Team Project HWS 2020

Cross-lingual Product Matching using Transformers

Hallo

- **Prof. Dr. Christian Bizer**
- Professor for Information Systems V
- Research Interests:
  - Web Data Integration
  - Data and Web Mining
  - Adoption of Data Web Technologies
  - Knowledge Base Construction
- Room: B6 - B1.15
- eMail: chris@informatik.uni-mannheim.de
- Consultation: Wednesday, 13:30-14:30

Hallo

- **Ralph Peeters**
- Graduate Research Associate
- Research Interests:
  - Product Data Integration
  - Entity Matching using Deep Learning
- Room: B6, 26, C 1.04
- eMail: ralph@informatik.uni-mannheim.de

Agenda of Today‘s Kickoff Meeting

1. You and Your Experience
2. Motivation and Project Goals
3. The WDC Product Corpus for Large-Scale Product Matching
4. Organization
5. Specific Subtasks
6. Schedule
7. Formal Requirements

You and Your Experience

- A Short Round of Introductions
  - What are you studying? Which semester?
  - Which DWS courses did you already attend?
  - What are your programming and data wrangling skills?
  - Did you already work on any data integration or cleansing projects?
- Participants

1. Küpfer, Andreas
2. Ebing, Benedikt
3. Schweimer, Daniel
4. Niesel, Fritz
5. Gutmann, Jakob

Motivation of the Team Project

The Web is a rich source of product information

- same product is described by 100s of websites
  - merchants, producer, consumers
  - different websites describe different aspects of a product
  - technical spec vs consumer experience
- there are plenty of offers for a product online
  - we can collect information on global scale
  - many websites point us at similar products

Using information about products from the Web, we can

- build comprehensive product catalogues and search engines
- construct global price comparison engines
- understand consumer and market behavior

Product Matching: An essential problem to solve

Matching products across websites is hard

- structural heterogeneity (differences in schemata)
- semantic heterogeneity (differences in meaning)
  - synonyms, homonyms
  - conflicting data
  - also: **different languages**

Features that help us distinguish products

- identifiers (GTINs, MPNs, ISBNs, …)
- titles (product name plus selected features
- descriptions (long free texts)
- specification tables and lists (detailed features as K/V pairs)
- pictures

UPC 610214632623

Samsung Galaxy S4 Verizon AT&T T-Mobile GSM
Unlocked Smartphone SRF

Produkttyp	Smartphone
Formfaktor	Touch
Integrierte Komponenten	Rückwärtige Kamera, Frontkamera,
Breite	70 mm
Tiefe	7.9 mm
Höhe	137 mm
Gewicht	130 g
Gehäusefarbe	White Frost

Das Samsung Galaxy S4 ist der unterhaltsame und hilfreiche Begleiter für Ihr mobiles Leben. Es verbindet Sie mit Ihren Liebsten. Es lässt Sie gemeinsam unvergessliche Momente erleben und festhalten. Es vereinfacht Ihren Alltag.

Difficulty of the Task depends on the Product Category

– Books
  - wide adoption of identification schema (ISBNs)
  - Entity resolution problem mostly solved ☺
  - other features like title and author often only used for sanity checks

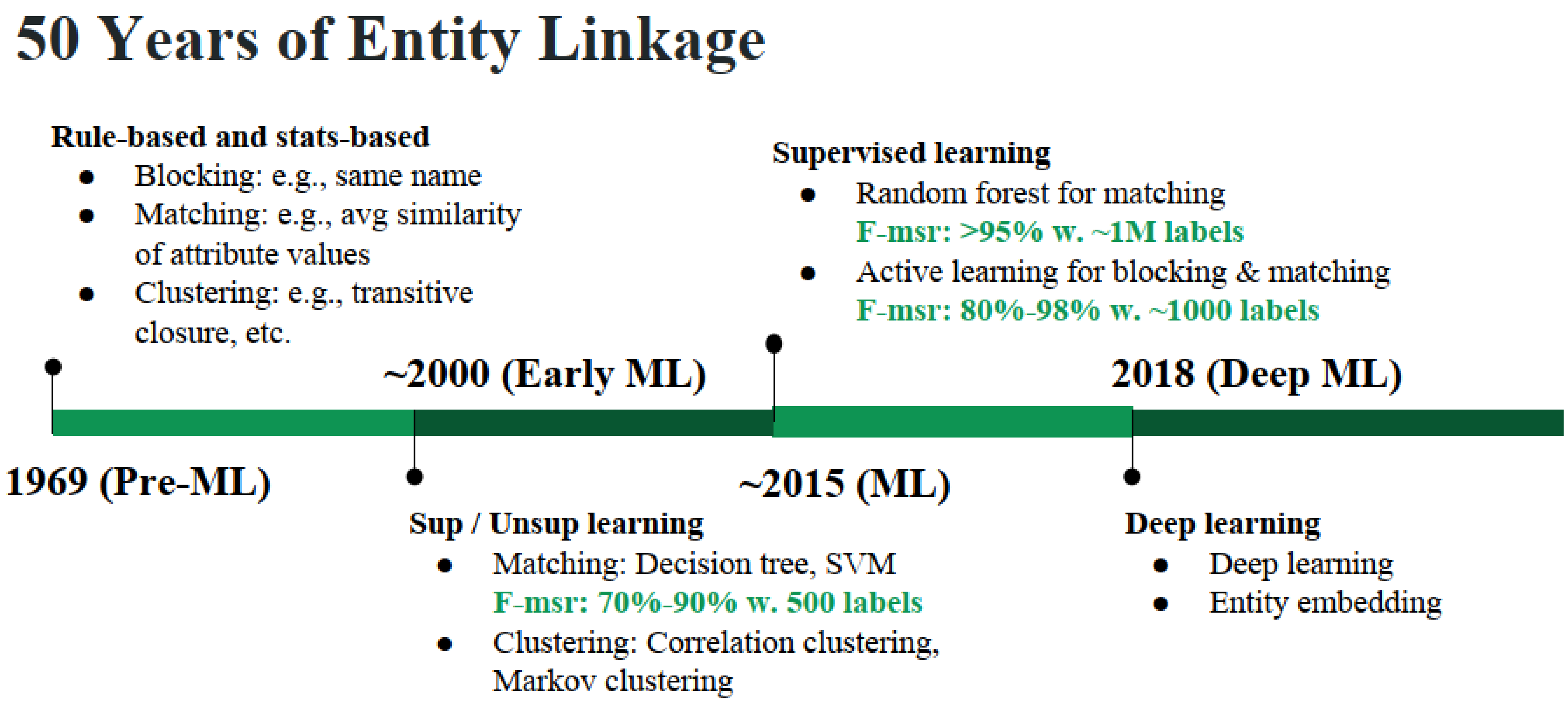
– Phones / Computers / Cameras
  - rather structured descriptions, often including tables / lists
  - different sites often describe same features

