NLP Proof of Concept
E-commerce products
MOP

Paweł Golik, Mateusz Jastrzebiowski and Aleksandra Muszkowska
Warsaw University of Technology
pawel.golik.stud@pw.edu.pl,
mateusz.jastrzebiowski.stud@pw.edu.pl,
aleksandra.muszkowska.stud@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

1 Dataset and EDA

We focus on Web Data Commons - Training Dataset and Gold Standard for Large-Scale Product Matching dataset (WDC for short) prepared by the staff of the University of Mannheim (Primpeli et al., 2019). The dataset contains offers in four categories - Cameras, Computers, Watches, and Shoes. Additionally, each offer is linked to a specific product (cluster_id) and contains textual attributes such as title, description etc. Each observation is a pair of such offers and a label indicating whether these two offers are for the same product (a positive pair) or not (a negative pair). Even in the case of a negative pair, both offers belong to the same category (but different clusters/products).

The training datasets are available in different sizes, varying from small to extra large. In every dataset, the ratio between positive and negative pairs is 1:3. Table 1 presents the exact sizes for each dataset. The proportions between the different collections within one category are as follows: 1 – small, 3 – medium, 15 – large, and 50 – extra-large (xlarge). We do not consider the extra-large datasets as our computational and time resources are limited.

The Gold Standard is verified manually and should be used for testing purposes. Each product contains highly similar negative pairs (complex cases) and less similar negative pairs (easy cases). Table 2 depicts the statistics for the Golden Standard dataset per each category.

The WDC dataset has already been used for product-matching tasks (Możdżonek et al., 2022) and the results of this research is depicted in Table 3.

Table 1: WDC datasets sizes. Source: (Peeters et al., 2022)

Category	Size	Positive	Negative	Total
Cameras	Small	486	1,400	1,886
	Medium	1,108	4,147	5,255
	Large	3,843	16,193	20,036
	xLarge	7,178	35,099	42,277
Computers	Small	722	2,112	2,834
	Medium	1,762	6,332	8,094
	Large	6,146	27,213	33,359
	xLarge	9,690	58,771	68,461
Watches	Small	580	1,675	2,255
	Medium	1,418	4,995	6,413
	Large	5,163	21,864	27,027
	xLarge	9,264	52,305	61,569
Shoes	Small	530	1,533	2,063
	Medium	1,214	4,591	5,805
	Large	3,482	19,507	22,989
	xLarge	4,141	38,288	42,429

2 Machine Learning Models

2.1 Encoder architectures

We use state-of-the-art architecture to test product-matching methods. Citing the work of (Możdżonek et al., 2022), we test the pre-trained mBERT and XLM-RoBERT models on a WDC dataset with two output classes to distinguish corresponding and non-corresponding pairs of the offer. To do this, we will use a pre-trained BERT model and then perform fine-tuning on our product matching task.
In contrast to the work of (Możdżonek et al., 2022), we will use not only the titles of the offerings but also attribute values, descriptions, attribute names, and units.

Table 2: Gold Standard Statistics per category. Source: (Primpeli et al., 2019)

Category	#positive	#negative	#combined	title	s:description	spec Table
Computers	150	400	550	100%	88%	21%
Cameras	150	400	550	100%	79%	5%
Watches	150	400	550	100%	77%	5%
Shoes	150	400	550	100%	88%	3%

Table 3: Current state-of-the-art F1 scores in product matching task for models trained on English WDC datasets. Mean value and standardized error (confidence level 95%) for each dataset were calculated from 4 samples. Source: (Możdżonek et al., 2022)

Category	Size	mBERT [x]	XLM-RoBERTa [y]	Ditto [z2]	WDC-Deepmatcher [z]
Cameras	Small	82.13(±4.70)	81.96(±7.75)	80.89	68.59
	Medium	87.86(±2.04)	88.11(±4.22)	88.09	76.53
	Large	90.88(±2.28)	92.36(±0.76)	91.23	87.19
	xLarge	-	-	93.78	89.21
Computers	Small	86.43(±3.69)	81.10(±13.40)	80.76	70.55
	Medium	90.13(±1.89)	88.69(±2.19)	88.62	77.82
	Large	92.48(±2.33)	93.71(±0.77)	91.70	89.55
	xLarge	-	-	95.45	90.80
Watches	Small	79.20(±7.89)	74.98(±13.36)	75.89	73.86
	Medium	84.11(±3.40)	81.30(±8.21)	82.66	79.48
	Large	90.28(±2.36)	91.26(±2.09)	88.07	90.39
	xLarge	-	-	90.10	92.61
Shoes	Small	87.31(±1.64)	83.78(±4.38)	85.12	66.32
	Medium	91.17(±4.21)	89.50(±3.69)	91.12	79.31
	Large	93.52(±2.63)	93.62(±0.67)	95.69	91.28
	xLarge	-	-	96.53	93.45

2.2 Extracting Embeddings

After training the model, our next goal is to extract each offer's embeddings for probing and similarity measures. To do this, we evaluate and then extract the embedding of each token. There are usually more tokens than words in the input description, so we match the appropriate tokens to each offer. As a result, we have a matrix of embeddings for each compared offer.

