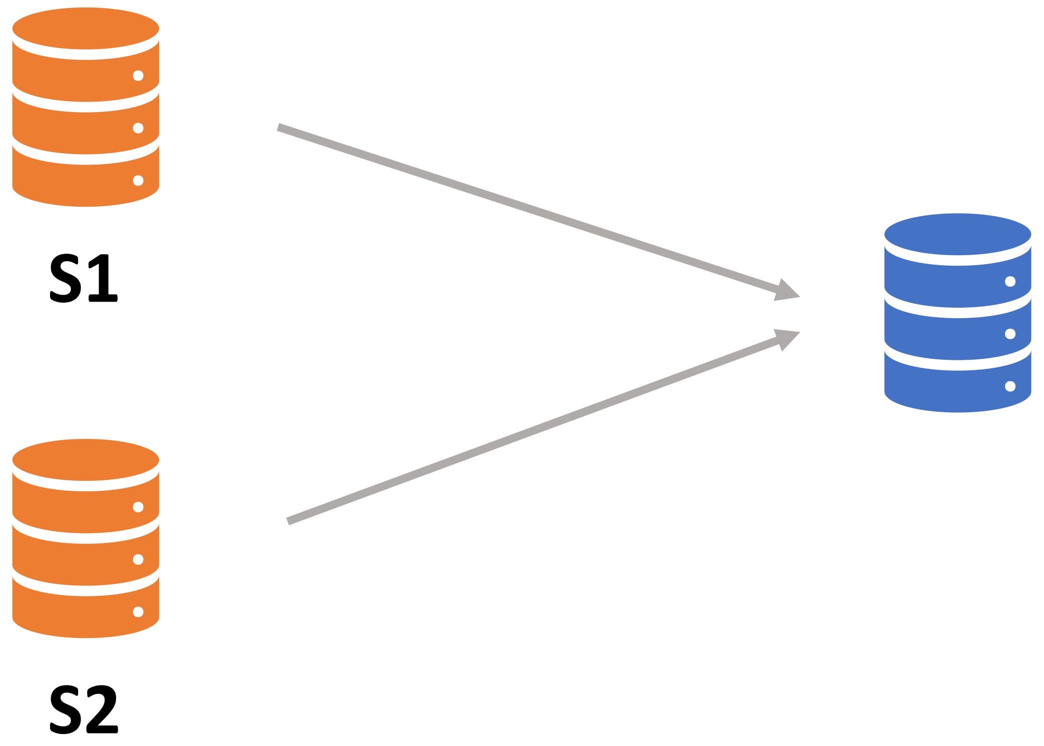
(Deep) Entity Resolution

Problem definition

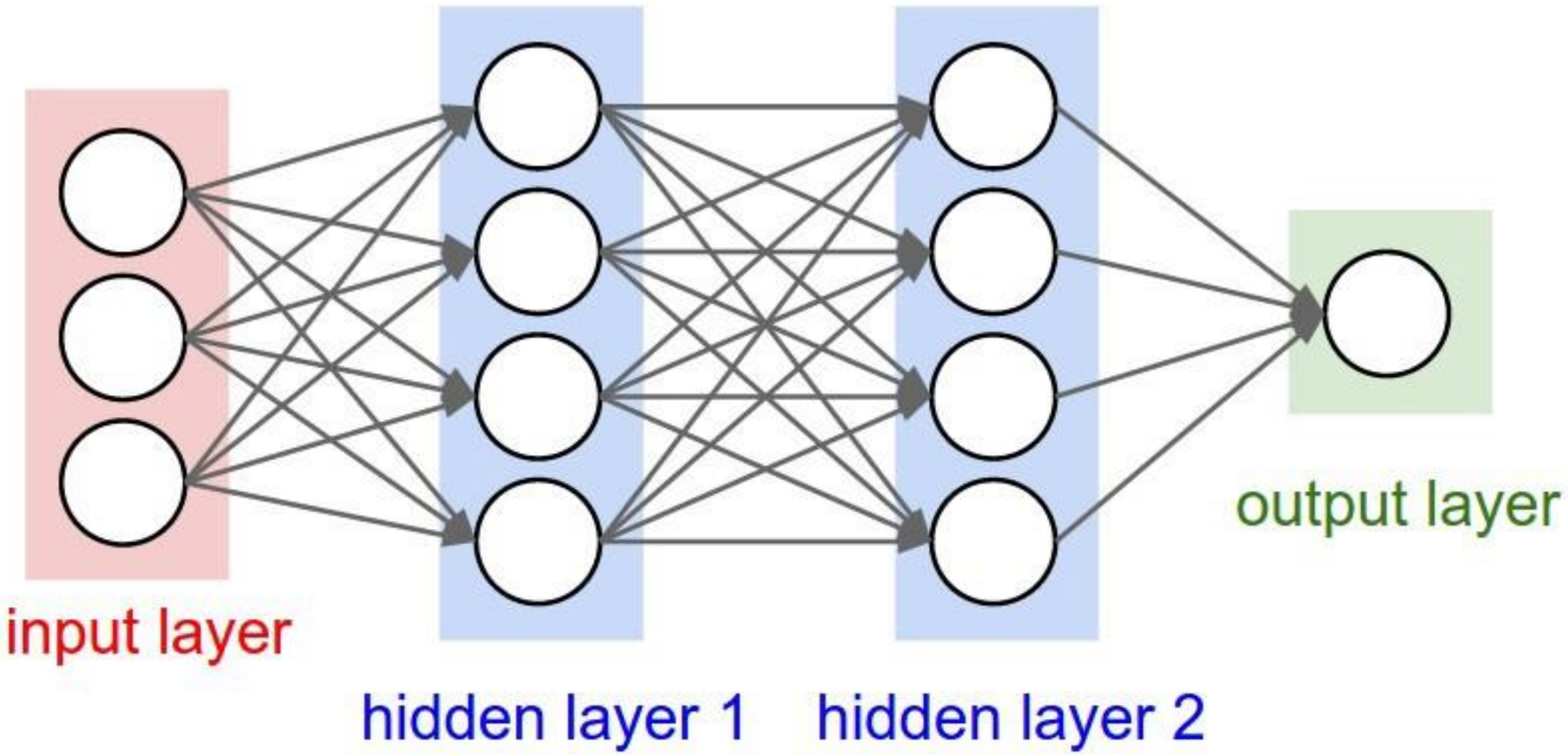
- Entity resolution: the process of identifying and merging records judged to rappresent the same real-world object.



Deep learning

A class of machine learning algorithms, based on artificial neural network architecture.

