Matching Products with Deep NLP Models

L. Akritidis[1], P. Bozanis[1]

[1]School of Science and Technology, International Hellenic University

14th International Conference on Information, Intelligence, Systems and Applications (IISA 2023)

University of Thessaly, Volos, Greece

10 – 12 July, 2023

Product matching

- Nowadays, the eCommerce industry is evolving towards enterprises and services that collect product-oriented data from multiple external sources which are:
  - uncontrolled
  - independent of each other,
  - providing information in a diverse manner
- Examples of modern dynamic eCommerce services:
  - Online auction houses (eBay, etc.)
  - Product comparison platforms (Pricerunner, Google products, etc.)
  - Typical electronic stores.
- The identification of products a difficult task.

Incoming offers

Offers

- o_1: ("Buy iPhone 13 no shipping", "cellphones", Ø)
- o_2: ("Apple iPhone 13 Pro 5G", "mobile", Ø)
- o_3: ("Intel CoreI7 12700K", "CPU", "3.2GHz; 12 core")
- o_4: ("Apple iPhone 13 128GB Pink", "mobile", Ø)
- o_5: ("Intel CoreI7 12700", "CPU", "3.2GHz; 12 core")
- o_6: ("Intel CoreI7 12700K", "Processors" , Ø)
- o_7: ("CoreI7 12900K", "Processors" , Ø)

Challenges (text-oriented)

- Diversity: Data coming from multiple external uncontrolled sources cannot be fully trusted.
  - Incorrect/inconsistent/noisy technical specs, descriptions, even images.
- Working with product titles only is a common approach. However, product titles are:
  - **Sparse**: considered as a form of short text, they exhibit high sparseness that, in turn, blurs the semantic similarity with other titles.
  - **High dimensional**: the traditional tf-idf text representations are highly dimensional especially in cases of large-scale data.
  - **Noisy**: Text cleaning is requires due to typos, amphisemy, and polysemy:
    - PS = PlayStation = Plystation

Challenges (product-oriented)

- Latent similarity:
  - Highly similar text sequences do not necessarily represent identical product entities and vice versa.
- Data enrichment
- Large data volumes.
  - Typical medium-sized stores include hundreds of thousands of products.
  - Product comparison platforms and auction houses: 1-3 orders of magnitude larger.
- High data velocity.
  - The product-related information changes rapidly.

Existing approach (1)

- UPM clustering algorithm with post-processing verification.
- L. Akritidis, A. Fevgas, P. Bozanis, C. Makris, "A Self-Verifying Clustering Approach to Unsupervised Matching of Product Titles", *Artificial Intelligence Review*, 53 (7): 4777-4820, 2020.
- L. Akritidis, M. Alamaniotis, A. Fevgas, P. Bozanis, "Confronting Sparseness and High Dimensionality in Short Text Clustering via Feature Vector Projections", In *Proceedings of the 32nd IEEE International Conference on Tools with Artificial Intelligence*, pp. 813-820, 2020.

Existing approach (2)

- General Idea: Represent titles with dense (indexed) vectors.
  - Makes the title low dimensional but renders it inappropriate for most machine learning libraries.
- Create latent variables by concatenating/combining existing features.
- The weight of each latent variable in the (dense) vector is determined by:
  - Its length (number of component variables),
  - Its frequency,
  - A summation term that accumulates attribute scores of the component features:
    - Their position in the title,
    - Their nature (technical spec, measurement unit, brand name, noise, etc.)
- Each latent variable in the latent space is a representative of the title.
  - Makes the title low dimensional.
- Products sharing common heavy latent-variables are clustered together.

Existing base (example)

- Two titles: Intel CoreI7 7700K 3.6GHz and CoreI7 3.6GHz 7700K,
- There are two common latent variables of length 2:
    - CoreI7 7700K and CoreI7 3.6GHz.
- And one common latent variables of length 3:
    - CoreI7 7700K 3.6GHz
- These two titles will be considered that they match because they share a common latent variable.

Criticism

- The operating environment of dynamic eCommerce systems is different than the one assumed by our previous work.
- An unsupervised, cold-start algorithm that does not use, or require a training set.
- Despite its good performance, it ignores:
  - the existing base of products.
  - the fast data updates (additions, modifications, deletions).
- Instead, it is capable of learning patterns from the underlying data only.
- Moreover, despite its superiority in terms of execution speed over all the other competitive algorithms, UPM was not designed to operate on large-scale data.

