. They propose a BERT-based solution as an improvement of the rule based method to extract numeric attributes present in product descriptions, to enrich the existent database.

Zou et. al. [28] introduce EIVEN, a generative framework to perform implicit attribute-value extraction from product descriptions. Brinkmann et. al. [7] uses ChatGPT to extract attributes and values from product descriptions. They compare its performance against different input designs, having the possibility that the model responds "I don't know" in case the answer is not present in the context. These input designs may be similar to QA for answering single questions or multiple questions, or may be stating the extraction task in a given response output format. The work also compares this approach with QA and NER.

3 Observatorio de Valores del Suelo

A Real Estate Observatory (REO) is a system that collects real estate data into a georeferenced database. An REO provides up-to-date and accurate information on trends, prices, supply, and demand for real estate in a specific location. This information is useful for professionals to make decisions based on the current market situation. In this regard two organisms of the Province of Buenos Aires,

OPISU and CIC introduce the project named "Observatorio de valores del suelo e instrumentos de financiamiento del desarrollo urbano" where the LIFIA research center participates in its construction. The main goal of the project is to have open data to contribute to urban financing through urban capital gains recovery instruments [10]. The Observatorio de Valores del Suelo (OVS) is a tool that will allow the State to quantify the appreciation of real estate prices, and therefore contribute to the implementation of policies to improve popular habitats [2].

The OVS database was built using real estate advertisements extracted from different e-commerce platforms in Spanish, and it is stored in RDF format. Semantic support is given to the OVS by a real estate ontology that defines the main concepts of this domain, such as the number of rooms, address, etc. [13]. Figure 1 depicts an instance of a property from the OVS. Some elements, such as room dimensions, bathroom dimensions, and coordinates, were extracted from tabular information. The description is stored as text according to the published listing, and it contains features that could be extracted as: {address: Avenida Montevideo y 105}, {lot dimensions: 8.66 x 28 mts}, {FOT: 1.8}. Structured

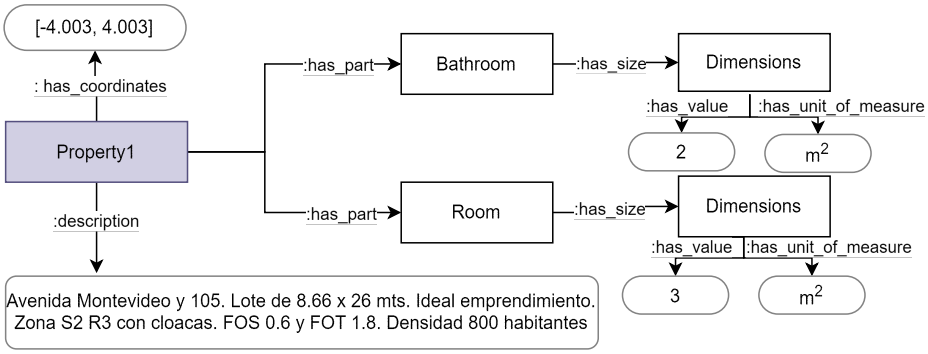

Fig. 1. Instance of a real estate in the OVS

data allows process automation, therefore recognizing attributes and their values in unstructured text is an essential task to improve the information on the OVS.

3.1 Problem definition

The problem addressed in this work is to extract attributes and their values present in the text descriptions of real estate listings stored in the OVS database. The variables to be extracted are defined by domain experts of LINTA, based on the relevance of those features to the OVS. They define a *variable* as an attribute that describes a real estate. For each variable, they define its name, a brief description, the data type (numeric, boolean, text) and the most common ways of writing them. Variables are explained below.

- **Address.** The address of the property. It is desirable to identify inconsistencies between the address field of the OVS and the address to be extracted from the description field of the OVS. If the address field is empty, it could be filled by the address extracted from the description.
- **FOT.** It's a numeric indicator of the land's potential for vertical construction.
- **Irregular lot.** The shape of the lot. It is a boolean variable that indicates whether the lot does not have a quadrilateral shape, which may imply a lower price of the lot.
- **Lot dimension.** Front and side sizes.
- **Corner.** It is a boolean variable that indicates if the property is located at a street intersection or not. These properties tend to be more expensive.
- **Neighborhood.** This variable indicates the name of the neighborhood.
- **Multiple facades.** It indicates the number of facades of the property.
- **Swimming pool.** It is a boolean variable that indicates if the property has a swimming pool or not.