$f(x) = Wx + b$



Entity resolution process

1. Labelling entity subset
2. Learning rules / ML
3. Blocking
4. Applying ML rules to match entity pairs

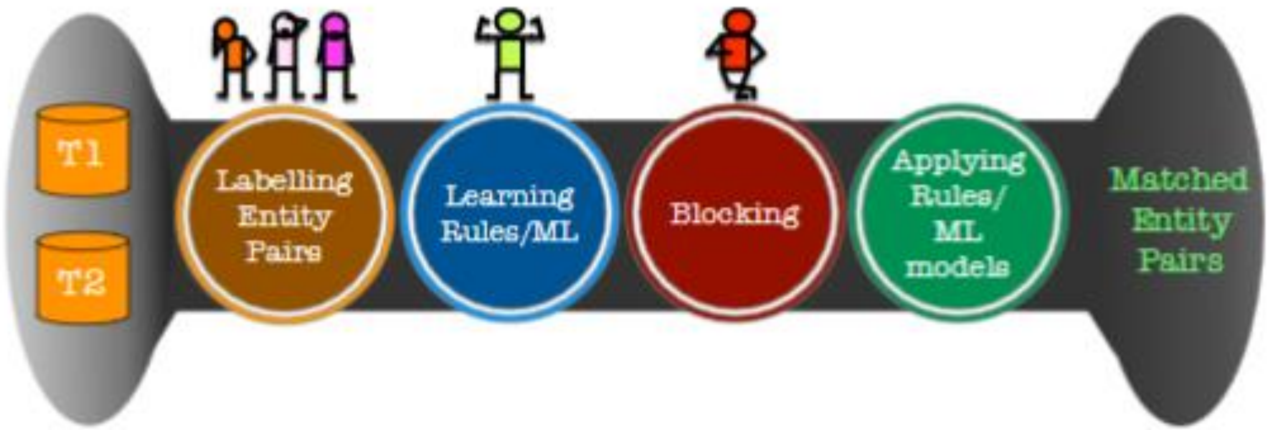


Figure 1: A Typical ER Pipeline

Model & framework

Distributed Representations of Tuples for Entity Resolution
https://arxiv.org/pdf/1710.00597.pdf (5 Aug 2018)