  ➔ entity resolution methods for structured data can be applied

– Cloths / Bags / ….
  - rather unstructured descriptions, not too many tables / lists
  - only weak agreement of attributes

  ➔ entity resolution / disambiguation methods for texts need to be applied

Entity Linkage over Time



Source: https://thodrek.github.io/di-ml/sigmod2018/sigmod2018.html

Transformers in NLP

Transformer architectures like BERT have large impact on NLP

- stacked encoder layers with self-attention mechanism
- every token can attend to every other token in both directions
- contextual representations, i.e. embedding depends on context
- multiple attention heads allow learning of different concepts

Pre-training / Fine-tuning paradigm

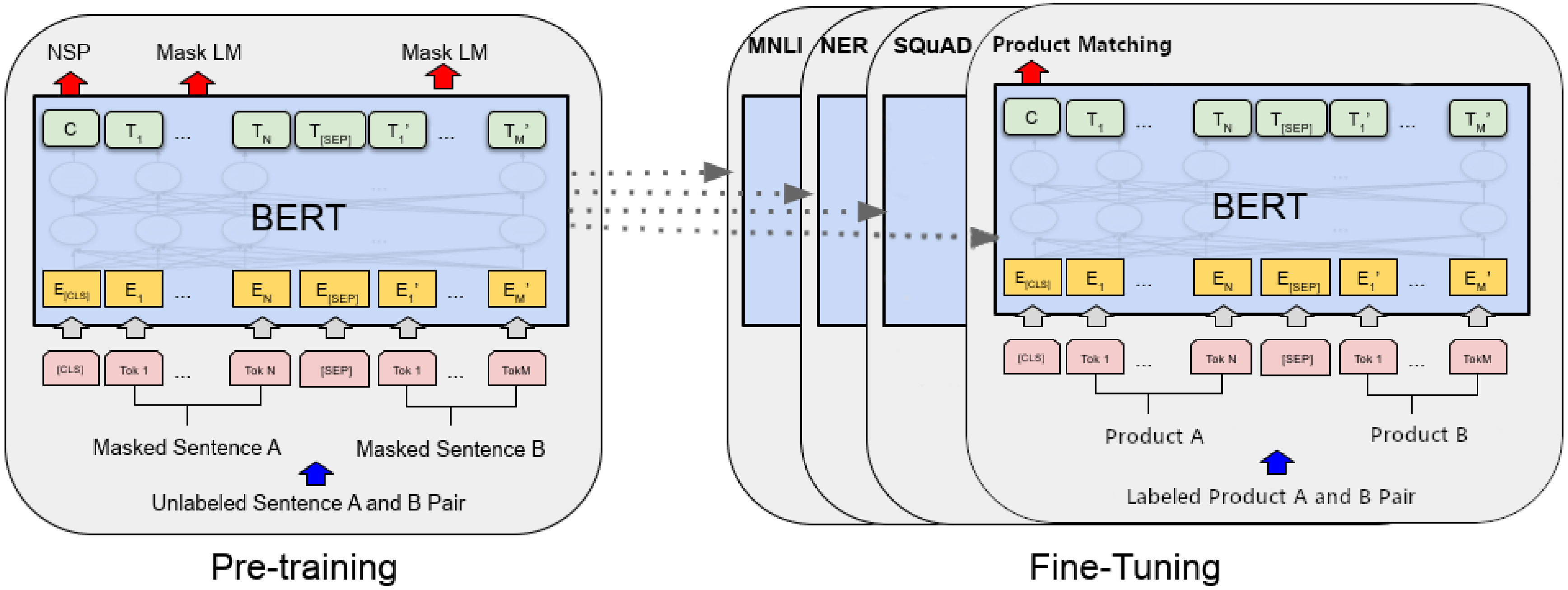
- pre-training: self-supervised general (masked) language modeling
- fine-tuning: Further training on task-specific data

→ shown to work extremely well for a variety of problems

→ including product matching ☺

→ All submissions of the ISWC2020 product matching challenge are based on BERT (https://ir-ischool-uos.github.io/mwpd/, https://ir-ischool-uos.github.io/mwpd/MWPD20/paper1.pdf)

Pre-training / Fine-tuning paradigm example: BERT



- Pre-training: Masked language modeling (MLM) and next sentence prediction (NSP) objectives on large natural language corpus
- Fine-tuning: Adaptation of final layer(s) to task followed by continued training with task-specific data

Training of select Multi-lingual Transformers

- **Multilingual BERT** (https://github.com/google-research/bert/blob/master/multilingual.md)
  - Pre-trained on 100 languages with the largest Wikipedias
  - Varying sizes handled by smoothed weighting

  → under-/oversampling of high-resource/low-resource languages
  - Shared WordPiece vocabulary of 110K tokens
- **XLM** (https://arxiv.org/abs/1901.07291)
  - Pre-trained on 15 language Wikipedias (those contained in XNLI)
  - Various other *parallel* corpora (e.g. MultiUN, EUbookshop, ...)

  → used for additional TLM objective, similar to MLM but with parallel sentences
  - Shared BPE vocabulary of 95K tokens
- **XLM-RoBERTa** (https://arxiv.org/abs/1911.02116)
  - Extension of XLM trained on parts of CommonCrawl for 100 languages
  - Vocabulary of 250K tokens

Project Goal

Can matching knowledge be transferred between languages?

- Can we augment performance on languages with small amounts of training data by adding data from more readily available languages (English)?
  - E.g. Fine-tune model for the task using english data
  - Futher fine-tune using few examples of target language (few-shot learning)
- Can we achieve good performance without **any** data from the target language? (Zero-shot performance)
- If it works well, how can we explain it?

Project Goal

1. Collect product data from a large number of websites in different languages and build train / test sets for experimental evaluation

2. Match multi-lingual product data using multiple multi-lingual Transformer-based models

3. Compare performance of methods w.r.t.:
   - Product categories (structured vs. semi-structured input)
   - Simple Baselines (e.g. Random Forest / SVM)
   - Mono-lingual Transformers
   - Product popularity (amount of training data)

4. Explain why models perform better than others
   - Conduct an error analysis
   - Apply explainability methods

Learning Targets

Improve your technical skills

- Work as a **Data Scientist**: clean, profile, integrate, classify data
- Understand the nature of **Web Data**
- Improve your technical expertise / programming skills

Improve your soft skills

- Work as part of a bigger team on a more complex project
- Organize yourself and assign tasks based on your skills
- Communicate and coordinate your work

How to find many offers of the same product?

Not an easy task!

1. Which sources to consider?
2. Which data to extract?
3. How to recognize identical products?
4. How to categorize products?

OR...

