E-commerce products
Project Report for NLP Course, Winter 2022

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

Abstract

In this project, we explored modern methods used in the product matching problem, which is a generalisation of the entity matching problem. These tools are mainly based on deep neural networks that encode individual offerings into vectors called embeddings representing specific knowledge. We analysed the embedded space and attempted to develop more probing tasks to investigate the properties of embeddings in this domain. We have expanded the previous project with new probing tasks and extensive comparative tests of various architectures and datasets. We compared the pre-train and fine-tuned models. Moreover, tests were conducted on a new natural dataset to ensure comprehensive testing. In addition, we checked whether encoded vectors of offers for the same products would show high similarity and, on the contrary, whether offers of unrelated products would not be similar. Our work helps understand black-box knowledge representations (embeddings) and sheds light on the similarity properties of embedded vectors. Moreover, our research adds to the knowledge of how language models process and encode information and how this understanding can be utilised to enhance the performance of natural language processing tasks.

1 Introduction

The e-commerce sector has seen much growth in recent years, compounded by the global coronavirus pandemic. Customers were previously limited to the offerings of local sellers. However, the growth of the Internet and delivery services can open up new sources of goods provided by e-commerce sites such as Allegro.pl and Amazon. The vast number of offers published daily by vendors leads to new challenges in efficiently finding offers of potential interest to customers. Unfortunately, many offers are presented in different formats and with different variations of product names which makes it challenging to build automated tools for matching offers of the same product. Having a tool capable of comparing two different offers with each other would allow for solving many problems, such as offers matching, but also suggesting similar offers in the absence of offers related to the selected product or detecting offers misclassified by vendors as a given product based on too little similarity to other, valid offers.

Relying on the traditional comparison of strings representing offers is not reliable because descriptions and titles of offers can be formed in many different ways (different order of a product number, brand, and name) or contain many additional words that do not help to distinguish the product (e.g., 'best-seller,' 'offer,' etc.). For this reason, many modern tools are based on transformer models, which encode input information as vector embeddings. These solutions are characterised by high efficiency, but they are so-called black boxes from which it is not easy to deduce which features are essential for making predictions. Constructing tools to examine the coded forms of offers would allow a better understanding of these methods. For this purpose, we propose probing tasks examined in the work described in the article. We conducted extensive experiments to understand the nature of embedding space using different models and on different datasets.

Section 2 presents articles related to e-commerce product matching architecture and probing tasks. Section 3 is devoted to describing the datasets we used in our project. Section 4 consists of a description of the research and experiment methodology. We describe the method of obtaining embeddings, fine-tuning the model and

creating probing tasks. Section 5 is devoted to experiments, their results, conclusions and interpretation.

2 Related Work

Modern state-of-the-art product matching methods rely on deep learning techniques using Transformer models, which allow for creating embeddings from input representing specific knowledge about the encoded entities (Możdżonek et al., 2022), (Tracz et al., 2020). The embeddings map words to real-valued vectors, which reveal semantic aspects, for example, if words are related in meaning or belong to the same topic. Creating such an embedding means enriching as well as filtering out information. As far as we know, most of the research in product matching focuses on building classifiers characterised by extracted embedding representation (Możdżonek et al., 2022). In the training phase, the encoder learns to transform the input into the embedding space, which serves well for the classification task. A slightly different approach is presented in (Tracz et al., 2020), where the embeddings are directly compared and passed to the Loss function assessing their similarity. The Loss function is then minimized for the embedded inputs, which causes the weights of the encoder to change accordingly. This approach may ensure more suitable embedding space for future examininig of the similarity of the embeddings as it already imposes a similarity constraint during the training phase.

2.1 Encoder architectures

For e-commerce product matching one can use a Cross-encoders (Wolf et al., 2019) architecture e.g. solution proposed in (Możdżonek et al., 2022). This architecture allows to get high accuracy of matching at the expense of the computation time. Cross-encoders based architectures also require a lot of data to fine-tune and it is impossible to extract embeddings for each sentence provided as an input as the architecture uses only one Bert model to calculate one aggregated embedding for the input.

To solve these problems a Bi-encoders architecture was proposed. Unlike Cross-encoders it uses two separate Bert models and pooling for embeddings calculation. Such an approach enables the user to calculate and easily extract separate embeddings (which are integral part of the architecture) for each sentence provided as an input. To asses the similarity of offers, Bi-encoders use similarity measure e.g. cosine-similarity measure. Overall, Bi-encoders are faster and require less data for fine-tuning than Cross-encoders but obtain lower accuracies.