- PyTorch
- Record Linkage Toolkit

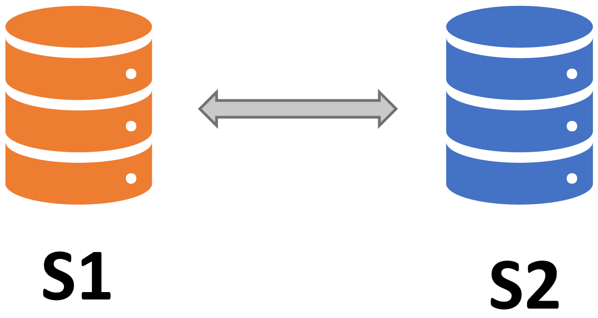
...nent system	Gott...
...stributed multimedia databases	Isabel F. C...
... Web, CORBA and databases	Athman Boug...
... with MIX	Chaitan Baru, Am...
...iented database system	Alexander Brodsky,
...ding mobility to PREDATOR	Phillippe Bonnet, Kyle
...ler of the cubetree storage organization	Nick Roussopoulos, Yan
...n engine	Michael B&#246
...proach to business process management	N. R. Jennings, T. J. Norma...
...ure for dynamic electronic markets	Benny Reich, Israel Ben-Sh...
...n of RMP: a virtual electronic market place	Susanne Boll, Wolfgang Kla...
...ting intelligence through online analytical web usage m...	Alex G. B&#252
...erce: assessment of current practices and future directi...	Martin Bichler, Arie Segev, J...
...lectronic catalogs	Sherif Danish
...rce: enabling the network economy	Bart Meltzer, Robert Glushk...
...c marketplace on the Web	Asuman Dogac, Ilker Durus...
...dleware for large scale data delivery	Mehmet Altinel, Demet Ak...
...nerator for Web information extraction	Ling Liu, Wei Han, David
... Internet	Reinhard Braumandl, A...
...INg Objects tracking	Ouri Wolfson, Prasad
...ronous transactions	Lyman Do, Prabhu P...
...soft repository	Thomas Bergstra...
... Telecom Italia	Stefano M. Tr...
...arehouse mechanisms	Matthias ...
...lity	E. A...
...rmation organization...	

Data sources

- Name:
  DBLP-ACM
- Source:
  Database Group Leipzig
  https://dbs.uni-leipzig.de/en
- Domain:
  Bibliographic
- Attributes:
  Id, title, authors, venue, year
- Tuples
  2.616 – 2.294

Types of entity resolutions

- **Clean-Clean**: each data source is duplicate free.
- Dirty-Clean: one of two sources contains duplicates.
- Dirty-Dirty: each source contains duplicates.



Entity representation

An **entity** is a real world object described by a fixed number of attributes.

Id	Authors	Title	Price	Venue
1	Bill William Gates	Follow your dreams	20	International

Id	Authors	Title	Price	Venue
3	Bill William G.	Follow dream	20	International

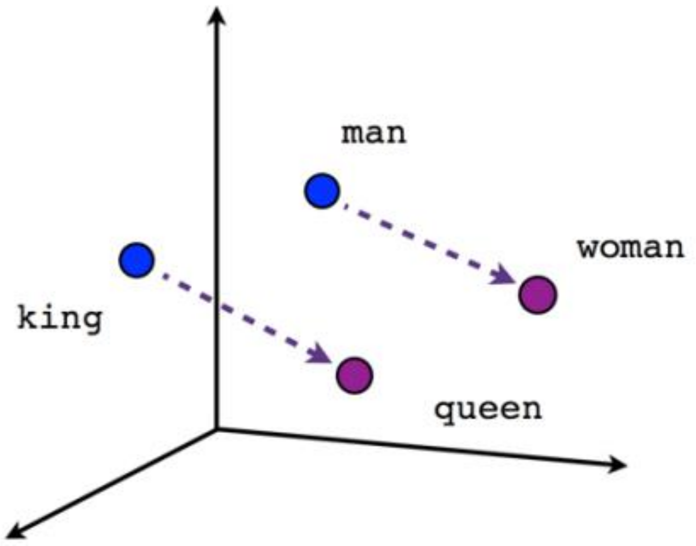
Comparing two entities means comparing attributes's words between two entities.