Use the **WDC Product Corpus for Large-Scale Product Matching**
http://webdatacommons.org/largescaleproductcorpus/v2/

The WDC Product Corpus for Large-Scale Product Matching

1. Semantic Annotations in HTML Pages
2. Web Data Commons Project
3. Web Data Commons – Product Corpus for Large-Scale Product Matching

Semantic Annotation of HTML Pages: Schema.org


schema.org Search

Home Schemas Documentation

Thing > Organization > LocalBusiness

A particular physical business or branch of an organization. Examples of LocalBusiness include a restaurant, a particular branch of a restaurant chain, a branch of a bank, a medical practice, a club, a bowling alley, etc.

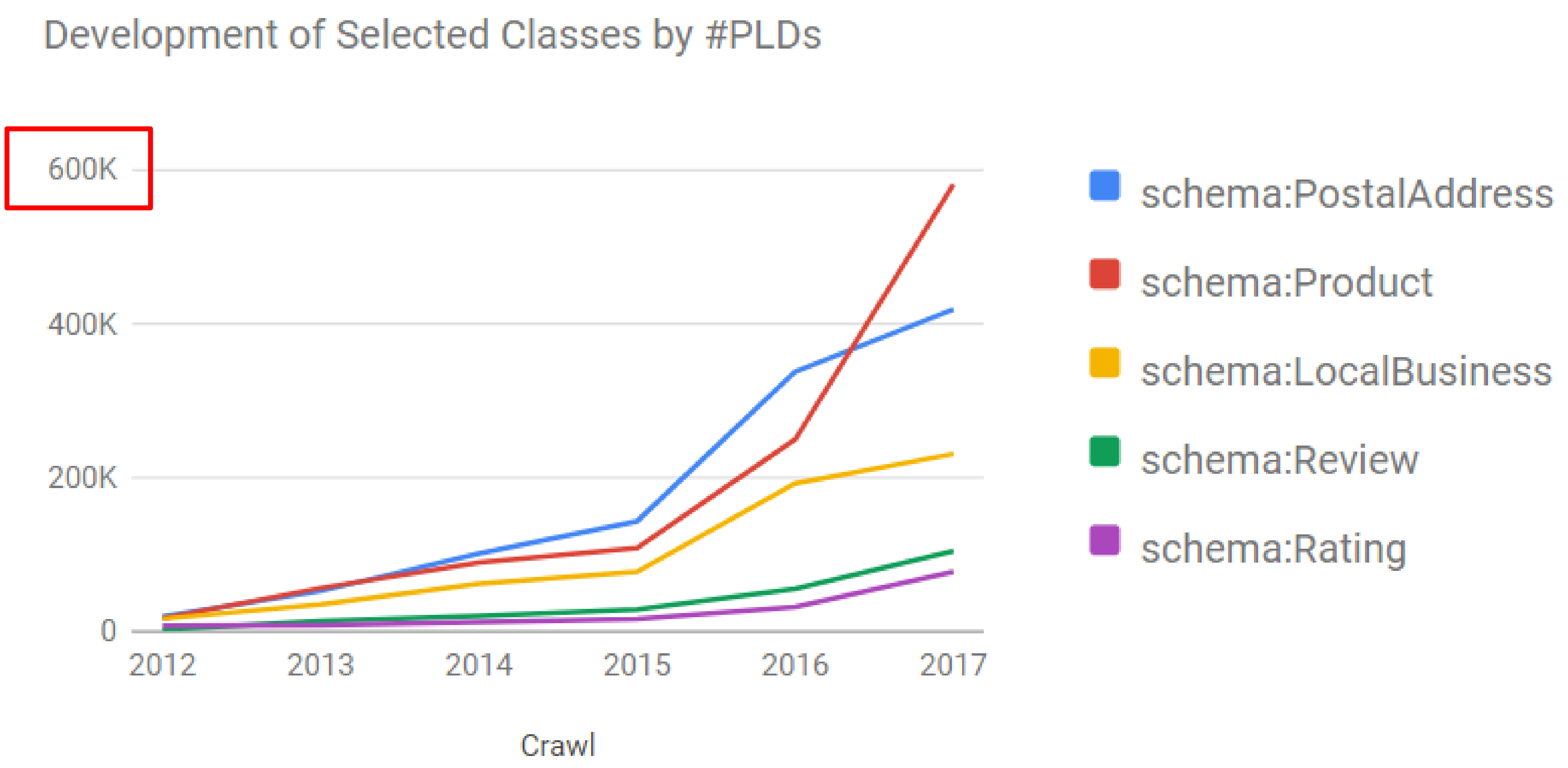
Property	Expected Type	Description
Properties from Thing		
description	Text	A short description of the item.
image	URL	URL of an image of the item.
name	Text	The name of the item.
url	URL	URL of the item.
Properties from Place		
address	PostalAddress	Physical address of the item.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
containedIn	Place	The basic containment relation between places.

- ask site owners since 2011 to annotate data for enriching search results
- 675 Types: Event, Place, Local Business, Product, Review, Person
- Encoding: Microdata, RDFa, JSON-LD

Example: Microdata Annotations in HTML

```
<div  itemtype="http://schema.org/Product">
  <span itemprop="name">Sony GTK-XB5L Audiosystem</span>
  <span itemprop=“gtin13“>04048945021687</span>
  <span itemprop=“description">high-power home audio system with Bluetooth technology</span>
</div>
 <div itemprop="aggregateRating" itemscope itemtype="http://schema.org/AggregateRating">
    <span itemprop="ratingValue"> 4 </span> stars-based on
    <span itemprop="reviewCount"> 250 </span> reviews.
 </div>
```

schema.org Annotations: Most Popular Classes



http://webdatacommons.org/structureddata/

Properties used to Describe Products 2017

Top 15 Properties	PLDs	
	#	%
schema:Product/name	535,625	92%
schema:Offer/price	462,444	80%
schema:Product/offers	462,233	79%
schema:Offer/priceCurrency	430,556	74%
schema:Product/image	419,391	72%
schema:Product/description	377,639	65%
schema:Offer/availability	337,876	58%
schema:Product/url	263,720	45%
schema:AggregateRating/ratingValue	184,004	32%
schema:Product/sku	126,696	22%
schema:AggregateRating/reviewCount	112,408	19%
schema:Product/aggregateRating	101,434	17%
schema:Product/brand	73,934	13%
schema:Product/productID	35,211	6%
schema:Product/manufacturer	21,967	4%

Samsung Galaxy S4 Verizon AT&T T-Mobile GSM Unlocked Smartphone SRF

Das Samsung Galaxy S4 ist der unterhaltsame und hilfreiche Begleiter für Ihr mobiles Leben. Es verbindet Sie mit Ihren Liebsten. Es lässt Sie gemeinsam unvergessliche Momente erleben und festhalten. Es vereinfacht Ihren Alltag.

