Relation extraction in structured and unstructured data: a comparative investigation on smartphone titles in the e-commerce domain

João Gabriel Melo Barbirato[1], Livy Real[2], Helena de Medeiros Caseli[1]

[1] Federal University of São Carlos – Computing Department
Caixa Postal 676 – 13565-905 – São Carlos – SP – Brazil

[2]Digital Lab – americanas s.a.
São Paulo SP, Brazil

jgmbarbirato@estudante.ufscar.br, helenacaseli@ufscar.br

livy.coelho@b2wdigital.com

Abstract. *As large amounts of unstructured data are generated on a regular basis, expressing or storing knowledge in a way that is useful remains a challenge. In this context, Relation Extraction (RE) is the task of automatically identifying relationships in unstructured textual data. Thus, we investigated the relation extraction on unstructured e-commerce data from the smartphone domain, using a BERT model fine-tuned for this task. We conducted two experiments to acknowledge how much relational information it is possible to extract from product sheets (structured data) and product titles (unstructured data), and a third experiment to compare both. Analysis shows that extracting relations within a title can retrieve correct relations that are not evident on the related sheet.*

1. Introduction

The main purpose of extracting information from text is to transform it into useful and well-structured knowledge [Pawar et al. 2017]. This can be done by means of well-known Natural Language Processing (NLP) tasks such as named-entity recognition, Information Extraction (IE) or Relation Extraction (RE).

Relation Extraction consists in automatically identifying relations in unstructured textual data [Pawar et al. 2017]. In the general domain, relationships instantiate facts with a high probability of being true (or highly plausible) [Xu et al. 2020]. But relation extraction in specific domains is also challenging, due to factors such as the higher variability of vocabulary, noisy and missing data, and the lack of standardization that is common in real scenarios. To exemplify this, next we show three product titles, in Portuguese, found on americanas.com:[1]

S1 `smartphone multilaser ms40s preto 4" câmera 3 mp + 5 mp 3g quad core 8gb android 6.0 p9025`
S2 `smartphone samsung galaxy s5 sm g900m branco tela 5.1", android 4.4, 4g, câmera 16mp`
S3 `celular positivo 2.4" 3g bluetooth fm mp3 p30c preto`

[1]Extracted in November 2020.

These are three different products in the smartphone (and cellphone) category. From these examples it is possible to identify three different brands (`multilaser`, `samsung` and `positivo`), two colors (`preto` and `branco`), two versions of the operating system Android (`6.0` and `4.4`) and two different camera resolutions (`3 mp + 5 mp` and `16 mp`). These are examples of product properties that could give rise to binary relations with the item being offered (the smartphone or cellphone).

In this context, this work aims to investigate how relations that were automatically extracted from unstructured data using BERT [Devlin et al. 2019] can enhance the information extracted from structured data. Bidirectional Encoder Representations from Transformers (BERT) is an encoder architecture capable of applying transfer learning for downstream NLP tasks through the fine-tuning process [Devlin et al. 2019]. In [Soares et al. 2019], the authors show that the encoder can also be used for RE from a corpus annotated with relations of interest. Thus, in this paper we present some experiments carried out with BERT Relation Extraction[2] to extract binary relations from e-commerce data.

The main contributions of this work are: (i) two BERT models fine-tuned to extract relations from Portuguese product titles in the smartphone/cellphone category; and (ii) a comparison between the extracted data showing how unstructured data can complement structured information.

This document is divided into five sections. Section 2 presents related work; Section 3 describes how the RE models were generated and evaluated, and discusses the results; Section 4 compares the extracted instances with a corpus built from structured data. Section 5 finishes this paper with some conclusions and proposals for future work.

2. Related Works

The Relation Extraction (RE) task consists in extracting well-defined relationships between two entities [Pawar et al. 2017] and saving them into a structured repository [Moens 2006, Sarawagi 2008]. Hearst [Hearst 1992] proposes lexical-syntactical patterns to identify relations. The ACE program [Doddington et al. 2004] aims to analyze other aspects in sentences, such as the occurrence of words and lexical categories. Over time, many works also considered named-entity recognizer models as a crucial part of the RE task [Sarawagi 2008] and vice-versa [Ji and Grishman 2006]. The task also became a subject of research in Machine Learning (ML) and NLP, where the main investigated approaches were Support Vector Machines [Zitouni and Florian 2008] and Conditional Random Fields [Li et al. 2011].