Word embeddings

The main idea is representing words as vectors and use similarity functions to compare words.

"Similarity" in this sense can be defined as:

- Euclidean distance (the actual distance between points in N-D space)
- Cosine similarity (the angle between two vectors in space).



Word embeddings

The simplest example of a word embedding scheme is a **one-hot encoding**. In a one-hot encoding, or “1-of-N” encoding, the embedding space has the same number of dimensions as the number of words in the vocabulary.

Vocabulary:
Man, woman, boy, girl, prince, princess, queen, king, monarch

	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

Each word gets a 1x9 vector representation

Word embeddings

We can create a more efficient 3-dimensional mapping for our example vocabulary by manually choosing dimensions that make sense.

Vocabulary:
Man, woman, boy, girl, prince, princess, queen, king, monarch

	Femininity	Youth	Royalty
Man	0	0	0
Woman	1	0	0
Boy	0	1	0
Girl	1	1	0
Prince	0	1	1
Princess	1	1	1
Queen	1	0	1
King	0	0	1
Monarch	0.5	0.5	1

Each word gets a 1x3 vector

Similar words... similar vectors

Word embeddings

The next step is to extend our simple 9-word example to the entire dictionary of words, or at least to the most commonly used words.

Forming N-dimensional vectors that capture meaning in the same way that our simple example does, where similar words have similar embeddings and relationships between words are maintained.

As such, various algorithms have been developed, some recently, that can take large bodies of text and create meaningful models. The most popular algorithms are:

- Word2Vect (Google)
- GloVE (Stanford)
- FastText (Facebook)

Word embeddings

GloVe: Global Vectors for Word Representation

(Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014)

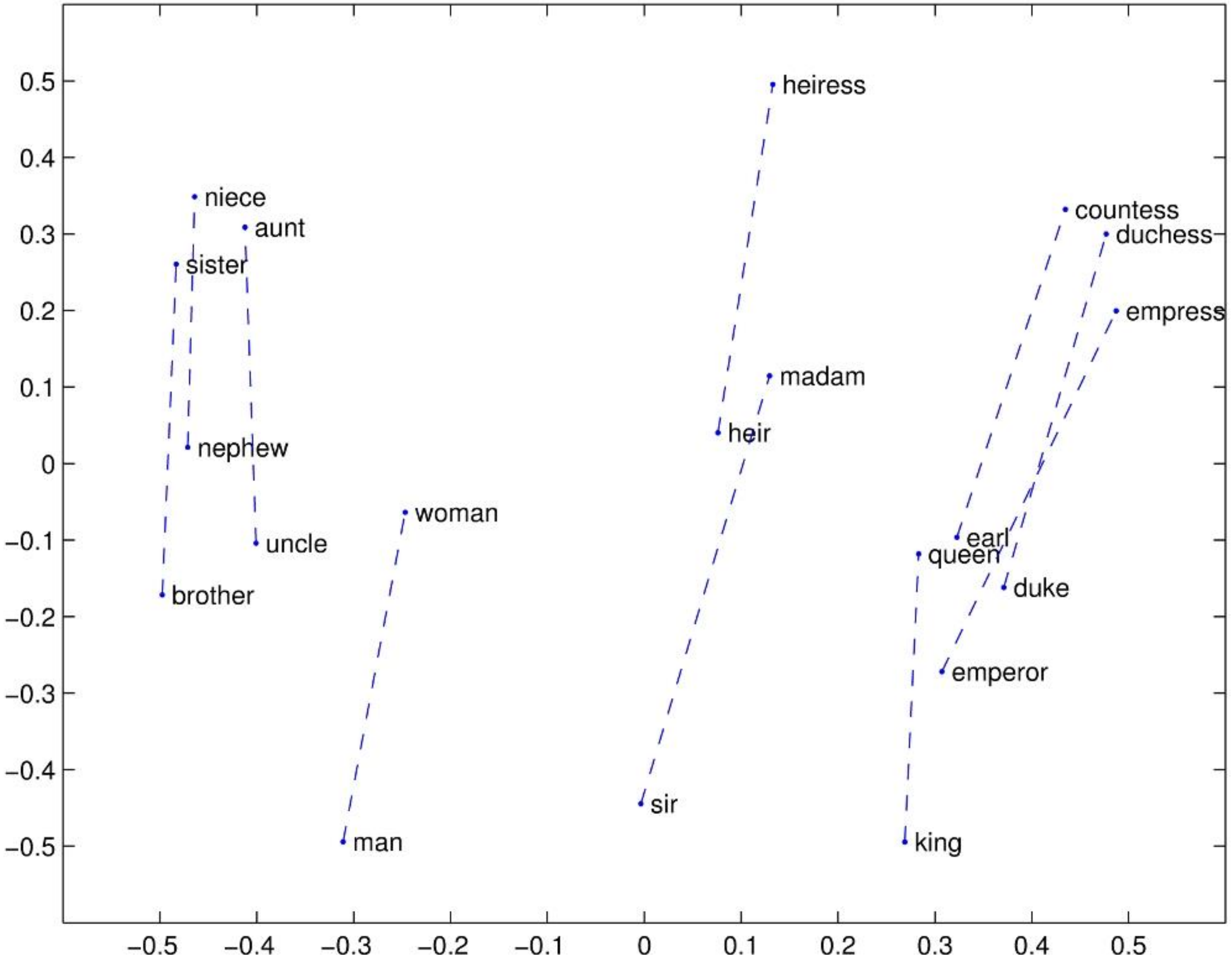
https://nlp.stanford.edu/projects/glove/