4 Extraction approaches

4.1 Rule-based matching

NLP allows machines to process texts written by humans. Thus, each linguistic unit is named token, and each token is annotated by syntactic and grammatical tags. Spacy[4] is an NLP tool that provides rule-based matching by sequence matching and dependency matching. Hence, a pattern is defined as a list of dictionaries, within each dictionary describes the attributes that a token should satisfy to match. As a result, rule-based matching can be performed by defining patterns to automatically detect the structure defined by the list of dictionaries. Sequence matching is used to detect sequences of tokens, while dependency matching can be applied to extract matches based on the syntactic dependencies on the text.

For example, a pattern to identify a basic format address like "Independencia 1916" can be defined as follows: [{'POS': 'PROPN', 'OP': '+'}, {'LOWER': 'al', 'OP': '?'}, {'LIKE_NUM': True}]. The PROPN POS tag means proper noun. Names are proper nouns, so street names could be recognized with this tag. The '?' operator means *optional*, so the preposition *al* could be present or not. Finally, a numeric value determines the house number.

To define precise patterns, it is necessary to know how each variable is commonly written in texts. The address can be present in the descriptions in different formats: (1) street and number (Independencia 1239), (2) street and intersection (Independencia y 2), (3) street and between streets (Independencia e/ Industria y Edgar Aschieri), (4) by neighborhood name and/or lot in which the offer is located inside a condominium (Barrio Grand Bell Lote 57), (5) other types of situations that are not reflected in the previous categories (Ruta 15 km 12).

[4] `https://spacy.io/`

Multiple patterns are defined to support the extraction of addresses. FOT is associated to one number, but in a few cases it may have multiple values. A pattern is defined for each case. Dimensions are usually present in the following format: a number, followed by a keyword such as 'x', and another number (10 x 15 mts.). Unit measures can be present or not. It is also usually to find dimensions in a complex format, describing with many words the feature (15 metros de fondo por 25 metros de largo). To extract the name of a neighborhood, the keyword 'Barrio' (neighborhood) is expected to be present, followed by a sequence of tokens with the PROPN POS tag. The amount of facades is usually present as a number followed by the keyword 'frentes', but it is also common to find mentions as 'salida a dos calles', which means that the lot has exits onto multiple streets. For boolean variables, the presence of the attribute in the text is recognized as a positive value. For example, if the word 'esquina' (corner) is present, it means that the property is located at the intersection between two streets.

Table 1 summarizes the patterns defined for each variable and examples that the patterns can recognize.

4.2 Named Entity Recognition

Named Entity Recognition (NER) [23] is the task of NLP which allows to extract concepts from text and categorize them according to previously defined labels. Since extracting specific domain variables requires training a NER model, Spacy is chosen for this purpose because of its large model for Spanish language and its customizable NER component. NER is usually trained by supervised learning, so it requires a big corpus of annotated data. Since available open data in the real estate domain is very limited, corpus data should be generated. The task of creating the corpus to train a NER model requires defining the tags, having a domain dataset to annotate, and defining the annotation strategy. The annotation strategy involves a group of domain experts who must follow certain guidelines to create homogeneous annotations, i.e. having consistent annotations over the data to ensure the model is trained correctly.

Tags are defined according to the variables to be detected. For each possible writing format of addresses, one tag is defined. Then, for each of the remaining variables, a tag is defined. Data to be annotated is extracted from the OVS database, selecting only the description attribute and generating several text documents. Documents are equally assigned to each person involved in the task, in addition to a file that defines the tags to be used. NER Annotator for Spacy[5] is the software used to perform the task because of its easy-to-use web interface. People must annotate each description of their assigned documents with the available tags when a mention is present. When the process is finished, all annotations are merged into a single file which is then used to train the NER model.