2.2 Probing tasks

As far as we know, we would be the first to describe probing tasks for embeddings in the product-matching domain. Probing has been extensively described in (Şahin et al. , 2020), but it focuses mainly on probing word embeddings. In our case, we need to probe the embeddings of the offers created from texts consisting of many words. Probing tasks, also known as diagnostic classifiers or auxiliary classifiers, involve using the encoded representations of one system to train another classifier on a task of interest. These tasks commonly evaluate if Language Models' representations contain any relevant information.

In our research, we focused on probing tasks that check for non-linguistic properties, such as whether the embedding space of an offer title contains information about the number of sentences in that title or whether it contains information about specific keywords, such as brand names. We wanted to investigate if probing tasks can be used as reliable diagnostic methods for linguistic information encoded in language model representations and other important information.

To accomplish this, we tested a variety of probing tasks, and our results can be found in the section 4. Our research contributes to understanding how language models encode information and how this information can be used to improve natural language processing tasks.

Lindstrom et al. (2020) propose novel probing tasks for the visual-semantic case (pairing images and text), defining three classification tasks relating to the images and text from which the embeddings were created:

- **ObjectCategories** - which of the 80 MS-COCO object categories are present in a given image,
- **NumObjects** - to estimate the number of objects in an image,
- **SemanticCongruence** - whether a caption has been modified (2020).

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

Category	Size	Positive	Negative	Total
Cameras	Medium	1,108	4,147	5,255
Computers	Medium	1,762	6,332	8,094

We took inspiration from this approach and created similar probing tasks corresponding to the product-matching domain.

Moreover, the Facebook AI Research group in (Conneau et al., 2018) has also proposed new probing tasks. Most of them were mainly devoted to understanding linguistic properties, but some posed our task. See Section 5.3 for a more extensive description of the SentEval library and our approach to their proposed tasks.

3 Dataset and EDA

3.1 WDC Dataset

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. In our project, we used the category Computers and Cameras (as an extension to the first project). Due to computational limitations, we focused on the medium size of the datasets. Table 1 presents the exact sizes for used datasets.

3.2 Natural Dataset

In Project #2, we probed on an additional, more 'natural' dataset serving as a reference point when evaluating the results obtained from the WDC dataset - Quora Question Pairs dataset. We sampled approximately 8 000 observations (to provide a similar size to the WDC 'medium' dataset). The dataset contains pairs of questions taken from the Quora website. Similarly, as for the WDC, we have positive pairs (two questions have the same meaning) and negative pairs (two questions regard different matters), but the texts are more 'natural' - do not contain many technical and domain-related words. Instead, they provide subjects, verbs, and objects (natural sentences).

An exemplary negative pair from the dataset is: ("What are good websites for escorts?", "How do I find a good escort?"), and a positive pair: ("How do I use Twitter as a business source?", "How can I use Twitter for business?").

4 Approach and research methodology

4.1 Obtaining embeddings

To begin with, it was necessary to create an embedded space, the properties of which we probe. To do this, we wanted to replicate work from (Możdżonek et al., 2022), which used a cross-encoder architecture with a Bert-like model to distinguish pairs of offers of the same products (positive pairs) from pairs of offers of different products (negative pairs).

The cross-encoder architecture gives better results but requires more data and extended training. In addition, its significant drawback is the lack of explicit use of embeddings for single sentences, which would require additional pooling from embeddings of single tokens.

Therefore, to solve the task of distinguishing between positive and negative pairs of offers, we decided to use a bi-encoder architecture that uses separate XLM-RoBERTa models for each sentence and produces embeddings for each sentence. The embeddings are then compared using cosine distance, which is compared with the target (a positive or negative pair). In this way, the embedded space is constructed to consider the similarity of the offers, and the retrieving of embeddings is straightforward.

4.1.1 Fine-tuning of encoders

We took the pretrained XLM-RoBERTa model ('xlm-roberta-base') from the HugginFace Transformers library. The fine-tuning objective is to predict whether a given pair is positive or negative. First, an embedding is calculated for each sentence from a pair; then, their cosine similarity score is computed and compared with the target

label - 0 (negative) and 1 (positive).