UPC 610214632623

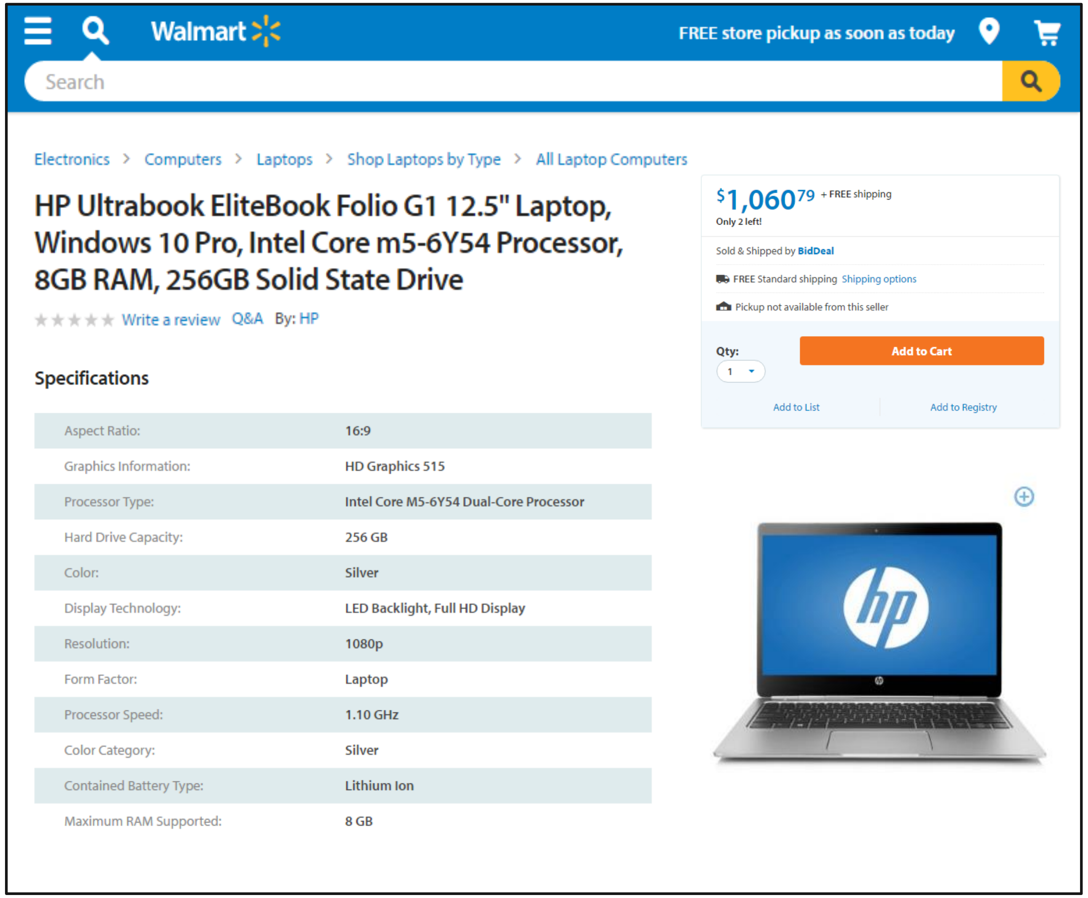
UPC 610214632623

Attribute/Value Table together with schema.org Annotations

s:breadcrumb

s:name

Attribute/Value HTML Table



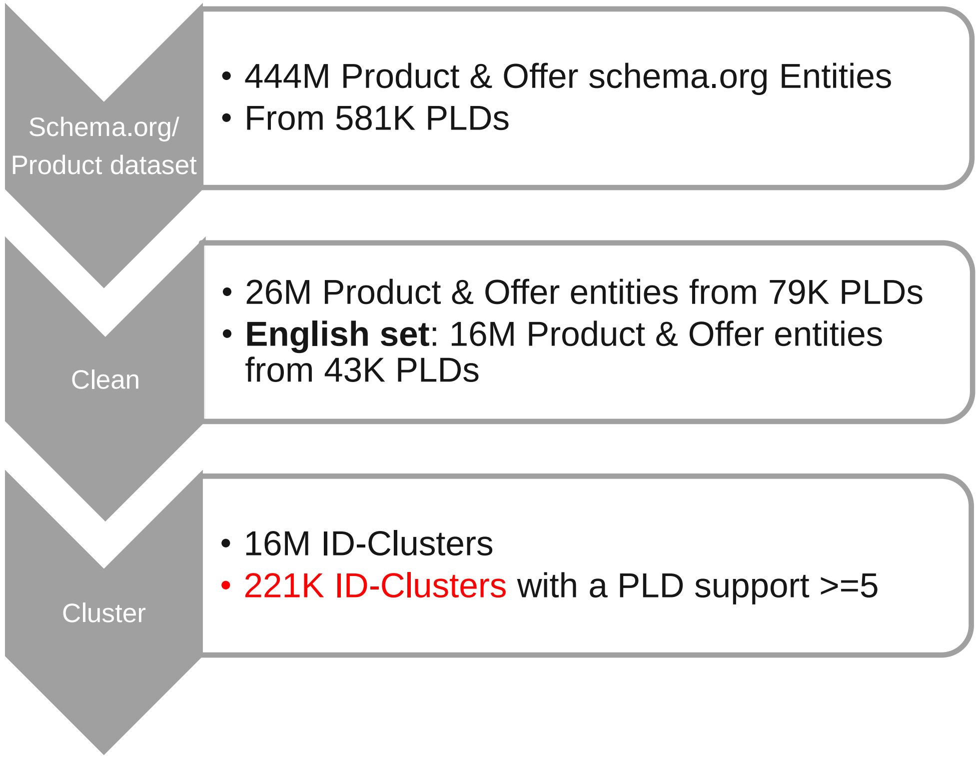
Qui, et al.: **DEXTER: Large-Scale Discovery and Extraction of Product Specifications on the Web.** VLDB 2015.
Petrovski, P., Bizer, C.: **Extracting attribute-value pairs from product specifications on the web.** In: International Conference on Web Intelligence, 2017.