2.3 Similarity measures

After extracting the embeddings of each offer, we propose testing the similarity of embeddings. To do this, we will use cosine metric, which behaves very well with high dimensional embeddings.
Moreover, we will test if it is possible to classify new offers by calculating their embeddings and comparing them with embeddings from the training dataset.

To test our hypothesis, we need to train our model on the whole dataset, which is time-consuming. Therefore we will provide all results after model training.

2.4 Probing

The next task is to explain the model and find the relationship between the embeddings and the input data. To this end, we propose to use the probing method. Its task is to examine what information is contained in the word embeddings created by the NLP model.
In our case, the probing classifier's input will be the offers' embeddings. We will examine whether they have learned a relationship with certain properties and check whether they contain information unrelated to the task.
It is worth to mention that probing tasks are not only to determine which features are taken

into consideration by the model but also which features aren't e.g. the length of concatenated title and description shouldn't be taken into consideration as it is not a good feature to decide if the offers are of the same product.
Our aim is to determine which features our solutions takes into consideration. As previously mentioned we would like to check if the length of the title and description is taken into consideration to be certain that the results we get from our solution are not accidental (because e.g. descriptions of the same product are similar in length). The next probing task we try to develop is checking if the model focuses of some of the specific words such as RAM or CPU (in case of computers).
We also think that it would be interesting to determine if the values of certain properties of products such as the number of cores in CPU or the amount of gigabytes of RAM are important for model to be able to distinguish the offers.

3 Reference to the reviews

In the reviews, apart from the comments about the presenting methods (such as the unclear explanation of the BERT model or too much text on slides), we encountered a general misunderstanding of probing tasks. We should have provided more explanation for the purpose of our project, i.e., we are not trying to prove that probing is the most suitable tool for black-box models explanation. We only wanted to examine the embeddings, measure their similarity and try applying probing tasks to test different hypotheses about the embedded space. We consider that the probing may eventually be inaccurate, and the results will not confirm the assumed hypotheses.

Another problem depicted is that we use only one dataset. Even if we get any results from probing, their meaning will be limited only to the specific dataset. On the other hand, probing tasks have to be tailored to a given problem (dataset). By choosing only one dataset, we wanted to focus more on the domain aspect and consider different probing tasks. Additionally, we are limited in terms of computational power and time resources. Nevertheless, trying another dataset is a good idea.

References

[Belinkov 2021] Yonatan Belinkov. 2021. *Probing Classifiers: Promises, Shortcomings, and Advances*. Computational Linguistics, 48(1):207–219.

[Lindström et al. 2020] Lindström, Adam & Björklund, Johanna & Bensch, Suna & Drewes, Frank. 2021. *Probing Multimodal Embeddings for Linguistic Properties: the Visual-Semantic Case*. *In Proceedings of the 28th International Conference on Computational Linguistics*, pages 730–744, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[Şahin et al. 2020] Şahin, Gözde Gül and Vania, Clara and Kuznetsov, Ilia and Gurevych, Iryna 2020. LINSPECTOR: Multilingual Probing Tasks for Word Representations. *Computational Linguistics*. **46**, 335-385 (2020,6), https://doi.org/10.1162/coli

[Możdżonek et al.2022] Możdżonek, Michał & Wróblewska, Anna & Tkachuk, Sergiy & Łukasik, Szymon 2022. *Multilingual Transformers for Product Matching – Experiments and a New Benchmark in Polish.*. 1-8. 10.1109/FUZZ-IEEE55066.2022.9882843.

[Duffner et al.2021] Duffner, Stefan & Garcia, Christophe & Idrissi, Khalid & Baskurt Atilla 2021. *Similarity Metric Learning. Multi-faceted Deep Learning - Models and Data*

[Tracz et al.2020] Tracz, Janusz & Wójcik, Piotr Iwo & Jasinska-Kobus, Kalina & Belluzzo, Riccardo & Mroczkowski, Robert & and Gawlik, Ireneusz 2020. *BERT-based similarity learning for product matching. In Proceedings of Workshop on Natural Language Processing in E-Commerce, pages 66–75, Barcelona, Spain. Association for Computational Linguistics.*

[Primpeli et al.2019] Primpeli, A., Peeters, R., & Bizer, C. 2019. *The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. Companion Proceedings of The 2019 World Wide Web Conference.*

[Peeters et al.2022] Peeters, Ralph & Bizer, Christian 2022. *Cross-language learning for product matching. WWW Companion, 2022a*