XLM-RoBERTa (Conneau et al., 2019) is a multilingual version of the transformer model RoBERTa. It was pre-trained using 2.5TB of CommonCrawl data, which includes 100 different languages. The model was trained in a self-supervised manner. The training objective was Masked Language Modeling, where the model is given a sentence with 15% of the words randomly masked and must predict the missing words. This allows the model to learn a bidirectional representation of the sentence which is different than traditional recurrent neural networks which process words one after the other. This inner representation can then be used for downstream tasks such as training a classifier using the features produced by the XLM-RoBERTa model as inputs.

We then fine-tuned the model on the WDC and natural dataset. We decided to use the 'medium' size set of two categories from WDC dataset: 'cameras' and 'computers'. As an input sentence representing an offer, we take only its 'title' feature to reduce the computational cost. Fine-tuning continued for 20 epochs, with a batch size of 16 and a cosine similarity function to evaluate embeddings, using the SentenceTransformer library that handled the bi-encoder architecture for us.

4.2 Probing tasks

The bert-based models are so-called black boxes from which it is difficult to deduce why such decisions were made. The probing aims to reveal what information an embedding encodes (Lindström et al. , 2020).

The general outline of probing (well described in (Belinkov , 2021)) is to take a model trained on some task, product matching in our case. Then generate representations using the model and train another classifier that takes the representations and predicts some properties. From the probing classifier's performance, it should be possible to conclude the probed embedding; if the classifier succeeds, it indicates that the semantic embedding captures interpretable information regarding the aspect under consideration. Unfortunately, the converse is invalid: if classifiers perform poorly, the reason may be that the embedding does not capture the property or that the chosen classifier was unsuitable for the task (Lindström et al. , 2020).

In our case, the probing classifier's input are be the offers' embeddings. We examine whether they have learned a relationship with certain properties and check whether they contain information unrelated to the task (high accuracy of probing classifiers).

4.2.1 Proposed probing tasks

The probing tasks we propose are tailored to the dataset we test. In the case of using the new category of the WDC dataset - Computers, the probing tasks are similar to the ones used in the Cameras category. To test embedding space on the new natural dataset, we provide modifications of the previous tasks.

- **Common words**: The goal of the task is to build a classifier that, based on the embedding, will predict whether the listing from which the embedding was calculated contained at least one of the common words ('camera,' 'digital,' 'lens') in case of Cameras dataset or ('computer', 'laptop', 'processor', 'gpu', 'cpu', 'hdd', 'ssd', 'memory') in case of Computers dataset. Such words can be added to any offer sentence without changing their meaning. The probing task is model-agnostic and dataset-agnostic under the assumption that one would choose a different set of common words.

- **Brand name**: The task of classifying embeddings received from offer sentences into two groups - offers that contain the brand name or those without it. We extracted the brand names from the dataset (they often appeared as a separate 'brand_name' feature), and then for some offers, we removed the brand name to obtain a more balanced dataset for probing.

- **Length of sentences**: To investigate whether the embeddings encode information that allows us to distinguish between the original sentence lengths representing the offers, we constructed a classifier that will try to predict sentence lengths based on the offer embedding. The values of sentence length were discretised into five categories to enable classification ([0, 10), [10, 15), [15, 20), [20, 100)). Choosing correctly sized bins is essential to ensure a balanced dataset.

- **Text Similarity**:

  Instead of operating on single offers (embeddings) in this probing task, we return to the

concept of offer pairs from the WDC dataset. For each offer A and offer B from the pair, we calculate one of the three metrics: **Levenshtein distance**, **the Jaccard metric** or **the Jaro–Winkler distance** (Farouk, M, 2019), obtaining a new target variable for probing, which denotes the similarity of the sentences (strings) in the pair. The variable is further discretized into five classes ('Similar,' 'Quite similar,' 'Neutral,' 'Hardly similar,' and 'Not similar'). The embedding of offer A and the embedding of offer B are passed as input to the probing classifier. The goal is to predict the similarity of sentences calculated using the one of the distances - similarity of two strings using Jaccard metric or the smallest number of edit operations (Levenstein: insertion, deletion, substitution, Jaro-Winkler: transpositions) required to transform one string into another.

- **Wh-words**: This task was prepared for the new natural dataset, composed of questions. The goal of the task is to build a classifier that, based on the embedding, will predict whether the listing from which the embedding was calculated contained at least one of the wh-words ('what', 'which', 'who', 'why'). Such words are commonly present in questions, and distinguishing them is important in question understanding.

- **Named Entity**: The task of classifying embeddings received from questions into two groups - sentences that contain the named entity word or those without it. This task regards the natural dataset and is equivalent to *Brand Name* task performed on the WDC dataset. To obtain labels for this probing task (if the named entity is in the sentence) we used language model: **bert-base-NER** (Tjong Kim Sang et al., 2003), which is a fine-tuned BERT model that is ready to use for Named Entity Recognition and achieves state-of-the-art performance for the NER task.