The WDC Product Corpus for Large-Scale Product Matching

– Product corpus grouping schema.org product/offer annotations by identifier value.
  - all WDC 2017 product data is included that
  - provides some sort of product ID (gtin, mpn, sku, identifier)

– Initial cleaning steps are performed

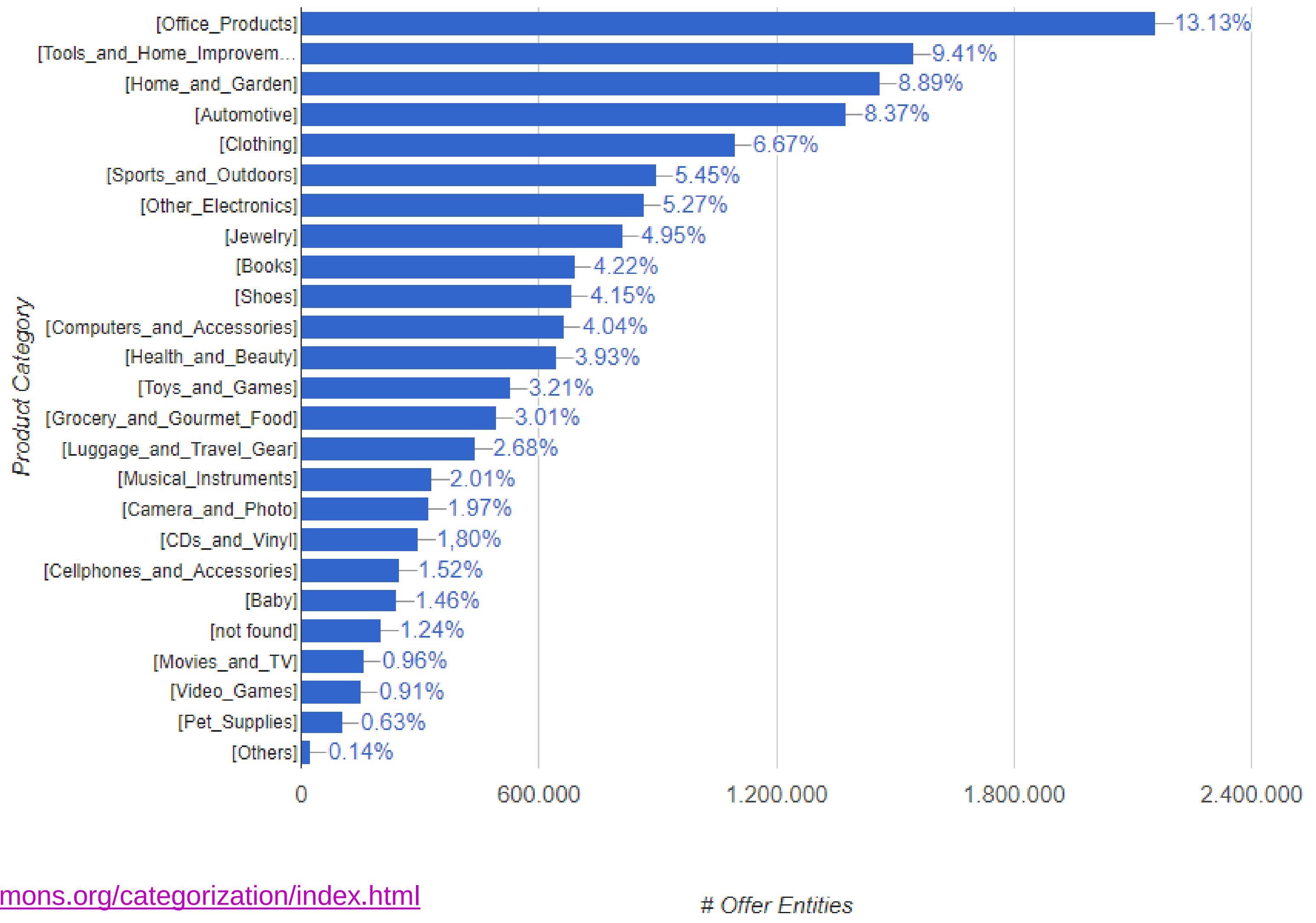
– Clustering of product descriptions from different PLDs (web sites) that share identifier values.



Details and Download:

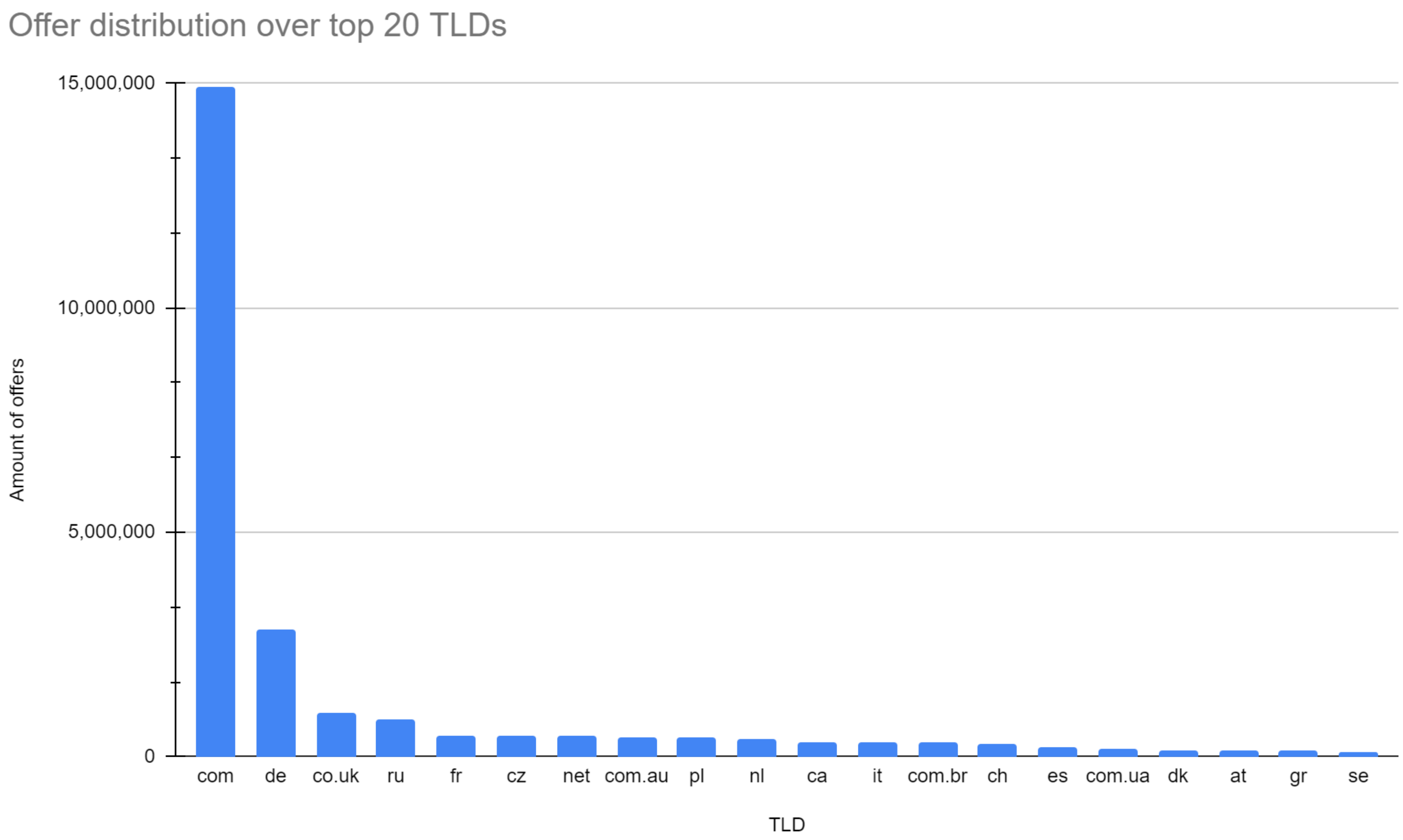
- http://webdatacommons.org/largescaleproductcorpus/v2/
- Primpeli, A., Peeters, R., & Bizer, C.: **The WDC training dataset and gold standard for large-scale product matching.** In: Companion Proceedings of the 2019 World Wide Web Conference. pp. 381-386 ACM (2019).

