Linking via Fuzzy Matching

Alexey Osipov
Sidenis, Swiss Re Institute

11.12.2020

Affiliation.

Casualty R&D

Problem.

We would like to link a database dataset with an input dataset by text fields:

Test cases.

1. Matching by meaningful content.
2. Matching by short names.
3. Matching with unstructured text.

Matching by meaningful content.

Statement of the problem

We have a database of industry classifications.
We would like:

- to classify a new object,
- to link classifications.

Key features:

- Synonyms are important.
- We can give up to 5 recommendations.

Example of linking.

Example

In database:

- **Establishments primarily engaged in the production of rice**
- **Growing of rice (including organic farming and the growing of genetically modified rice)**
- **Establishments primarily engaged in performing crop planting, cultivating, and protecting services.**

A new object:

- **Rice factories.**

How to link?

Key points.

- Preprocessing.
- Embeddings.
- Word2vec, GloVe.
- Embedding for a phrase, e.g., by averaging.
- Compare via cosine similarity.
- Other phrase embeddings: doc2vec.

R packages.

- word2vec (word2vec, doc2vec)
- text2vec, textTinyR

Linking by short phrases.

Statement of the problem

We have a database of large companies.
We would like to understand if a company belongs to the database.

Key features:

- All names are not so long.
- Misprints are possible.
- Same names can be represented in different ways.

Example of linking.

Example.

We have data like

- **BP plc**
- **BHP**
- **British Petroleum**

We would like to link a phrase with it:

- **BP Amoco plc**

How to link?

Algorithm.

- Preprocessing/normalization.
- Fuzzy matching by string distances:
  1. Jaccard similarity.
  2. Jaro-Winkler.
  3. Levenshtein.
  4. Damereau-Levenshtein.
  5. Optimal string alignment.
  6. Longest common substring.
- Filtering by threshold values.
- Disambiguation logic.

Relevant R packages.

R packages.

- stringdist
- hash
- megalodon

Linking with unstructured text.

Statement of the problem

We have a database of claims. A claim name is unstructured text. We would like to link a company with the corresponding claims.

Key features.

- Large size of the database.
- Not all text fields are relevant.
- Linking with unstructured text.
- Misprints are possible.
- Same names can be represented in different ways.
- Text fields can be quite long.

Examples.

Examples.

In database we have

- **Grupo Acerinox, S. A. 257768**
- **Cyclone Klaus**
- **DO NOT USE Sweco Lietuva UAB**

We would like to link it with company like:

- **Acerinox**

General scheme.

Scheme.

1. Classification block (distinguish noisy entries).
2. Entity extraction block.
3. Linking block

Classification block

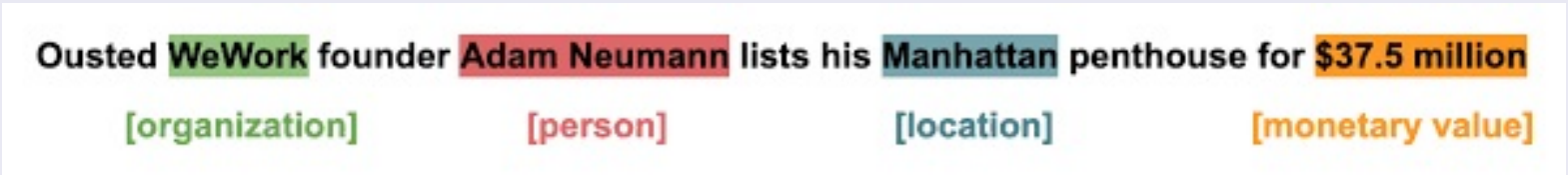
Key points:

- features from the other fields of the database
- features from claim name
- features based on presence of key words
- processing of features
- manually labelled dataset for supervised classification
- xgboost

Entity extraction block.

Key points

- spacy NER model, spacyr



- regular expressions

Linking block.

Key problems.

- Cross product is large.
- The text field have different size (the database one is large, the input text one is smaller).

Algorithm.

- calculate metric that can be computed fast (TFIDF-based)
- calculate second level metrics (jaccard-based)
- choose threshold via clustering and manual validation
- k-means from stats, hdbscan from dbscan
- disambiguation block

Thank you very much!

Messages.

- Sometimes taking into account synonyms helps, sometimes it does not.
- **megalodon** can be used to hunt for large companies.
- We can get better results by taking into account the specifics of the problem.