Wikipedia 2014 + Gigaword 5

6B tokens

400K vocab

50-dimensional vectors



Distributed representation of tuples

Id	Authors	Title	Price
1	Bill William Gates	Follow your dreams	20

Word	GLOVE
Bill	[0.4, 0.8, 0.9]
William	[0.3, 0.9, 0.7]
Gates	[0.5, 0.8, 0.8]

Authors	Distributed rappres.
Bill William Gates	[0.4, 0.83, 0.8]

For each attribute A_k of tuple t:

- Tokenize A_k into a set of words W
- Lookup for a token w_1 ∈ W in Glove
- $v_k(t)$: = average of vectors

Unknowns tokens & missing values

Id	Authors	Title	Price
23	Kauffman	Illuminae files	

Word	GLOVE
<unk>	[-0.79, 0.86, 0.11, …]
<nan>	[0.92, -0.31, 0.25, …]

Glove contains special token <unk> for unknowns words.

Distributional Similarity

Attributes	Values	Distributed rappres.
Id	1	[0.1, 0.1, 0.1]
Authors	Bill William Gates	[0.4, 0.83, 0.8]
Title	Follow your dreams	[0.53, 0.18, 0.67]

Attributes	Values	Distributed rappres.
Id	5	[0.1, 0.1, 0.1]
Authors	Bill William G.	[0.43, 0.82, 0.7]
Title	Follow dream	[0.43, 0.28, 0.61]

Attributes	Cosine Similarity
Id	-0.7
Authors	0.94
Title	0.95

For each pair of tuples (t, t'):

- Compute the distributed rappresentation for t and t'
- Compute their distributional similarity vector