Distribution of Offers per Category in the English Training Set



source: http://webdatacommons.org/categorization/index.html

Distribution of Offers in WDC Product Corpus by TLD



Organization

Duration: 6 months (01.11.2020 – 30.04.2021)

Participants: 5 people

Type of work: Team and subgroup based

Milestones: 4 project phases

ECTS Points: 12

Evaluation

- Intermediate presentations
- Final report
- Individual contribution to the deliverables

Questions and Subtasks

1. Which (sub-)categories to consider? → **Data Selection and Profiling I**
2. (How to integrate the schemata of the different sources? → **Data Selection and Profiling I**)
3. Which products to select (enough train/test data available?) → **Data Selection and Profiling II**
4. How to split product offers for training and testing sets? → **Data Selection and Profiling II**
5. Which baseline methods to consider? → **Experiments**
6. Which Transformer-based models to try? → **Experiments**
7. How do the different methods compare? → **Experiments**
8. What can you learn from the results? → **Explanation**
9. How can you explain the results? → **Explanation**

Phase 1: Data Selection and Data Profiling I

Participants: all team members

Duration: 22.10. – 13.11.

Deliverables: 20 min. presentation, code & data

Input: WDC Large-scale Product Corpus (*provided here*)

Goal: Identify **3** languages in addition to English and collect enough product offers

Tasks:

1. Find product (sub-)categories for which:
   - matching cannot be solved without using non-english words (i.e. by using language-agnostic words)
   - **enough** (see slide 31) offers exist in English and **3** other languages
   - crawl for more offers in selected shops if necessary (use product identifiers from corpus)
2. Decide on 2 (sub-)categories, 1 more structured, 1 less structured
3. Profile the clusters/offers of these (sub-)categories (features/density/similarity)

Languages and how to detect them

- Apart from English, we want to collect offers in **3** additional languages
  - stay within Latin alphabet, no Chinese, Arabic, …, letters or signs
  - *recommended:* take those languages which appear in the corpus most frequently
  - important: For at least one of the 3 foreign languages, we want to have a sizeable amount of examples (see slide 34ff.), so we can also build a training set for that language
- How to detect languages?
  - Use 3-step approach:
    1. Use TLD for selection of relevant clusters (you can find the ID-URL mapping here)
    2. Apply language detection algorithm of your choice to offers of that cluster
    3. For further offers (i.e. from .com domain) run algorithm from 2. over respective clusters
  - Example algorithms:
    - Dictionary-based approach (likely too simple)
    - Machine-learning approach like fastText language detection

Phase 1: How to get started?

- Get the offers of the english corpus

  http://webdatacommons.org/largescaleproductcorpus/v2/

  File: offers_corpus_english_v2_non_norm.json.gz

- Get the offers of the full corpus

  http://webdatacommons.org/largescaleproductcorpus/v2/

  File: offers_corpus_all_v2_non_norm.json.gz

- Get acquainted with data, attributes, etc.
- Start by looking at English/non-English offers from each available category and identify categories which depend on non-English words for correct classificiation
- Select one rather structured and one unstructured category
- Profile available clusters of these categories

Phase 1: Deliverables

- Statistics about the data you have selected:
  - Which languages did you choose and why?
  - Which (sub-)categories did you choose and why?
  - How many different clusters (products) did you find per category?
  - What is the distribution of cluster-sizes (overall and per language)
  - How many clusters contain offers in 2, 3, 4 languages? (cross-language clusters)
  - Example of 3 hard matches and 3 hard non-matches for each language per category
  - What additional shops did you crawl? Resulting amount of additional offers?
- Must haves:

  Histogram showing the cluster-size distribution per category
    - Overall
    - Per language

Phase 2: Data Selection and Profiling II

Participants: two subgroups (one subgroup per selected category)

Duration: 14.11. – 11.12.

Deliverables: 30 min. presentation, code & data

Input: Selected product categories and relevant clusters

Tasks:

1. Select products (clusters) following the rules on next slide
2. Build training sets and testing sets for both categories
   - Framing the problem as multi-class classification (label = cluster_id or GTIN)
   - Framing the problem as a binary pair-wise matching problem (label = match/non-match)

3. Produce statistics for both types of sets

Phase 2: How to select suitable clusters?

- Try to find/crawl 150 clusters that fulfill the following criteria:
  - Contain (non-exact) offers for all languages (English + 3 selected languages)
  - Contain at least 15 (10 training / 5 testing) offers for English as well as at least 10 (5 training / 5 testing) for one selected language other than english
  - Contain at least 5 offers for the remaining 2 languages
  - Make sure that for every cluster you select, you also select at least 3 clusters containing very similar products (e.g. *iPhone 6* vs *iPhone 6s*)

    → Hard to distinguish corner-cases, otherwise the matching can be trivial

- Ways to find similar clusters:
  - Use keyword search, e.g. model name
  - Calculate similarity metric and order by similarity

Phase 2: How to build training and test sets (multi-class)?

– Select from each previously identified cluster:
  - At least 15 offers for training (10 for English / 5 for selected other language)
  - At least 20 offers for testing (5 for each of the 4 languages)

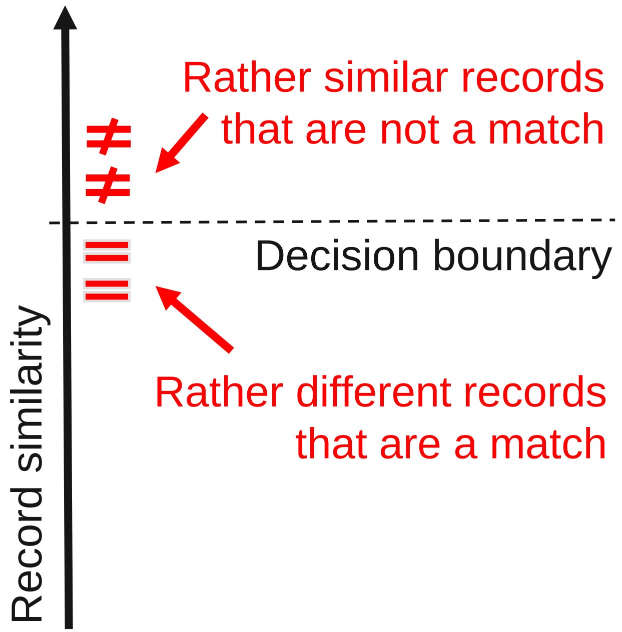
  → important: You have to manually check that the testing offers actually belong in this cluster, otherwise you will introduce noisy labels in your testing data!