More recent studies showed promising results to RE using deep neural networks, such as Convolutional Neural Networks [Zeng et al. 2014] and Recursive Neural Networks [Socher et al. 2012, Hashimoto et al. 2013]. Deep contextualized language models, such as BERT [Devlin et al. 2019], have gained attention in ML and NLP tasks [Peters et al. 2018, Radford et al. 2018, Devlin et al. 2019], such as "Question Answering" [Devlin et al. 2019] and RE [Soares et al. 2019]. Thus, this work explores a fine-tuned BERT architecture for RE, as will be described in the next sections.

[2] `https://github.com/plkmo/BERT-Relation-Extraction`

2.1. Relation Extraction with BERT

The Bidirectional Encoder Representations From Transformers (BERT) [Devlin et al. 2019] is an encoder architecture for generating contextualized language models. The model is versatile, able to understand context on the left and right to solve various NLP tasks, such as Next Sentence Prediction, Question Answering and Sentiment Analysis [Devlin et al. 2019].

In [Soares et al. 2019] the authors used BERT to represent relations via training following the matching the blanks (MTB) approach. By applying BERT to the task of extracting binary relations between entities, the authors start from a corpus of blocks of text containing two marked entities as illustrated in Table 1.

Table 1. Examples of marked entities and its substitution to "blanks". Adapted from [Soares et al. 2019]

r_A	In 1976, e_1 (then of Bell Labs) published e_2 , the first of his books on programming inspired by the Unix operating system.
r_B	The "e_2" series spread the essence of "C/Unix thinking" with makeovers for Fortran and Pascal. e_1's Ratfor was eventually put in the public domain.
r_C	e_1 worked at Bell Labs alongside e_3 creators Ken Thompson and Dennis Ritchie.
Mentions	e_1 = Brian Kernighan, e_2 = Software Tools, e_3 = Unix

Henceforth, the training set is created by replacing the entity with a special symbol `[BLANK]` in order to predict the hidden entity. The symbol is introduced probabilistically to ensure that the model learns the relationship not only by the entities, but by the words around them. This process was called "matching the blanks". For the authors, MTB training aims to solve the data redundancy problem observed in texts on the web, where an arbitrary pair of entities is probably mentioned several times throughout a sequence.

The authors propose a representation method called *entity markers*: given a sequence of tokens, starting with token `[CLS]` and ending with `[SEP]`, the tokens that mention a certain entity are delimited. For this, they used the $\text{BERT}_{\text{LARGE}}$ pre-trained model and Wikipedia in English as the training corpus, with interconnected paragraph blocks. In their experiments with the MTB method, the authors observed an F-score value of 89.5%, better than the 71.5% value that was observed for the TACRED [Zhang et al. 2017] relation prediction model on the SemEval 2010 dataset. In addition, the MTB obtained 89.2 10-way 1-shot[3] on the FewRel dataset, against 94.3% obtained from humans. Finally, it is worth mentioning that there is an open implementation of this work[4].

3. Experiments and Results

This section describes datasets, experiments and results. We used a dataset of products from the smartphone category (smartphones and cellphones)[5]. This dataset has instances

[3]This is a training method which contains 1 instance of a single class between 10 of them.
[4]`https://github.com/plkmo/BERT-Relation-Extraction/`
[5]This category was chosen because of its high demand on e-commerce platforms.

of structured information in product sheets (as shown in Figure 1) as well as unstructured information in product titles and descriptions (as shown in Figure 2)[6].

Código	132152081
Código de barras	7898573294772
Marca	ASUS
Modelo	ZC553KL-4I092BR
Cor	Rosa
Tipo de Chip	Micro Chip
Quantidade de Chips	Dual Chip

Figure 1. Example of a product's data sheet

Figure 2. Example of a product's title and description

This entire dataset contains 956 products from the smartphone category. It was separated in two sets: (i) one with 540 items with structured information (product sheets) and (ii) one with 416 product titles annotated with entities and binary relations.

Product sheets – From the 540 products, 77 different properties were recovered from their data sheets. Not all products have all properties. For example, the property called "*garantia do fornecedor*" (vendor guarantee) is present in all 540 products, while the property called "*conexões*" (connections) is only present in 201 products.