[5] `https://tecoholic.github.io/ner-annotator/`

Table 1. Defined patterns for each variable

Variable	Pattern	Matching examples
Address (street and number)	[{'LOWER': {'IN': ['calle', 'avenida', 'av', 'diagonal', 'diag']}, 'OP': '?'}, {'TEXT': '.', 'OP': '?'}, {'POS': 'PROPN', 'OP': '+'}, {'LOWER': 'al', 'OP': '?'}, {'LIKE_NUM': True}] [{'LOWER': {'IN': ['calle', 'avenida', 'av', 'diagonal', 'diag']}, 'OP': '?'}, {'TEXT': '.', 'OP': '?'}, {'POS': 'PROPN', 'OP': '+'}, {'LOWER': 'n'}, {'TEXT': '°'}, {'LIKE_NUM': True}]	Av. Manuel Belgrano al 6200 Diag. 73 nro° 3450 Alberti 3359
Address (street and intersection)	[{'LOWER': {'IN': ['calle', 'avenida', 'av', 'diagonal', 'diag']}, 'OP': '?'}, {'TEXT': '.', 'OP': '?'}, {'POS': {'IN': ['PROPN', 'NUM']}, 'OP': '+'}, {'LOWER': {'IN': ['y', 'e', 'esquina', 'esq.']}}, {'LOWER': {'IN': ['calle', 'avenida', 'av', 'diagonal', 'diag']}, 'OP': '?'}, {'TEXT': '.', 'OP': '?'}, {'POS': {'IN': ['PROPN', 'NUM']}, 'OP': '+'},]	calle Alsina y Av. San Martín calle 19 y calle 36 7 y 50
Address (street and between streets)	[{'LOWER': {'IN': ['calle', 'avenida', 'av', 'diagonal', 'diag']}, 'OP': '?'}, {'TEXT': '.', 'OP': '?'}, {'POS': {'IN': ['PROPN', 'NUM']}, 'OP': '+'}, {'LOWER': {'IN': ['e/', 'entre', 'e']}}, {'LOWER': {'IN': ['calle', 'avenida', 'av', 'diagonal', 'diag']}, 'OP': '?'}, {'TEXT': '.', 'OP': '?'}, {'POS': {'IN': ['PROPN', 'NUM']}, 'OP': '+'}, {'LOWER': 'y'}, {'LOWER': {'IN': ['calle', 'avenida', 'av', 'diagonal', 'diag']}, 'OP': '?'}, {'TEXT': '.', 'OP': '?'}, {'POS': {'IN': ['PROPN', 'NUM']}, 'OP': '+'},]	calle 7 e/ 71 y 72 Independencia e/ San Martín y Belgrano
Address (lot in a private neighborhood)	[{'LOWER': 'lote'}, {'IN': ['NUM', 'PROPN']}]	lote 24
FOT	[{'RIGHT_ID': 'fot', 'RIGHT_ATTRS': {'TEXT': {'IN':['fot', 'f.o.t']}}}, {'LEFT_ID': 'fot', 'REL_OP': '>', 'RIGHT_ID': 'NUM', 'RIGHT_ATTRS': {'DEP': 'nummod'}}] [{'LOWER': {'IN':['fot', 'f.o.t']}}, {'LOWER': {'IN':['res', 'residencial', 'com', 'comercial', 'industrial']}, 'OP': '?'}, {'IS_PUNCT': True, 'OP': '?'}, {'LIKE_NUM': True}]	FOT 1 F.O.T 3,6 FOT residencial: 2.5 FOT comercial: 3
Irregular	[{'LOWER': 'irregular'}] [{'LOWER': {'IN':['lote', 'forma']}}, {'POS': 'ADJ'}]	Terreno irregular de ...
Lot dimensions	[{'LIKE_NUM': True}, {'LOWER': {'IN': ['mts','m','metros']}, 'OP': '?'} {'LOWER': {'IN': ['por','y','x']}}, {'LIKE_NUM': True} {'LOWER': {'IN': ['mts','m','metros']}, 'OP': '?'}]	8.66 x 26 8.66 x 26 m 17,32 mts y 26 mts
Corner	[{'LOWER': 'esquina'}]	Lote en importante esquina...
Barrio	[{'LOWER': {'IN': ['barrio', 'estancia', 'country', 'club']}}, {'POS':'PROPN', 'OP': '+'}]	Barrio Grand Bell
Facades amount	[{'RIGHT_ID': 'frentes', 'RIGHT_ATTRS': {'LOWER': {'IN': ['frentes', 'frente']}}}, {'LEFT_ID': 'frentes', 'REL_OP': '>', 'RIGHT_ID': 'NUM', 'RIGHT_ATTRS': {'DEP': 'nummod'}}]	2 frentes tres frentes
Swimming pool	[{'LEMMA': {'IN': ['piscina', 'pileta']}}]	El lote posee pileta