Result: Training and Test sets for the multi-class case

Phase 2: How to assemble good training data (pair-wise)?

For English and one selected language:

– Build Training data which should:
  1. be balanced as random pairs would be highly skewed towards non-matches
  2. contain corner cases as they are most informative
     - especially “near-miss” negative examples are more informative for training than randomly selected pairs which tend to be “easy” non-matches.
     - iPhone 6 vs. iPhone 6s
     - rule of thumb: 50% corner cases
     - match offers using several simple matching techniques (e.g. Jaccard) and sort offer pairs according to their similarity



– You can use the weak supervision of the cluster_ids to automatically build training sets
– You do not need to manually check each pair

Phase 2: How to assemble good testing data (pair-wise)?

For each language:

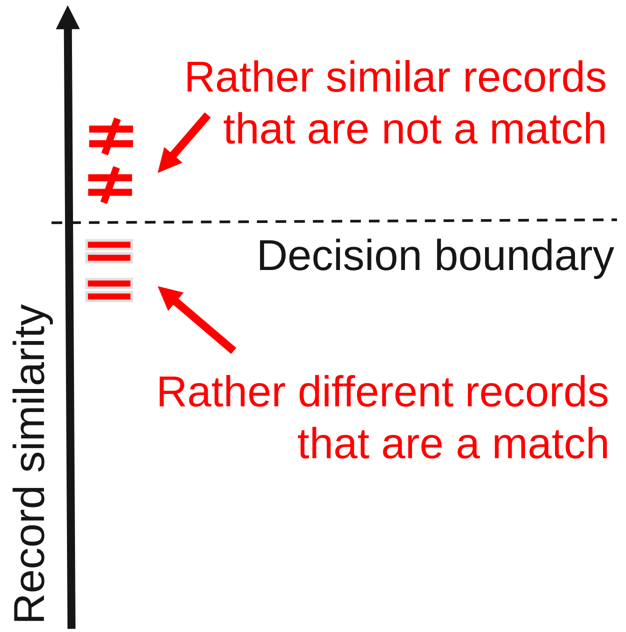
– manually label a set of record pairs (e.g. 500 pairs) including corner cases

– Use the following rule of thumb:

1. matching record pairs (25% of GS)
2. non-matching record pairs (75% of GS)

→ 50% of each group corner-cases
the other 50% random pairs

– You have to verify these pairs manually!



Result: Training and Test sets for the pairwise matching task

Phase 3: Experiments

Participants: two subgroups (one subgroup per selected category)

Duration: 12.12. – 22.01.

Deliverables: 30 min. presentation, code & data

Input: Train and test sets from phase 2

Tasks:

1. Design baseline methods (e.g. TFIDF-based features combined with Random Forest learner) including a mono-lingual Transformer Research and select suitable multi-lingual Transformer models
2. Come up with an experiment plan **and send it to us by 18.12**
3. Implement methods and start running experiments

→ More specifics on next slide

Phase 3: Experiments specifics

- Fine-tune Transformers and train baselines on
  - non-English product data (of one language)
  - non-English product data (of one language) + English product data

  → Tells us if and how much better we get on a language without readily available training data when we augment it with more easily available training data in english

  - just English product data

  → Tells us if only using English training data is enough for good performance on other languages (zero-shot performance)

- Evaluate models on
  - Test sets of single languages

Phase 4: Explanation

Participants: two subgroups (one subgroup per selected category)