Classifier

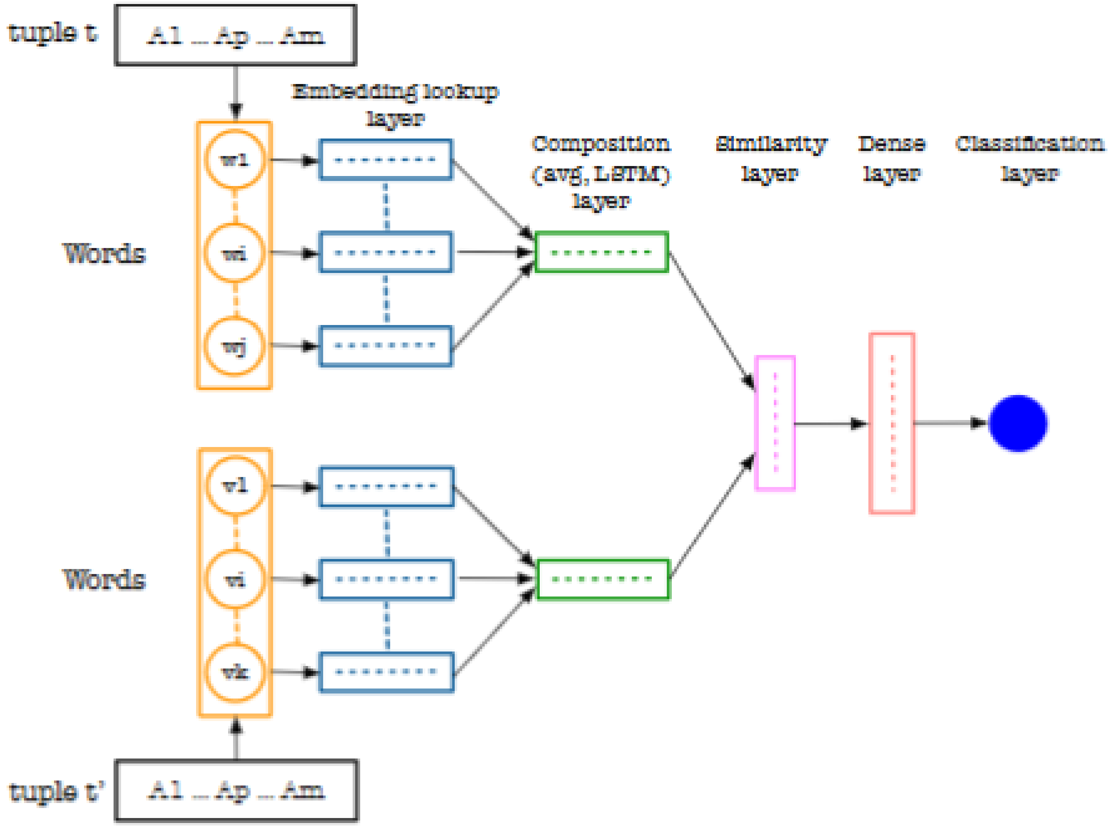


Figure 5: Deep Entity Resolution Framework

1. Input layer: similarity vector [1x5]
2. Hidden layer: fully connected[5x50]
3. Output Layer: binary vector[50x2]
4. Softmax()

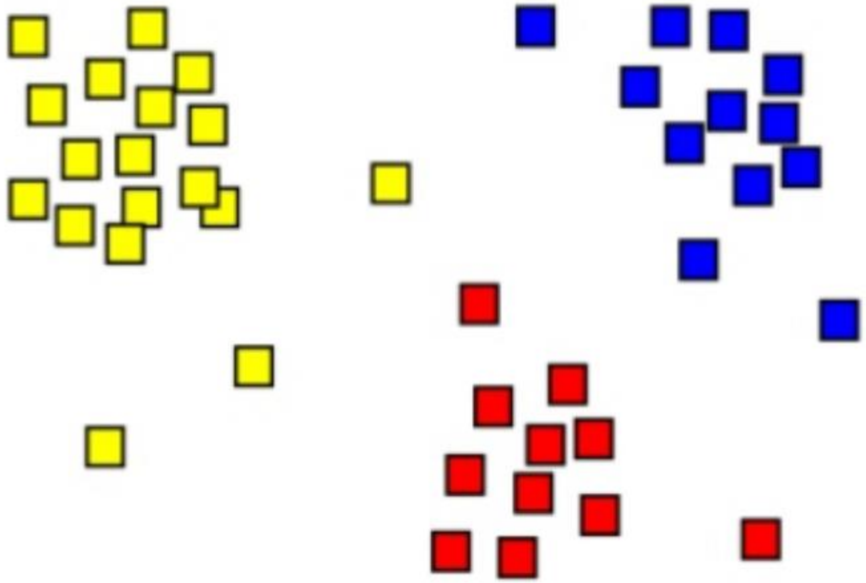
- Learning rate: 1e-4
- Loss function: negative log likelihood
- Batch size: 20

Blocking

It is not possible to compare all possible pairs of records!