5 Experiments and conclusions

5.1 Model fine-tuning

The first step in the experiment pipeline was to fine-tune the model on every dataset. We used the 'xml-roberta-base' model in the bi-encoder architecture. We fine-tuned the model on the WDC dataset on Cameras, Computer and the new Natural dataset. The result can be seen in Table 2. The hyperparameters of the model can be found in Table 7.

Table 2: Result of fine-tuning.

Dataset	Accuracy
WDC, Cameras, Medium	85%
WDC, Computers, Medium	86.45%
Quora Question Pairs, Natural	79.34%

5.2 Probing tasks

The whole experiment pipeline is divided into two parts based on model selection. We tested every probing task on every dataset on a pre-trained and fine-tuned model to. The testing pipeline was composed of the following steps:

1. Extracting embeddings for pre-trained and fine-tuned models.

2. Preparing a new dataset composed of embeddings and corresponding labels.

3. Splitting data into train and test in ratio 3:1.

4. Training classifiers and testing on the testing dataset.

In all probing tasks, we tested various classifiers: Random Forest, eXtreme Gradient Boosting, and Logistic Regression. Parameters of the classifiers can be found in Table 6. We trained the Logistic Regression model with the Lasso penalty, which is suitable for many features. In our case, there were 768 features for the tasks: Common words, Brand name, Length of sentences, Wh-words, Named entity and 1536 (two offers) for the Text Similarity tasks. Thanks to Lasso regularisation, the classifier selects features that affect prediction, reducing the number of features considered.

The results of the experiments can be seen in Table 3, Table 4, Table 5. As we can see, in most cases, we obtained the best results using a logistic regression model, but XGBoost model also provided good results (e.g. length of sentences on pre-trained model on WDC computers).

Table 3: Accuracy scores[in %] - probing tasks. WDC Cameras dataset.

Classifier	Logistic Regression		Random Forest		XGB	
Model	Pre-trained	Fine-tuned	Pre-trained	Fine-tuned	Pre-trained	Fine-tuned
Common words	93.77	82.75	92.21	77.25	**94.61**	79.28
Brand name	100.00	69.10	100.00	62.66	100.00	66.11
Levenshtein	35.95	33.04	40.71	35.70	41.28	38.37
Jaro-Winkler	35.19	31.20	41.53	35.76	40.14	35.76
Jaccard	33.54	31.58	36.46	36.02	36.40	34.05
Length of sentences	80.72	66.71	76.77	59.52	**82.16**	61.92

Table 4: Accuracy scores[in %] - probing tasks. WDC Computers dataset.

Classifier	Logistic Regression		Random Forest		XGB	
Model	Pre-trained	Fine-tuned	Pre-trained	Fine-tuned	Pre-trained	Fine-tuned
Common words	**93.24**	92.46	**93.24**	92.55	92.81	92.89
Brand name	**96.36**	86.14	91.33	69.24	94.97	73.14
Levenshtein	38.59	33.07	45.47	37.60	45.43	38.59
Jaro-Winkler	35.05	36.74	43.04	36.74	43.37	37.23
Jaccard	38.43	32.33	42.70	35.34	45.26	37.48
Length of sentences	82.41	67.50	79.98	53.37	**85.09**	57.19

5.3 SentEval Library

In project #2, we explored the capabilities of a library dedicated to probing tasks - SentEval by Meta (Conneau et al., 2018). However, this library does not allow the automatic creation of probing tasks based on the submitted data but rather automates the construction of a probing classifier based on input files that constitute the dataset for a given probing task and a transformer that calculates embeddings. We need to prepare corresponding input files on our own. Also, SentEval focuses more on grammatical aspects of sentences (like tenses, parts of speech, etc.). Still, such probing tasks are not suited for the WDC dataset, where an offer is usually a sequence of different parameter names. In the notebook 'project2_notebooks/SentEval_probings.ipynb', we provided a sample solution for sentence-length probing tasks using SentEval.

5.4 Results interpretation and discussion

Learning from Project #1, we conducted additional tests to ensure a better understanding. We performed all probing tasks before and after fine-tuning a transformer on our dataset to capture the impact of this process on the ability of embeddings to represent a probed property. Additionally, given that the WDC dataset contains many technical words, we tested probing on a "natural" dataset posing a reference point. Tables 3, 4, and 5 show exact results for each probing task.