4.3 Transformers

Transformers is an efficient neural network architecture that uses an attention mechanism to optimally capture word dependencies [24]. BERT [11] is a pre-trained model based on the transformers architecture which can be used to perform different NLP tasks, such as Question-Answering (QA). QA systems generate an answer to a question given a certain context written in natural language [4]. The system should have the capability of recognizing whether the answer is present or not, and return "I don't know" in case of the latter. Models chosen are BERT-based that support the Spanish language, and are available in HuggingFace[6] platform:

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 all of these are BERT-based implementations, they have some differences. (1) BETO [9] is a BERT model trained on a big Spanish corpus. (2) DeBERTa [16] is a model that improves BERT and RoBERTa using disentangled attention and an enhanced mask decoder. mDeBERTa is the multilingual version of DeBERTa, which includes support for the Spanish language. (3) is the BETO model that uses distillation to reduce the complexity of the base model, maintaining a good performance. All of these models are trained over SQuAD2 dataset [21], to ensure that the model can answer "I don't know" when appropriate.

Given a set of desirable features to extract and a collection of real estate descriptions, a question is formulated for each feature, setting the description as the context. When the answer is not present, the models should return an empty answer.

Another rising technology is conversational models based on attention mechanisms. GPT-3 [8] is a large pre-trained model that uses reinforcement learning to acquire new data based on user input. GPT-3 generates an output given an input instead of extracting a sequence of text as the rest of the presented approaches do. Furthermore, the more precise the input format provided, the more accurate the response will be. Specifically, the provided input enumerates the required features to extract, and for each one it is specified the type of the variable and how to extract the mention (if it should extract an exact mention, or parse it into a specific format). The model is commanded to answer without diverging from the given guidelines.

5 Methodology

This section describes the ground truth dataset and the metrics used to evaluate the extraction approaches.

[6] `https://huggingface.co/`

5.1 Data

To evaluate the proposed approaches, a dataset with truthful values is needed, i.e., descriptions with the correct variables and values detected. This is known as *ground truth* dataset. It is created by hand from real estate descriptions on online advertisements, keeping in mind to cover different writing formats for each variable. The dataset[7] contains the values of the variables mentioned in the description, considering null those variables that are not present. As manual labeling data is an expensive task, only 100 advertisement descriptions are included in the actual ground truth dataset.

5.2 Metrics

According to several IE works, the evaluation is carried out using precision, recall and F1-score [15] to assess the performance of each approach over the ground truth dataset. F1-measure is a harmonic measure between precision and recall. Precision allows us to correctly identify attributes with their values. Recall measures the model's ability to identify all the true positives. In particular, to evaluate variables that have string values, partial matching is used to compare the predicted value with the expected one. This means that if the similarity is above 0.9, then the pair is annotated as a true positive.

6 Results

This section reports the results obtained for each approach to extract attribute-value pairs.

6.1 Extraction approaches evaluation

Rule-based matching To evaluate this approach, all defined patterns were applied to data. In the case of multiple patterns for a single variable, all of them are applied simultaneously and in case of multiple matches rules are defined for each case. Results are shown in Table 2.

Named Entity Recognition After the model is trained according to section 4.2, the model performed as shown in Table 3.

Transformers mDeBERTa was the best-scored BERT-based QA model. Results are summarized in Table 4, reporting metric results that each model obtained for each variable. Moreover, GPT-3 outperforms in almost all variables, as seen in Table 5.