- Group similar entities into blocks
- Execute comparisons only inside each block

1. Each profile is represented by one or more **blocking keys**
2. All profiles having the **same** blocking keys are placed in the same blocks



Blocking

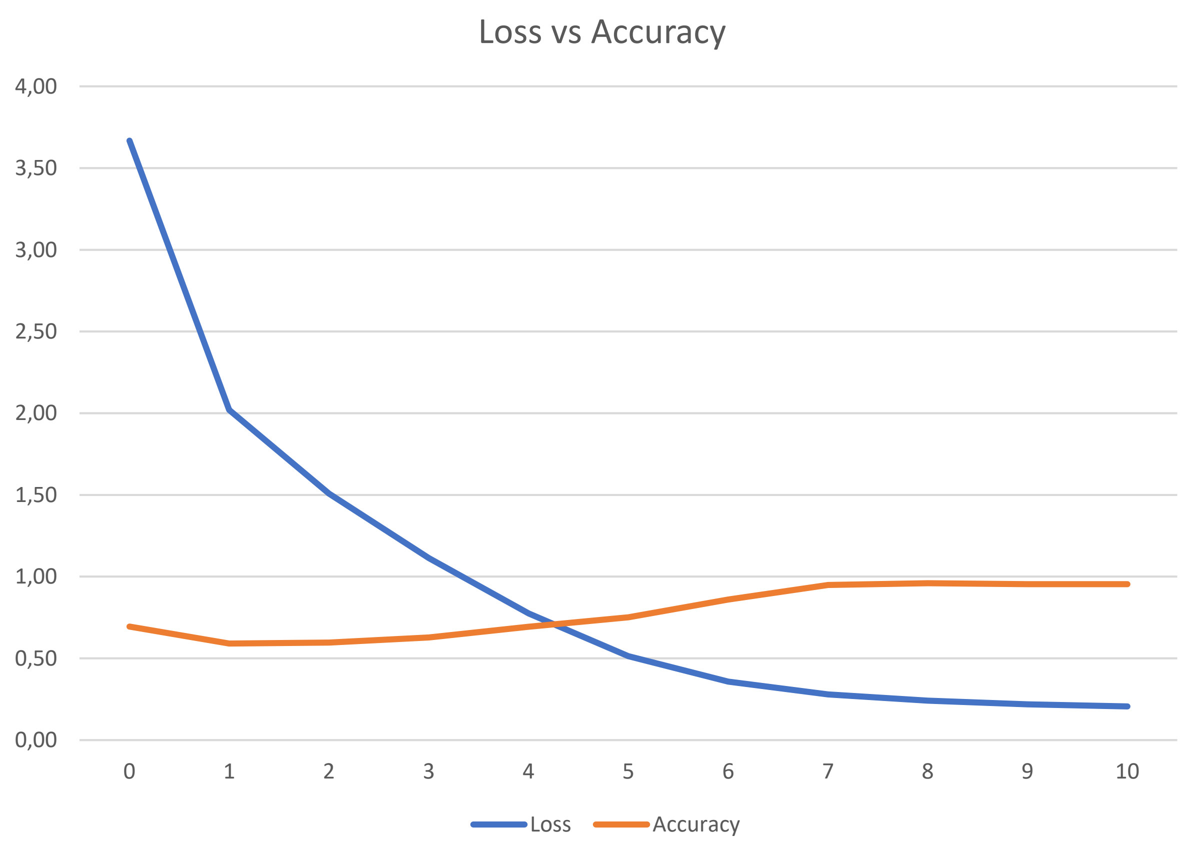
Cartesian product

- Candidate pairs: 6.001.104

Sorted Neighbourhood algorithm

- Window size = 5
- Candidate pairs: 7163
- Train set: 5372
- Test set: 1791

Training

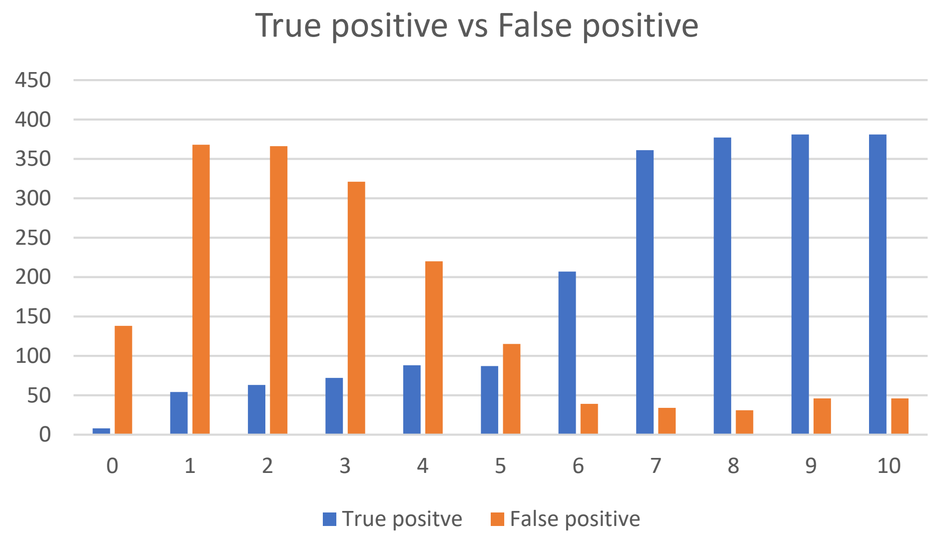


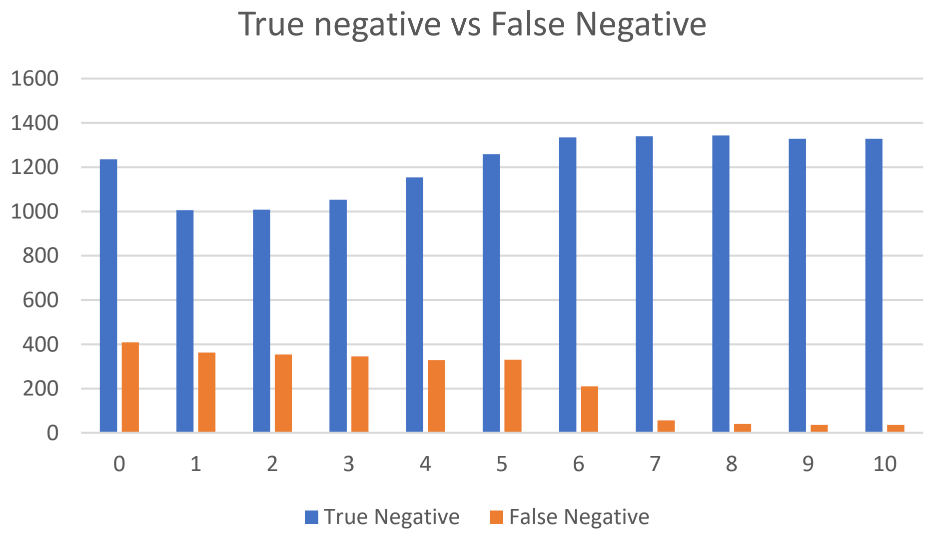
Epoch: 10

Loss: 0.206

Accuracy: 0.954

Training





True Positive: 381 89,22%
False Positive: 46 10,78%

True Negative: 1328 97,36%
False Negative 36 2,63%

Precision: 0.89
Recall: 0.91
F-score: 0,90

Tool

https://github.com/rs9000/DeepEntityMatching

How to use

```
usage: train.py [-args]

arguments:
  --source1           Source file 1
  --source2           Source file 2
  --separator         Char separator in CSV source files
  --n_attrs           Number of attributes in sources files
  --mapping           Partial ground truth mapping of sources files
  --blocking_size     Window size of the blocking method (Sorted Neighbourhood)
  --blocking_attr     Attributes of blocking
  --word_embed        Word embedding file
  --word_embed_size   Word embedding vector size
  --save_model        Save trained model
  --load_model        Load pre-trained model
```

Improve the model

- Pre-process tokens (parse text, remove stop-words etc.)
- Use a RNN instead of the average between token vectors
- Use a more efficient blocking method
- Use a bigger word-embedding