Duration: 23.01 – 05.03

Deliverables: 30 min. presentation, code & data

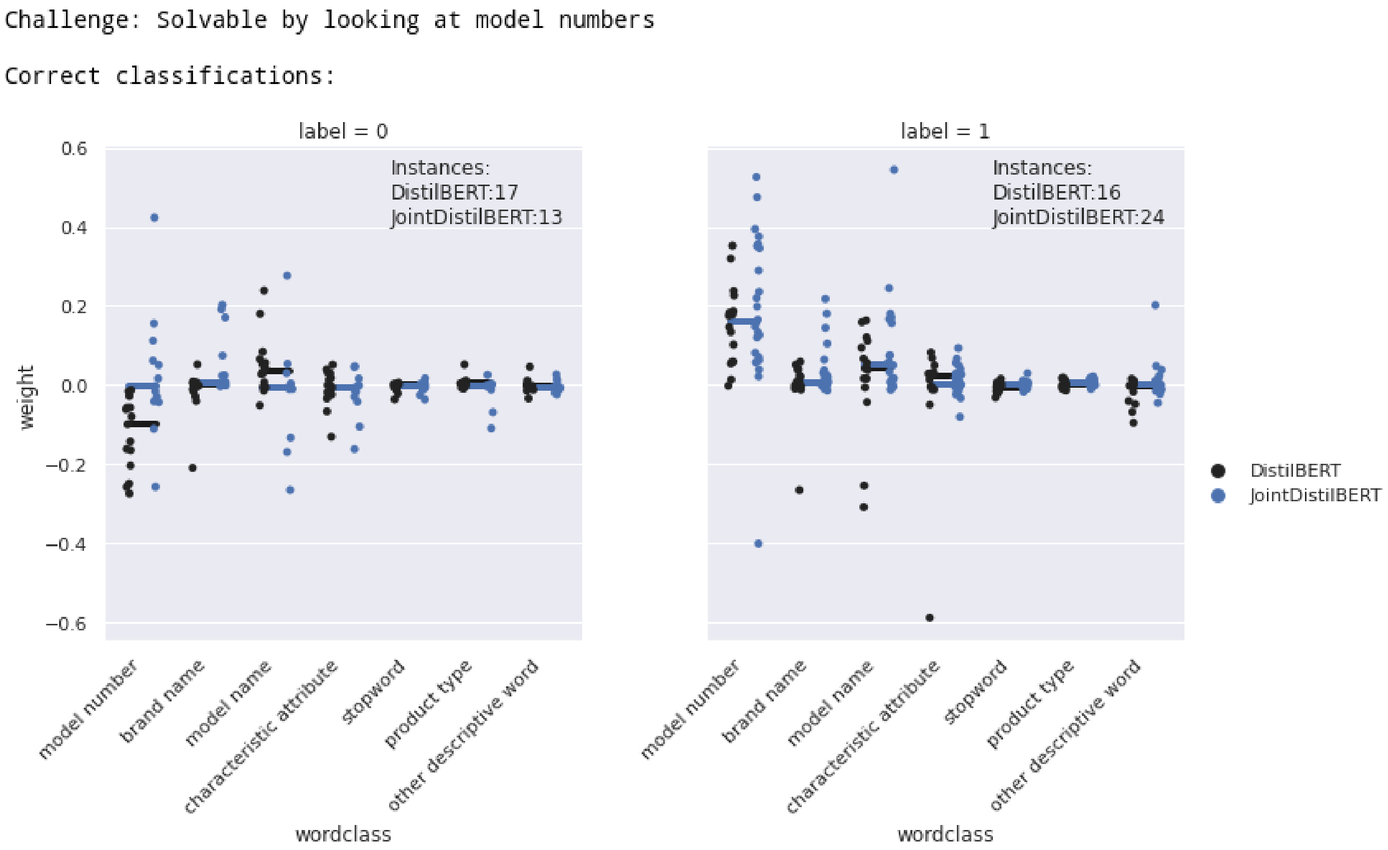
Input: Results from Phase 3

Tasks:

1. Error analysis
   - Look at correctly/incorrectly classified examples for select models
   - Come up with error classes and sort examples into them
   - Calculate statistics for error classes
2. Explanation
   - Apply explainability algorithm to instances from 1 (e.g. LIME)
   - Aggregate single explanations to allow you to make general statements about each model

What an Explanation may look like

– Example of work-in-progress explanation from ongoing research at our chair
  - We use LIME to explain instances and then aggregate by wordclasses
  - Allows us to see what each model focuses on



Schedule

Date	Session
Thursday, 22.10.2020	Kickoff meeting (today)
	Phase 1 (all members): Data Selection and Data Profiling I
Friday, 06.11.2020 (17:30)	Meet Ralph and report plan/current results
Friday, 13.11.2020	1st Deliverable: 20 minutes presentation, code & data - Subgroup formation
	Phase 2 (in 2 subgroups): Data Selection and Data Profiling II
Friday, 27.11.2020 (17:30)	Meet Ralph and report plan/current results
Friday, 11.12.2020	2nd Deliverable: 30 minutes presentation, code & data
	Phase 3 (in 2 subgroups): Experiments
Monday, 04.01.2021 (17:30)	Meet Ralph and report plan/current results
Friday, 22.01.2021	3rd Deliverable: 30 minutes presentation, code & data
	Phase 4 (in 2 subgroups): Explanation
Friday, 12.02.2021 (17:30)	Meet Ralph and report plan/current results
Friday, 05.03.2021	4th Deliverable: 30 minutes presentation, code & data
Friday, 30.04.2021	Final Report Submission

Formal Requirements & Consultation

Deliverables

1. On the deliverable dates provide us via e-mail with:
   - **Presentation slides**
   - **Task to member report**: excel sheet stating which team member conducted which subtask
   - **Code/Data**: link or zipped folder with your code and data

2. **Final Report**
   - **15 pages** including appendices, not including the bibliography
   - every additional page reduces your grade by 0.3
   - Created with Latex template of the Data and Web Science group (https://www.uni-mannheim.de/dws/teaching/thesis-guidelines/)

All deliverables should be sent to Chris & Ralph!

Formal Requirements & Consultation

Final grade

- 20% for every phase
- 20% for final report
- Late submission: -0.3 per day

Consultation

- Send one e-mail per team or subgroup stating your questions to Ralph

Useful Software

- Entity Resolution
  - HuggingFace Transformers: https://huggingface.co/transformers/
  - DeepMatcher : https://github.com/anhaidgroup/deepmatcher
  - Magellan: https://sites.google.com/site/anhaidgroup/projects/magellan
- Crawling
  - Scrapy: https://scrapy.org/
- Processing and GPUs
  - Google Colab: https://colab.research.google.com/
- Team Cooperation
  - GitHub/Lab
  - Video-Chat application of your choice

Related Work: (Multi-lingual) Transformers

- Guillaume Lample and Alexis Conneau, “Cross-lingual Language Model Pretraining,” arXiv:1901.07291 [cs], Jan. 2019
- A. Conneau et al., “Unsupervised Cross-lingual Representation Learning at Scale,” arXiv:1911.02116 [cs], Apr. 2020
- S. Wu, A. Conneau, H. Li, L. Zettlemoyer, and V. Stoyanov, “Emerging Cross-lingual Structure in Pretrained Language Models,” arXiv:1911.01464 [cs]
- Ziqi Zhang, Christan Bizer, Ralph Peeters and Anna Primpeli, MWPD2020: Semantic Web Challenge on Mining the Web of HTML-embedded Product Data @ISWC2020, to be published
- Ralph Peeters, Christian Bizer, and Goran Glavaš. 2020. Intermediate Training of BERT for Product Matching. In DI2KG Workshop @ VLDB2020.
- Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. arXiv:2004.00584 [cs] (April 2020).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, et al. 2017. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems. 6000–6010.

Related Work: (Multi-lingual) Transformers

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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. 4171–4186.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, et al. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 (2019).
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. arXiv:1910.01108 [cs] (2020).

Related Work: General (Deep) Data Matching (1/2)

- Anna Primpeli, Ralph Peeters, and Christian Bizer. 2019. The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. In Workshop on e-Commerce and NLP (ECNLP2019), Companion Proceedings of WWW. 381–386
- Ralph Peeters, Anna Primpeli, Benedikt Wichtlhuber, and Christian Bizer. 2020. Using schema.org Annotations for Training and Maintaining Product Matchers. In Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics.
- Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, et al. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In Proceedings of the 2018 International Conference on Management of Data. 19–34.
- Kashif Shah, Selcuk Kopru, and Jean David Ruvini. 2018. Neural Network Based Extreme Classification and Similarity Models for Product Matching. In Proceedings of the 2018 Conference of the Association for Computational Linguistics, Volume 3 (Industry Papers). 8–15.
- Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity Resolution. Proceedings of the VLDB Endowment 11, 11 (2018), 1454–1467.
- Peter Christen. 2012. Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer-Verlag, Berlin Heidelberg.

Related Work: General (Deep) Data Matching (2/2)

– Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2019. End-to-End Entity Resolution for Big Data: A Survey. arXiv:1905.06397 [cs] (2019).

– Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. Proceedings of the VLDB Endowment 3, 1-2 (2010), 484–493.

– Petrovski, Petar, Volha Bryl, and Christian Bizer. "Integrating product data from websites offering microdata markup." *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014.

– Ristoski, Petar, and Peter Mika. "Enriching product ads with metadata from HTML annotations." *International Semantic Web Conference*. Springer, Cham, 2016

Questions?