[7] `https://www.kaggle.com/datasets/lucianatanevitch/real-estate-descriptions`

Table 2. Rule based matching performance by variable

Variable	Precision	Recall	F1 Score
address	0.37	0.90	0.53
fot	0.94	0.92	0.93
irregular	1.0	0.73	0.85
dimensions	0.95	0.59	0.73
corner	1.0	1.0	1.0
neighborhood	0.61	0.33	0.43
facades	0.81	0.45	0.58
pool	1.0	1.0	1.0

Table 3. NER Performance Metrics by Variable

Variable	Precision	Recall	F1 Score
address	0.79	0.63	0.7
fot	1.0	0.82	0.9
irregular	1.0	0.65	0.78
dimensions	0.94	0.63	0.75
corner	1.0	0.95	0.97
neighborhood	0.66	0.37	0.47
facades	0.88	0.38	0.53
pool	1.0	0.73	0.85

6.2 General results

Table 6 summarizes the F1-score that each model obtained in each variable. The best scores for each variable are highlighted. As can be seen, rule-based matching had the best results for boolean variables, while GPT-3 had the best results for string variables.

As expected, Rule-based detection of the lot in an intersection or having a swimming pool performed well because some keywords are always present when the feature is present in the text. On the other hand, address, facades' amount, and neighborhood name detection were performed with low F1-scores. This is probably because of the high variability in writing formats; consequently, it would require the development of several detection patterns. NER had good results, considering its low training data, misses can be related to this fact. In Transformers architecture, QA had low scores in several variables. While more rigorous testing has not been conducted, it's possible that the lack of precision could be attributed to the generality of the purpose for which the models were trained. In such a specific domain as evaluation, they might fall short. GPT-3 misses are related to incorrect inferences and format strings. For example, if the description doesn't mention at least two lot dimensions, GPT-3 should abstain from answering. In some descriptions, lot surface is present, but lot dimensions are not, and it was observed that sometimes GPT-3 retrieved a number that multiplied by another number yields that surface area. This is incorrect to assume, since the lot is not necessarily squared. Moreover, in address

Table 4. QA Performance Metrics by Variable

Variable	Precision	Recall	F1 Score
address	BETO: 0.17 mDeBERTa 0.37 Distilled: 0.25	BETO: 0.77 mDeBERTa 0.88 Distilled: 0.86	BETO: 0.27 mDeBERTa 0.53 Distilled: 0.38
fot	BETO: 0.36 mDeBERTa 0.8 Distilled: 0.22	BETO: 0.67 mDeBERTa 0.8 Distilled: 0.62	BETO: 0.47 mDeBERTa 0.8 Distilled: 0.33
irregular	BETO: 1.0 mDeBERTa 1.0 Distilled: 1.0	BETO: 0.43 mDeBERTa 0.6 Distilled: 0.65	BETO: 0.6 mDeBERTa 0.75 Distilled: 0.78
dimensions	BETO: 0.39 mDeBERTa 0.52 Distilled: 0.38	BETO: 0.77 mDeBERTa 0.92 Distilled: 0.78	BETO: 0.52 mDeBERTa 0.67 Distilled: 0.51
corner	BETO: 1.0 mDeBERTa 1.0 Distilled: 1.0	BETO: 0.15 mDeBERTa 0.55 Distilled: 0.4	BETO: 0.26 mDeBERTa 0.7 Distilled: 0.57
neighborhood	BETO: 0.41 mDeBERTa 0.52 Distilled: 0.41	BETO: 0.17 mDeBERTa 0.46 Distilled: 0.29	BETO: 0.25 mDeBERTa 0.48 Distilled: 0.34
facades	BETO: 0.5 mDeBERTa 0.73 Distilled: 0.61	BETO: 0.47 mDeBERTa 0.7 Distilled: 0.61	BETO: 0.48 mDeBERTa 0.71 Distilled: 0.61
pool	BETO: 1.0 mDeBERTa 1.0 Distilled: 1.0	BETO: 0.82 mDeBERTa 0.82 Distilled: 0.52	BETO: 0.9 mDeBERTa 0.9 Distilled: 0.68

Table 5. GPT-3 Performance Metrics by Variable

Variable	Precision	Recall	F1 Score
address	0.57	0.87	0.69
fot	0.92	0.97	0.94
irregular	0.66	0.95	0.78
dimensions	0.89	0.98	0.93
corner	0.76	1.0	0.86
neighborhood	0.66	0.92	0.77
facades	0.52	1.0	0.68
pool	1.0	1.0	1.0

detection, it tends to include the location which is not expected to be part of the address (because it's another variable) so it was computed as a false positive.