Annotated titles – 416 product titles were annotated using the Prodigy[7] tool by 2 linguists[8], who marked the following entities: `Model`, `Brand`, `Color`, `Internal_memory`, `Camera`, `Display_size`, `Chip_capacity`, `OS` (operating system) and `Processor`. Thus, each mention of a `Model` (subject) entity and an entity of another type (object) in the same title (that is, each pair of marked entities) becomes an instance of a binary relation of interest in the dataset. Examples of such relations include `has_brand(Model, Brand)` and `has_color(Model, Color)`. A total of 8 different relations were identified.

3.1. Experiments

Experiments were designed to answer the following research questions using the two datasets:

Q1 – How much relational information is it possible to extract from product sheets?
Q2 – How much relational information is it possible to extract from product titles?
Q3 – How complementary is the relational information extracted from titles to the one extracted from the product sheets?

To answer **Q1**, Subject-Predicate-Object (SPO) triples were constructed using properties extracted from the product sheets as well as their respective values. Therefore, the following design was adopted:

[6] `https://www.americanas.com.br/`. Last access: June 2021
[7] `https://prodi.gy/`
[8] Discrepancy cases were resolved by a third linguist, although the agreement rate between the annotators was above 72%.

- **Subject entity** – this is the value of a `Model` entity. If the product's sheet did not contain this attribute, a Named-Entity Recognizer (NER) trained in the e-commerce domain was used to recognize the `Model` entity from the product title. This NER was generated by another team linked to the partnership with americanas s.a.
- **Relation label** – this is one of the 8 relations of interest.
- **Object entity** – this is the value of the corresponding property in the product sheet. For example, `Full HD - 1920x1080` or `5.2"` may be values for the `has_display_size` relation. Similarly, `Android` is a possible value for the `has_os` relation.

In order to answer **Q2**, we trained the MTB [Soares et al. 2019] approach on product titles annotated with entities and relations. Following an implementation of MTB[9], each instance used in the model's fine-tuning consists of: (1) a sentence (in the case of this experiment, a product title) with two marked entities and (2) the label of the relation between them. The annotated titles dataset was split into training, validation and test partitions as detailed in Table 2.

Table 2. Relation instances on smartphone dataset and their distribution into training, validation and testing sets.

	train	valid	test	**total**
`has_brand`	199	103	103	405
`has_camera`	108	70	53	231
`has_chip_capacity`	124	63	66	253
`has_color`	170	89	92	351
`has_display_size`	117	67	68	252
`has_internal_memory`	127	77	73	277
`has_os`	68	39	40	147
`has_processor`	18	9	8	35
Total	931	517	503	1951

The original source code was adapted[10] to use models that are capable of dealing with Brazilian Portuguese:

- **BERTimbau**[11] [Souza et al. 2020] – this is a trained BERT model for Brazilian Portuguese based on web documents from various domains.
- **Multilingual BERT**[12] [Devlin et al. 2019] (mBERT) – this is a BERT model trained for more than 100 languages, including Portuguese, based on Wikipedia content[13].

These models were trained with batch size 128, MTB learning rate 10^4 and fine-tuning learning rate 7×10^5 (as suggested by the original implementation). Both models trained MTB within 18 epochs (approximately 3 days each model), while requiring 60 and 65 epochs (approximately 2 hours each model) to fine-tune BERTimbau and mBERT,

[9] `https://github.com/plkmo/BERT-Relation-Extraction`
[10] `https://github.com/joaobarbirato/BERT-Relation-Extraction`
[11] `https://huggingface.co/neuralmind/bert-base-portuguese-cased`
[12] `https://huggingface.co/bert-base-multilingual-uncased`
[13] More details on Multilingual BERT training are available at `https://github.com/google-research/bert/blob/master/multilingual.md`

respectively. All training steps were performed on a 40 core Intel(R) Xeon(R) Silver 4210 CPU 2.20GHz machine.

Finally, regarding **Q3**, a third experiment was carried out to compare the information extracted from structured (**Q1**) and unstructured (**Q2**) data. The same NER model used on **Q1** was used to process the 540 titles corresponding to each product used for **Q1** to automatically mark entities. These marked titles served as input to the MTB BERTimbau model for inferring the relations.

3.2. Results

To answer **Q1**, $2,825$ model-attribute-value triples were extracted from the 540 product sheets. Table 3 shows some examples of relation instances extracted from product sheets. From the extracted relations it is possible to see that there is still room for improvement. For example, entities `Moto G (3ª Geração)` and `Moto G 3` were considered as different entities. Disambiguating entities is one possible solution to such problems.

