A Hybrid Model Words-Driven Approach for Web Product Duplicate Detection

Marnix de Bakker, Flavius Frasincar, and Damir Vandic

vandic@ese.eur.nl
Erasmus University Rotterdam

ERASMUS UNIVERSITEIT ROTTERDAM

Introduction

- Duplicate detection of products
- Aggregation of Web product offerings
- Type of data (title vs attributes)
- Example of two titles referring to same product:
  - Samsung - 40" Class / LCD / 1080p / 60Hz / HDTV
  - Samsung 40" 1080p 60Hz LCD HDTV LN40D503

Algorithms

We investigate three algorithms:

- Title Model Words Method
  D. Vandic et. al. *Faceted Product Search Powered by the Semantic Web* Decision Support Systems, 53(3):425–437, 2012.
- Hybrid Similarity Method (proposed)
- TF-IDF Duplicate Detection

Title model words method

The main steps (high-level):

1. First, perform a word-based cosine similarity check
2. Search for a model word pair where the non-numeric parts are *approximately* the same, but the numeric parts are different
3. Otherwise, compute alternative average weighted similarity between title names

Title model words method

Example 1

- ‘Samsung - 46” Class/ LED / 1080p / **120Hz** / HDTV’

vs.

- ‘Samsung - 46” Class/ LED / 1080p / **200Hz** / HDTV’

Title model words method

Example 2

- ‘Samsung - **55”** Class/ LED / 1080p / 120Hz / HDTV’

vs.

- ‘Samsung - **46”** Class/ LED / 1080p / 120Hz / HDTV’

Hybrid Similarity Method

- Extends the Title Model Words Method
- Deals effectively with product attributes, stored as key/value pairs (KVP's)
    - e.g. ('Weight', '20.5 lbs.')
- Designed for:
    - title and product attributes (KVP's)
    - two sources of product descriptions

Hybrid Similarity Method

- Assumption: no duplicates within one Web shop
- Main idea:
  - Put each product from Web shop 1 in own cluster
  - Try to match each product from Web shop 2 to a cluster
  - Considers only clusters with size 1

Hybrid Similarity Method

- First try to find a match using Title Model Words Method
- If this fails:
  - compute the *hybrid similarity* and cluster the two products if its higher than a threshold

Hybrid Similarity (1)

Part 1: similarity between values for matching keys

- Consider all pairs of KVP’s, if keys match update running average with similarity between values
- We experimented with cosine similarity and the Jaro-Winkler similarity measure

Hybrid Similarity (2)

Part II: use model words from values

For all non-matching pairs of KVP's:

- compute percentage of matching model words (extracted from the values)
- ignore keys in this computation

Hybrid Similarity Method

Final similarity:

$$hybridSim = \theta \times avgSim + (1 - \theta) \times mwPerc$$

where

- θ is a weighting factor
- avgSim is the average similarity based on the matching keys (the first part)
- mwPerc is the matching model words percentage (the second part).

Hybrid Similarity Method

Example differently structured data

- TV from Bestbuy.com has the KVP:
  [‘Product Weight’,
  ‘19.1 lbs. with stand (16.9 lbs. without)’
  ]
- Same TV on NewEgg.com:
  [‘Weight Without Stand’, ‘16.9 lbs.’]
  [‘Weight With Stand’, ‘19.1 lbs.’]

TF-IDF Method

- Employs TF-IDF,
  - TF is the number of times that a term occurs in the attribute values
  - IDF is the logarithm of the total number of products divided by the number of products containing the term.
- Cosine similarity with a threshold

Evaluation setup

- Data set of 282 TV’s from two Web shops
  - BestBuy.com and NewEgg.com
- There are 82 pairs (164 products) that are duplicates
- 20 random test sets (10% of total size)
- Wilcoxon signed rank test

Evaluation results

Method	Average F1-measure	Average precision	Average recall
Title model words	0.357	0.556	0.279
TF-IDF	0.201	0.433	0.133
Hybrid Similarity	0.656	0.741	0.647

Evaluation results

H0: row < col

p-values	Title model words	TF-IDF	Hybrid Similarity
Title model words	X	0.989	0.000
TF-IDF	0.049	X	0.000
Hybrid Similarity	1.000	1.000	X

Conclusions and future work

- Proposed a duplicate detection method that uses also key/value pairs
- Benchmarked against existing approaches
- Hybrid Similarity method is best performing on F1
- TF-IDF is performing surprisingly well

Conclusions and future work

Future work

- Experiment with more similarity measures
- Use semantics of product attributes/values
- Focus on efficiency (scalability)

Questions?