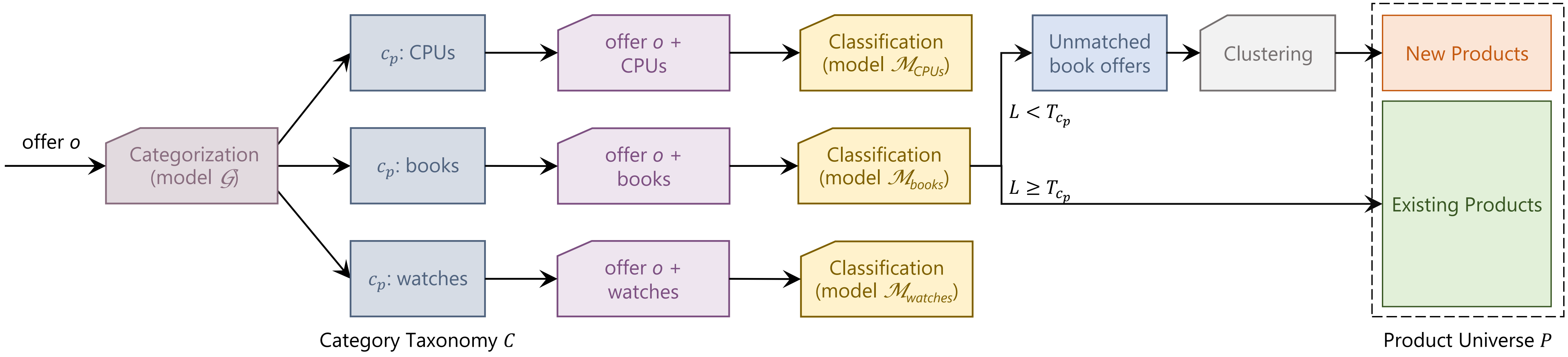
Objectives

- Design, development, and implementation of machine learning algorithms for effectively processing large-scale e-Commerce data coming from multiple diverse sources.
- Take into consideration:
  - the existing base of products and
  - the fast data updates.
- Effectively address all the aforementioned issues:
  - Sparseness, high dimensionality, noisy samples, latent similarity, ...

Functionalities

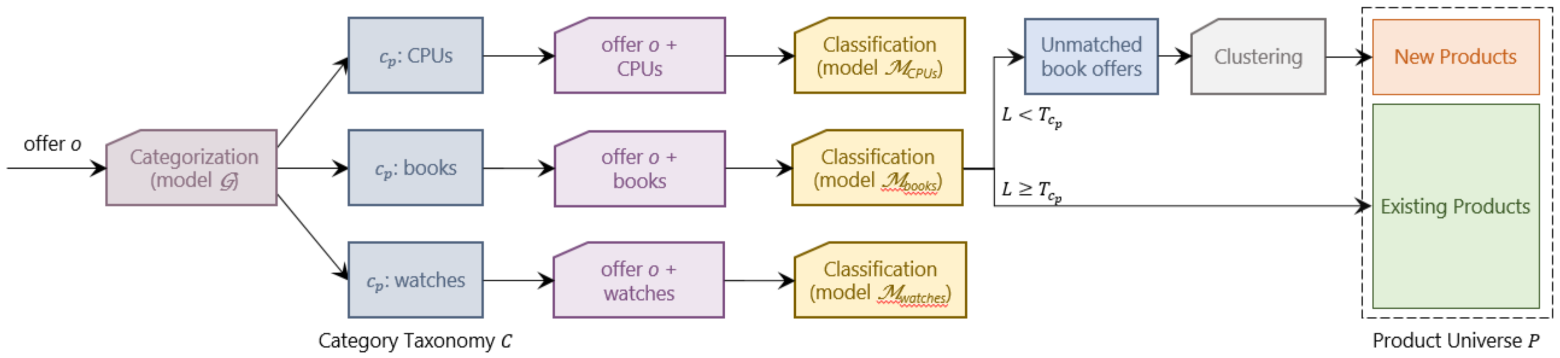
- Given an existing database P and a set of incoming product offers O, the list of functionalities includes:
- Analysis of the offers: for each offer $o \in O$, identify the product $p \in P$ that it represents. Create a match between o and p.
- In several cases, o may refer to a product that is not present in P. Create a new record p', insert it to P, and create a match between o and p'.
- Categorization is extremely important:
  - it facilitates category-based navigation.
  - It has been proved to enhance both matching quality and execution speed.
- Optional: Several applications require that the products of P that match no incoming offers to be deleted, deactivated, or marked as unavailable.