5.4.1 Common words/wh-words

This probing task aimed to classify whether a sentence contains at least one of the predefined 'common words'. For the 'natural' dataset, as the 'common words,' we chose 'why-words ('who,' 'where,' 'when,' etc.).

The results are quite interesting because, for all datasets, probing models performed worse for embeddings from the fine-tuned model than for embeddings calculated before fine-tuning. These results may indicate that fine-tuning causes the transformer to focus less on common words, and such behavior is expected since common words do not provide any discriminative value and can occur in all offers (e.g., a word 'digital' can be applied for all cameras from the WDC dataset).

Nevertheless, our probing models were still quite successful in detecting common word occurrences from embeddings, which proves that the embedded space considers this information. The best F1 scores for the 'natural' dataset were 0.70 (before fine-tuning) and 0.66 (after fine-tuning) obtained with the Random Forest classifier; for the WDC 'computers' dataset were 0.55 (before fine-tuning) and 0.52 (after fine-tuning) obtained with the XGB classifier; for the WDC 'cameras' dataset were 0.95 (before fine-tuning) and 0.79 (after fine-tuning) obtained with the XBG classifier.

5.4.2 Brand names/named entities

This probing task aimed to differentiate between embeddings calculated from an offer with and

Table 5: Accuracy scores[in %] - probing tasks. Natural dataset.

Classifier	Logistic Regression		Random Forest		XGB	
Model	Pre-trained	Fine-tuned	Pre-trained	Fine-tuned	Pre-trained	Fine-tuned
Wh-words	**70.74**	66.50	68.18	63.25	68.49	62.68
Named Entity	**81.44**	75.52	78.04	67.38	79.79	68.95
Levenshtein	28.42	28.42	27.37	26.32	33.68	28.42
Jaro-Winkler	32.63	23.16	24.21	23.16	33.68	21.05
Jaccard	29.47	18.95	22.11	28.42	20.00	22.11
Length of sentences	**62.18**	60.54	61.19	59.43	61.88	59.97

Table 6: Probing classifiers.

Classifier	Parameters
RandomForest	n_estimators = 100
	criterion = gini
	min_samples_split = 2
GradientBoosting	loss = log_loss
	learning_rate = 0.1
	n_estimators = 100
LogisticRegression	multi_class =multinomial
	penalty = l1
	solver = saga

without a brand name. In the case of the 'natural' dataset, instead of brand names, we considered named entities.

In the case of the WDC dataset (both 'cameras' and 'computers'), we noticed that the results of this probing task decreased after fine-tuning. Contrary, for the 'natural' dataset, we can observe that the embeddings from a fine-tuned model allow probing classifiers to achieve better results. A possible explanation is that in the case of the 'natural' dataset, named entities usually pose a very important role in a question. Still, in the case of the WDC dataset, we can have many offers from the same producer (brand), and this information may not be the most important for product matching.

The best F1-scores for the WDC 'computers' dataset were 0.96 (before fine-tuning) and 0.86 (after fine-tuning); for the WDC 'cameras' dataset, were 1.0 (before fine-tuning), and 0.69 (after fine-tuning); for the 'natural' dataset were 0.75 (before fine-tuning) and 0.81 (after fine-tuning). The best results were obtained from the Random Forest classifier in all cases.

5.4.3 Text similarity

Text similarity measures cannot be applied successfully to the product/question matching problem since they do not provide robustness against using synonyms or changing the order in the sentence. For this reason, we nowadays use transformer-based models to solve such tasks. When designing this task, we wanted to check whether such "naive" metrics (and properties they measure) were considered by the transformer.

For all datasets, WDC and 'natural', the results for this probing task are poor before and after fine-tuning. Once again, from poor results, we cannot reason that our embeddings do not have such information encoded. But we didn't confirm it either, which is expected due to the nature of text similarity metrics. We tested different metrics, and for none of them, the probing was successful.

The best accuracies for this probing task were: for the 'computers' WDC dataset - 45% (before fine-tuning) and 38.6% (after fine-tuning) obtained from the XGB classifier; 41% (before fine-tuning) and 35% (after fine-tuning) obtained from the Random Forest classifier; and for the 'natural' dataset - 36.7% (before fine-tuning) and 21% (after fine-tuning) obtained from the XGB classifier.