7 Conclusions

Rule-based matching is an old but powerful approach that doesn't need annotated data. Crafting efficient patterns requires knowing the syntactic structure of data, as the method is inherently biased towards the data it's based on. NER

Table 6. F1-scores obtained for each approach, in attribute-value extraction

General results (f1-score)								
	Address	**FOT**	**Irregular**	**Dimensions**	**Corner**	**Neighborhood**	**Facades**	**Pool**
Rule-based	0.53	0.93	0.85	0.73	1.0	0.43	0.58	1.0
NER	0.7	0.9	0.78	0.75	0.97	0.47	0.53	0.85
QA BETO	0.27	0.47	0.6	0.52	0.26	0.25	0.48	0.9
QA mDeBERTa	0.53	0.8	0.75	0.67	0.7	0.48	0.71	0.9
QA distilled	0.38	0.33	0.78	0.51	0.57	0.34	0.61	0.68
GPT-3	0.69	0.94	0.78	0.93	0.86	0.77	0.68	1.0

models are an effective but very expensive approach since they require a large corpus of annotated data. The annotation process requires a team of domain experts to ensure consistency in the task. Transformers are currently at the forefront of neural network architectures. There are various pre-trained models that were trained over huge datasets so that they can be used in general-purpose tasks. Moreover, abstractions such as *pipeline*[8] simplify the use of models for several tasks such as Question-Answering, as they enable manipulation without the need to consider intricate implementation details. Different input formats should be considered to reach the most accurate output, which means that questions should be written in different formats while comparing the given answers. GPT-3 is a huge pre-trained model with the capability of generating answers, compared to the Question-Answering models evaluated, which extract the answer from the context. GPT-3 holds the advantage of generating responses to questions based on context rather than merely extracting exact matches. This means it can produce boolean values; for instance, if the input mentions a "triangle shaped lot", it can detect irregularity, or if it states that the lot is located at an intersection, it can associate it with having two fronts. This work leaves several aspects open for future exploration. Once pairs are extracted, the next step is to align them with the existent ontology. The comparison of information extracted from descriptions against the structured data of an advertisement can reveal inconsistencies in the information. These inconsistencies may be partial (for example, if the *address* field in the structured data contains "Av. Montevideo 500", while the description extracts "Montevideo") or total (if the *irregular* field in the structured data is false, but the extraction from the description yields true). Addressing such discrepancies will require the verification and correction of the data.

Evaluated approaches could be improved. In approaches that extract exact mentions from text (Rule-based, NER, QA) it is possible to define rules based on NLP to recognize negative modifiers, especially for detecting the truth value of boolean variables. Moreover, in noisy descriptions, NLP techniques can be used to detect whether an extracted feature belongs to the property or not. NER could be improved by annotating more data, in addition to Question-Answering

[8] `https://huggingface.co/docs/transformers/main_classes/pipelines`

models which could be trained by fine-tuning, which means to give a pre-trained model a dataset with specific domain data. Finally, ensuring a high performance on the extraction models can be useful to derive new data based on detected variables.

On the other hand, data augmentation techniques could be considered in future works to generate larger datasets.