Proposed architecture (preliminary)



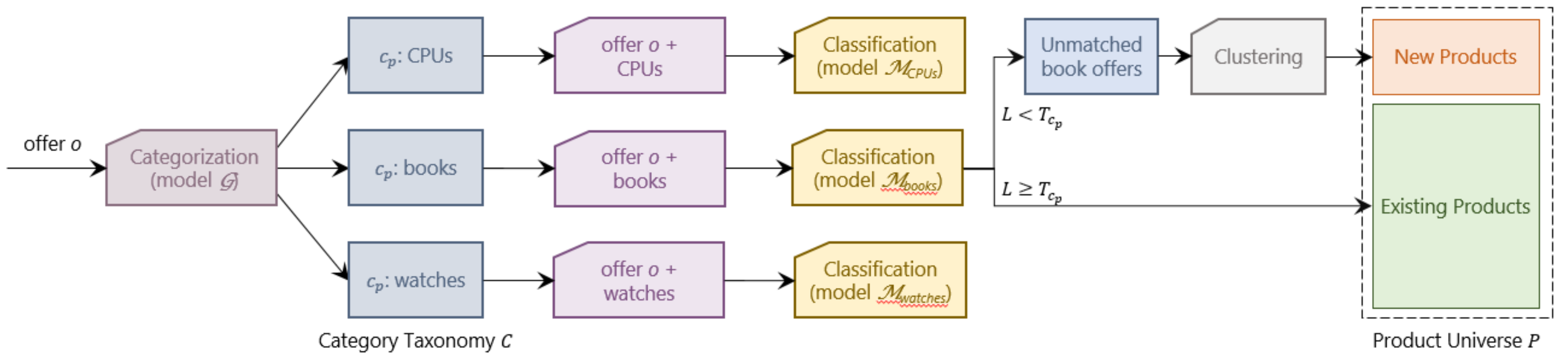
Offer categorization

- Given a pre-existing product taxonomy C, a deep learning model $\mathcal{G}$ assigns a category $c \in C$ to o.
  - $\mathcal{G}$: A BiLSTM network with 768 units, Dropout (0.2) and ReLU activation.
- Preprocessing filters: case folding, punctuation removal, noise removal.
  - dots and dashes, are significant for recognizing the identify of a product.
- Short-text representation:
  - Word embeddings (Word2Vec, GloVe, BERT, etc.) are not suitable (they encode words).
  - Averaging the word embeddings to derive short text embedding does not work well.
  - **Sentence embeddings are required (e.g. SentenceBERT).**



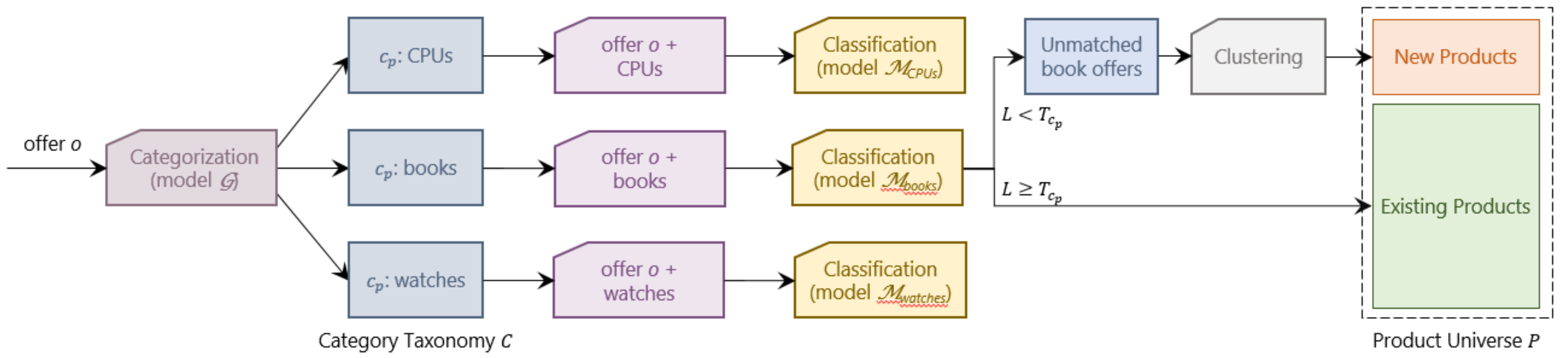
Offer classification (1)