5.4.4 Length of sentences

In this task, the probed property was the length of an input sentence from which the embedding was calculated. Usually, this property should not be important. Still, in the case of the WDC dataset, the offers are usually a sequence of different parameter values, so their length may not necessarily be irrelevant. Similarly to the natural dataset - long questions are often rephrased using a similar number of words, and vice-versa.

In the case of the WDC dataset (both 'computers' and 'cameras'), as expected, the probing classifiers managed to distinguish embeddings created from sentences with lengths from different bins ($[0, 10)$, $[10, 15)$, $[15, 20)$, $[20, 100]$). However,

the fine-tuning process decreased the probing results. This may mean that eventually, paying too much attention to the length is not the best idea. Nevertheless, it is somehow relevant (maybe with a combination of other features, so the fine-tuned embeddings still allow for length differentiation, but the results are worse than before fine-tuning).

This probing task yielded poor results before and after fine-tuning for the 'natural' dataset. Noteworthy, when a probing classifier does not work well, we cannot reason that the probed property is neglected in the embedded space. The reason may be that our probing classifier cannot extract this information either.

The best F1-Scores for the WDC 'computers' were 0.83 (before fine-tuning) and 0.55 (after fine-tuning) obtained from the XGB classifier; for the WDC 'cameras' were 0.78 (before fine-tuning) and 0.52 (after fine-tuning); for the 'natural' dataset 0.37 (before fine-tuning) and 0.28 (after fine-tuning).

6 Conclusions

Our projects have proven the validity of constructing probing tasks to explore the properties of embedded spaces. Probing tasks do not guarantee ground truth, but they do allow us to shed light on some of the features of input spaces that are considered by black-box transformers. A proper comparison of probing task results allows many interesting conclusions to be drawn and can enable the model to be improved, knowing its good and weak points. We also shed light on the process of fine-tuning, indicating how it affects the model's consideration of the discussed properties.

7 Future works

Different types of probing tasks can be examined. There are many possibilities for properties to be probed, e.g., different common words or occurrences of a product price, etc.

Another interesting research can regard comparing different types of transformers and also performing probing tasks in different stages of fine-tuning (before, after some epochs, when the best accuracy is obtained, and when over-fitting occurs) to see how different properties influence the model.

Table 7: Transformer-model parameters.

Name	xlm-roberta-base
Fine-tuned on	WDC dataset
Categories	'Cameras', 'Computers'
Size	'medium'
train_batch_size	16
num_epochs	80
warm_steps	len(training_dataset) $*$ 10
train_loss	cosine similarity
weight_decay	0.01

8 Team contribution

Name	Paweł Golik
Work	Research, Models fine-tuning, Preparing embeddings, Natural Dataset preparation and preprocessing, SentEval probing tasks preparation, Contributing to writing report, Contribution to preparing presentation
Time [h]	34

Name	Mateusz Jastrzebiowski
Work	Research, Text similarity: Levenstein, Jaro-Winkler, Jaccard, Sentence len probing task preparation, Dataset modification for probing tasks, Testing probing tasks, Contributing to writing report, Contribution to preparing presentation
Time [h]	35

Name	Aleksandra Muszkowska
Work	Research, Common words, Brand name, Wh-words, Named entity probing task preparation, Dataset modification for probing tasks, Testing probing tasks, Contributing to writing report, Contribution to preparing presentation
Time [h]	35

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*

[Farouk, M2019] Farouk, M. 2019. *Measuring sentences similarity: a survey. arXiv preprint arXiv:1910.03940*

[Wolf et al.2019] Wolf, Thomas & Sanh, Victor & Chaumond, Julien & Delangue, Clement. 2019. *Transfertransfo: A transfer learning approach for neural network-based conversational agents.*

[Mazare et al.2018] Mazare, Pierre-Emmanuel & Humeau, Samuel & Raison, Martin & Bordes, Antoine. 2018. *Training millions of personalized dialogue agents.*

[Conneau et al.2018] Conneau et al., ACL 2018 *What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties*

[Tjong Kim Sang et al.2003] Tjong Kim Sang et al. 2003 *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*

[Conneau et al.2019] Conneau, Alexis and Khandelwal, Kartikay and Goyal, Naman and Chaudhary, Vishrav and Wenzek, Guillaume and Guzmán, Francisco and Grave, Edouard and Ott, Myle and Zettlemoyer, Luke and Stoyanov, Veselin 2019 *Unsupervised Cross-lingual Representation Learning at Scale*

[Conneau et al.2018] Conneau, Alexis and Kiela, Douwe 2018

SentEval: An Evaluation Toolkit for Universal Sentence Representations