References

1. Automatic Extraction of Product Information, `https://dida.do/projects/numeric-attribute-extraction-from-product-descriptions`
2. ¿Qué es? | Observatorio de valores de suelo, `https://observatoriosuelo.gba.gob.ar/institucional/que-es`
3. Al-Amoudi, A., Alomari, A., Alwarthan, S., Rahman, A.: A rule-based information extraction approach for extracting metadata from pdf books. ICIC Express Letters **12**, 121–132 (02 2021). https://doi.org/10.24507/icicelb.12.02.121
4. Allam, A.M.N., Haggag, M.H.: The question answering systems: A survey. International Journal of Research and Reviews in Information Sciences (IJRRIS) **2**(3) (2012)
5. Anantharangachar, R., Ramani, S., S, R.: Ontology Guided Information Extraction from Unstructured Text. International journal of Web & Semantic Technology **4**(1), 19–36 (Jan 2013). https://doi.org/10.5121/ijwest.2013.4102
6. 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
7. Brinkmann, A., Shraga, R., Der, R.C., Bizer, C.: Product information extraction using chatgpt. ArXiv **abs/2306.14921** (2023)
8. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., 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.M., 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 (Jul 2020), arXiv:2005.14165 [cs]
9. Cañete, J., Chaperon, G., Fuentes, R., Ho, J.H., Kang, H., Pérez, J.: Spanish pre-trained bert model and evaluation data. In: PML4DC at ICLR 2020 (2020)
10. Del Río, J.P., Dioguardi, F., May, M., Torres, D.: Normalización y análisis exploratorio de datos inmobiliarios web. In: XI Jornadas de Sociología de la UNLP 5-7 de diciembre de 2022 Ensenada, Argentina. Sociologías de las emergencias en un mundo incierto. Departamento de Sociología. Facultad de Humanidades y Ciencias de la ... (2022)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
12. Dey Chowdhury, R., Sarkar, A., Banik, M., Bobbili, P.: Product attribute extraction and product listing analysis from e-commerce websites (06 2023). https://doi.org/10.13140/RG.2.2.11045.47842
13. Dioguardi, F., Torres, D., Antonelli, R.L., Río, J.P.d.: Construcción de un grafo de conocimiento para un observatorio inmobiliario. In: XXVIII Congreso Argentino de Ciencias de la Computación (CACIC)(La Rioja, 3 al 6 de octubre de 2022) (2023)

14. Ghani, R., Probst, K., Liu, Y., Krema, M., Fano, A.: Text mining for product attribute extraction. ACM SIGKDD Explorations Newsletter **8**(1), 41–48 (Jun 2006). https://doi.org/10.1145/1147234.1147241
15. Grishman, R.: Information extraction: Techniques and challenges. In: Information Extraction A Multidisciplinary Approach to an Emerging Information Technology: International Summer School, SCIE-97 Frascati, Italy, July 14–18, 1997. pp. 10–27. Springer (1997)
16. He, P., Liu, X., Gao, J., Chen, W.: Deberta: Decoding-enhanced bert with disentangled attention (2021)
17. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: state of the art, current trends and challenges. Multimedia Tools and Applications **82**(3), 3713–3744 (Jan 2023). https://doi.org/10.1007/s11042-022-13428-4
18. Linková, M., Gurskỳ, P.: Attributes extraction from product descriptions on e-shops. In: ITAT. pp. 23–26 (2017)
19. Mahlawi, A.Q., Sasi, S.: Structured data extraction from emails. In: 2017 International Conference on Networks Advances in Computational Technologies (NetACT). pp. 323–328 (2017). https://doi.org/10.1109/NETACT.2017.8076789
20. Probst, K.: Semi-Supervised Learning of Attribute-Value Pairs from Product Descriptions
21. Rajpurkar, P., Jia, R., Liang, P.: Know What You Don't Know: Unanswerable Questions for SQuAD (Jun 2018), arXiv:1806.03822 [cs]
22. 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
23. Sharma, A., Amrita, Chakraborty, S., Kumar, S.: Named entity recognition in natural language processing: A systematic review. In: Proceedings of Second Doctoral Symposium on Computational Intelligence: DoSCI 2021. pp. 817–828. Springer (2022)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need (Aug 2023), arXiv:1706.03762 [cs]
25. Vijayarajan, V., Dinakaran, M., Lohani, M.: Ontology based object-attribute-value information extraction from web pages in search engine result retrieval. In: Kumar Kundu, M., Mohapatra, D.P., Konar, A., Chakraborty, A. (eds.) Advanced Computing, Networking and Informatics- Volume 1. pp. 611–620. Springer International Publishing, Cham (2014)
26. 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
27. Zheng, G., Mukherjee, S., Dong, X.L., Li, F.: OpenTag: Open Attribute Value Extraction from Product Profiles [Deep Learning, Active Learning, Named Entity Recognition]. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1049–1058 (Jul 2018). https://doi.org/10.1145/3219819.3219839, `http://arxiv.org/abs/1806.01264`, arXiv:1806.01264 [cs, stat]
28. Zou, H.P., Yu, G.H., Fan, Z., Bu, D., Liu, H., Dai, P., Jia, D., Caragea, C.: Eiven: Efficient implicit attribute value extraction using multimodal llm (2024)