Table 3. Examples of relation instances extracted from the product sheet dataset.

Relation	Subject	Object
`has_internal_memory`	SM-N975F/2DL	256gb
`has_color`	ZC554KL-4A115BR	preto
`has_display_size`	Galaxy S8	5.8”
`has_camera`	Moto G (3ª Geração)	13mp

To answer **Q2**, from the 503[14] instances in the test set, MTB models trained using BERTimbau and multilingual BERT (mBERT) correctly extracted, respectively: 378 and 376 instances. On average, the model trained using BERTimbau performed better regarding the F-score values, with 3.41 percentage points more than mBERT, as shown in Table 4. Indeed, in [Souza et al. 2020] the authors pointed out a similar difference between the F-score values for BERTimbau and mBERT.

Regarding **Q3**, the model from **Q2** was applied to the same dataset as **Q1** in order to compare the information extracted from structured and unstructured data. From the 540 items in the product sheet dataset, we processed the product titles to generate $4,933$ inputs for the model trained with BERTimbau infer the relation instances. Since different titles can generate the same relation instance, from these titles, BERTimbau output $2,575$ distinct triples. Comparing the extracted triples with the entities identified by the NER model we noticed that $2,072$ were equal. We considered these as the correct ones although this decision may be ignoring the NER errors. Table 4 shows detailed results for each model, relation and research question.

The results regarding **Q2** indicate the applicability of BERT Relation Extraction to extract binary relations from product titles. The model trained using BERTimbau was selected to be used in our third experiment due to its very good F1-score (almost 94%).

One of the main reasons for the worse result in the experiment related to **Q3** compared to the one regarding **Q2** are the differences in quality and standardization between

[14]It is worth mentioning that different titles can generate the same relation instance. Of 503 product titles, BERTimbau and mBERT output 405 and 407 distinct relation instances, respectively.

Table 4. Evaluation values (%) (a) in test sets for the MTB models and (b) in Q1 dataset using the MTB BERTimbau trained model

		(a) **Q2**				(b) **Q3**		
		MTB BERTimbau		**MTB mBERT**			**MTB BERTimbau**	
Relation	**Support**	**Accuracy**	**F1**	**Accuracy**	**F1**	**Support**	**Accuracy**	**F1**
has_processor	8	87.50	**93.33**	62.50	66.67	476	50.84	66.30
has_os	40	90.00	92.31	90.00	**93.51**	605	77.85	79.63
has_internal_memory	73	100.00	97.99	100.00	**99.32**	15	80.00	4.57
has_display_size	68	89.71	94.57	92.65	**96.18**	759	74.18	83.90
has_color	92	98.91	**94.79**	97.83	91.37	645	94.26	84.04
has_chip_capacity	66	89.39	89.39	92.42	**93.85**	589	78.95	84.16
has_camera	53	100.00	**96.36**	100.00	95.50	1101	92.28	93.81
has_brand	103	90.29	**92.08**	85.44	87.13	743	75.24	81.84
Mean_{micro}	-	93.23	**93.85**	90.10	90.44	-	77.95	72.28

these two datasets. The titles used for **Q1** follow stricter standardization rules and quality requirements, as they refer to products sold by a single large e-commerce company. The titles used for the NER model training were provided by a diverse set of small sellers, and therefore are noisier and less standardized. We believe that this difference in data was responsible for the poor performance of the NER in this new dataset. We manually observed that the NER tagged many false instances of `Model`, which could have drastically affected many predicted relation instances.

4. Qualitative Analysis

In this section we compare the relation instances extracted from both datasets (structured and unstructured) to better understand how different and complementary are the triples extracted from them by comparing, respectively, results from **Q1** with **Q2** and **Q1** with **Q3**; thus answering **Q3**. Numbers verified in both analysis were obtained using set operations in code.

Table 5 quantifies the amount of instances extracted (**Q2** vs **Q1** – Different) and inferred (**Q3** vs **Q1** – Complementary). Columns (a) and (c) quantify the instances present only in **Q2** and **Q3**, respectively. The other columns quantify the instances that were present both in **Q2** and **Q1** (b) and **Q3** and **Q1** (d).