- This stage matches o with a single product $p \in P$. It uses a set of models $\mathcal{M}$ that includes $|C|$ classifiers, one per category.
- The classification is performed by a picking a model $\mathcal{M}_c \in \mathcal{M}$ that has been trained with the products belonging to the category c of o.
  - $\mathcal{M}_c$: any model, but we employed a BiLSTM similar to $\mathcal{G}$.
- A softmax function allows the interpretation of the output as a probability L that is indicative of the classification trustworthiness.



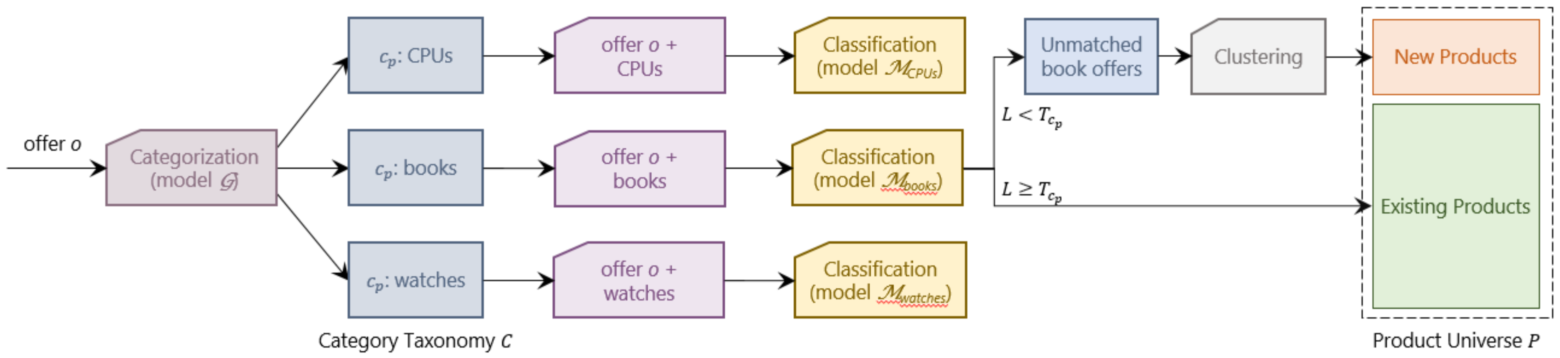
Offer classification (2)

- This strategy avoids querying the entire database P.
- It limits the possibility of false matches: it searches only among products of the same category as the offer o.
- Matching is much faster because it is deployed on a subset of the original data.



Clustering of the unmatched offers

- In case L is smaller than a category-specific threshold T_c, we assume that o matches none of the products $p \in P$.
- Create $|C|$ pools of unmatched offers, one per category $c \in C$.
- Apply a clustering algorithm to create new clusters and place the similar offers there. The cluster labels are subsequently utilized to create new products and append them to P.



Conclusions & Future Work

- In this Work-In-Progress paper we introduced a deep learning approach to the problem of product matching in e-Commerce systems.
- The proposed method can effectively match an incoming offer to a product entity, whereas it is also capable of handling offers of new products that do not match any of the existing entries.
- The category-based approach is designed to improve both matching quality and efficiency.
- Our current work is mainly oriented towards the proper selection of the categorization and classification models and the design of their architecture.
- Additional research is also conducted towards the identification of category-based aspects that will further improve the effectiveness of our method.

Thank you for watching

I would be happy to answer your questions.

Please send them to lakritidis@ihu.gr

Co-financed by Greece and the European Union (European Social Fund-SF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning 2014-2020" in the context of the project "Support for International Actions of the International Hellenic University" (MIS 5154651)