How different are they? From the 405 relation instances predicted by the BERTimbau model in **Q2**, 378 (approximately 93%) were correct. It was verified, then, how many of these correctly extracted instances were equal to the ones extracted from the product sheet dataset. Only 11 common instances were found. Consequently, about 97% of the correctly predicted instances (367 instances) are correct and new. In other words, it is possible to derive a lot of correct information from product titles that are not yet available in product sheets.

How complementary are they? Based on this information, it is possible to identify how the information in product titles complements the information found in product sheets. Only 202 (9.75%) of the $2,072$ correctly inferred triples in **Q3** were extracted from product sheets. Consequently, about 90.25% of the correctly predicted instances ($1,870$ instances) are correct and new. In other words, we again conclude that it is possible to derive a lot of correct information from product titles that are not yet available in product sheets.

Table 5. Amount of instances retrieved in the product sheets (Q1) in comparison with instances extracted by the BERTimbau model (Q2 and Q3)

	Q2 vs Q1 – Different		**Q3 vs Q1 – Complementary**	
Relation	**Only Q2** (a)	**Q2** $\cap$ **Q1** (b)	**Only Q3** (c)	**Q3** $\cap$ **Q1** (d)
`has_color`	74	2	406	73
`has_brand`	67	3	199	58
`has_internal_memory`	56	1	9	-
`has_display_size`	48	-	296	5
`has_chip_capacity`	46	-	206	1
`has_camera`	42	3	381	57
`has_os`	28	2	262	8
`has_processor`	6	-	111	-
Total	367	11	1,870	202

5. Conclusion

In this paper we investigated relation extraction from structured and unstructured data for the e-commerce domain using a BERT model fine-tuned for this task. We concluded that the fine-tuned model using BERTimbau performs a little better than the one based on Multilingual BERT. We compared how different and complementary are the information extracted from product titles and the structured information present in product sheets.

Experiments showed that about 97% of the relation instances extracted from an external dataset and 90.25% of the triples extracted from the same source were correct and new, i.e. not present in product sheets. From these experiments, we can conclude that processing unstructured data from product titles, which is much more abundant and easier to collect, is a promising approach for generating structured data that can be useful for a variety of e-commerce applications such as filtering and recommendation.

From the qualitative analysis, it is clear that the automatic relation extraction in a corpus of unstructured data composed of product titles contributes towards constructing a relation instance corpus. Evidently, the information on e-commerce is incomplete and the MTB method contributes to the completion of entity linkages.

As future work, it is possible to optimize MTB training hyperparameters, as this was not done due to implementation difficulties, integration with BERT models for Portuguese and training time. We also intend to use the extracted relation instances to build a knowledge graph (KG) and study its effectiveness in tasks for the e-commerce domain, such as product recommendation and search. The results presented in this paper support this idea, since most of the instances extracted by the MTB models were not in the base KG, which was built from structured data. This analysis shows that the relation extraction can help with the knowledge graph completion problem.

Acknowledgments

This paper and the research behind it would not have been possible without the support of americanas s.a. Digital Lab, specially José Pizani and Ester Campos, who closely followed this research. This work is is part of the project "Dos dados ao conhecimento: extração e representação de informação no domínio do e-commerce" (Projeto de extensão - UFSCar #23112.000186/2020-97).

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M., and Weischedel, R. M. (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Hashimoto, K., Miwa, M., Tsuruoka, Y., and Chikayama, T. (2013). Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1372–1376.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Ji, H. and Grishman, R. (2006). Analysis and repair of name tagger errors. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 420–427.

Li, Y., Jiang, J., Chieu, H. L., and Chai, K. M. A. (2011). Extracting relation descriptors with conditional random fields. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 392–400.

Moens, M.-F. (2006). *Information extraction: algorithms and prospects in a retrieval context*, volume 21. Springer Science & Business Media.

Pawar, S., Palshikar, G. K., and Bhattacharyya, P. (2017). Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*.

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

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding with unsupervised learning.

Sarawagi, S. (2008). Information extraction. *Found. Trends Databases*, 1(3):261–377.

Soares, L. B., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Socher, R., Huval, B., Manning, C. D., and Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., and Achan, K. (2020). Product knowledge graph embedding for e-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 672–680, New York, NY, USA. Association for Computing Machinery.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2335–2344.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Zitouni, I. and Florian, R. (2008). Mention detection crossing the language barrier. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 600–609.