A Graph-based Blocking Approach for Entity Matching Using Contrastively Learned Embeddings

Presenter: **John Bosco Mugeni**
Supervisor: Toshiyuki Amagasa
University of Tsukuba

Published in (ACM SIGAPP) Applied Computing Review, 2022
Computer Science

Contents

Introduction: entity matching

- Entity Matching: is the task of discovering matching entries among disperate data sources.
- The goal is to then link these entries with a high-match quality
- However, the process meets quadratic complexity problem w.r.t dataset size

Table A

category	brand	model no.	price
garden - general	d-link	dcs-1100	99.82
furniture	3m	fr530cb	67.88
stationery & office machinery	brother	dk2113	64.88

Table B

category	brand	model no.	price
footrests	3m #	fr530cb #	67.34
file folder labels	avery	5029	14.2
surveillance cameras	d-link	dcs-1100	99.82

Figure: An example of matching tuples

Introduction: entity matching

- Entity Matching: is the task of discovering matching entries among disperate data sources.
- The goal is to then link these entries with a high-match quality
- However, the process meets quadratic complexity problem w.r.t dataset size

Table A

category	brand	model no.	price
garden - general	d-link	dcs-1100	99.82
furniture	3m	fr530cb	67.88
stationery & office machinery	brother	dk2113	64.88

Table B

category	brand	model no.	price
footrests	3m #	fr530cb #	67.34
file folder labels	avery	5029	14.2
surveillance cameras	d-link	dcs-1100	99.82

Figure: An example of matching tuples

Introduction: entity matching

- Entity Matching: is the task of discovering matching entries among disperate data sources.
- The goal is to then link these entries with a high-match quality
- However, the process meets quadratic complexity problem w.r.t dataset size

Table A

category	brand	model no.	price
garden - general	d-link	dcs-1100	99.82
furniture	3m	fr530cb	67.88
stationery & office machinery	brother	dk2113	64.88

Table B

category	brand	model no.	price
footrests	3m #	fr530cb #	67.34
file folder labels	avery	5029	14.2
surveillance cameras	d-link	dcs-1100	99.82

Figure: An example of matching tuples

Introduction: blocking

- "Blocking" is introduced for efficient execution of entity matching
- The naive pairwise comparison (right figure) requires exorbitant computation due to a massive search space in contrast to a partitioned search space due to "blocking" (left figure)

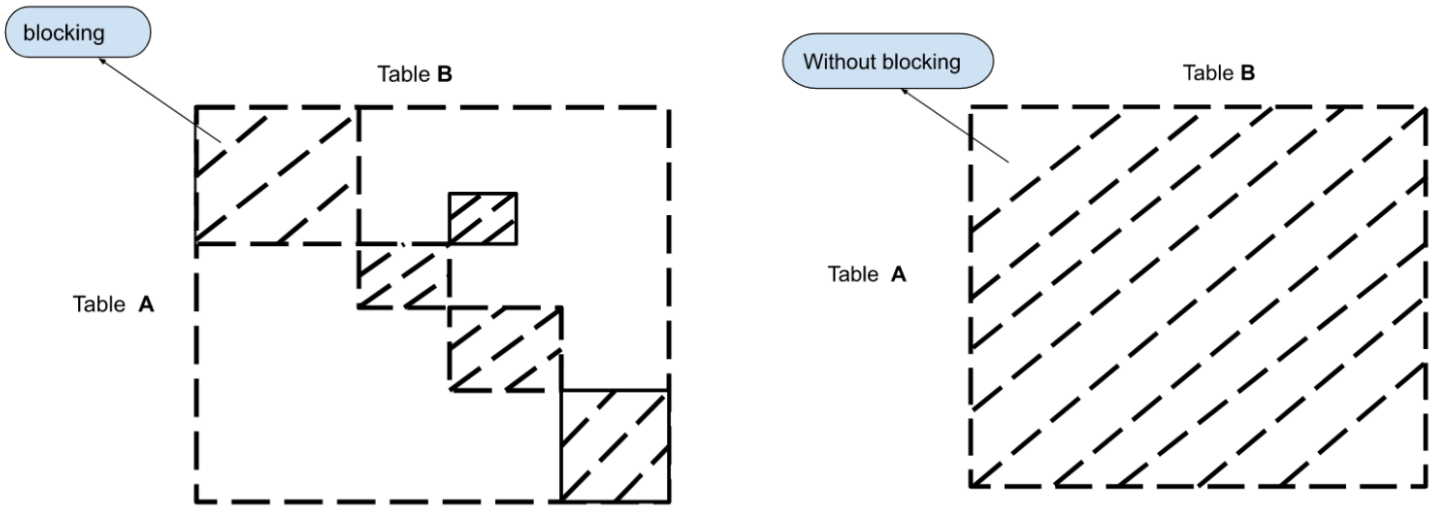


Figure: Types of blocking frameworks

Introduction: blocking

- "Blocking" is introduced for efficient execution of entity matching
- The naive pairwise comparison (right figure) requires exorbitant computation due to a massive search space in contrast to a partitioned search space due to "blocking" (left figure)

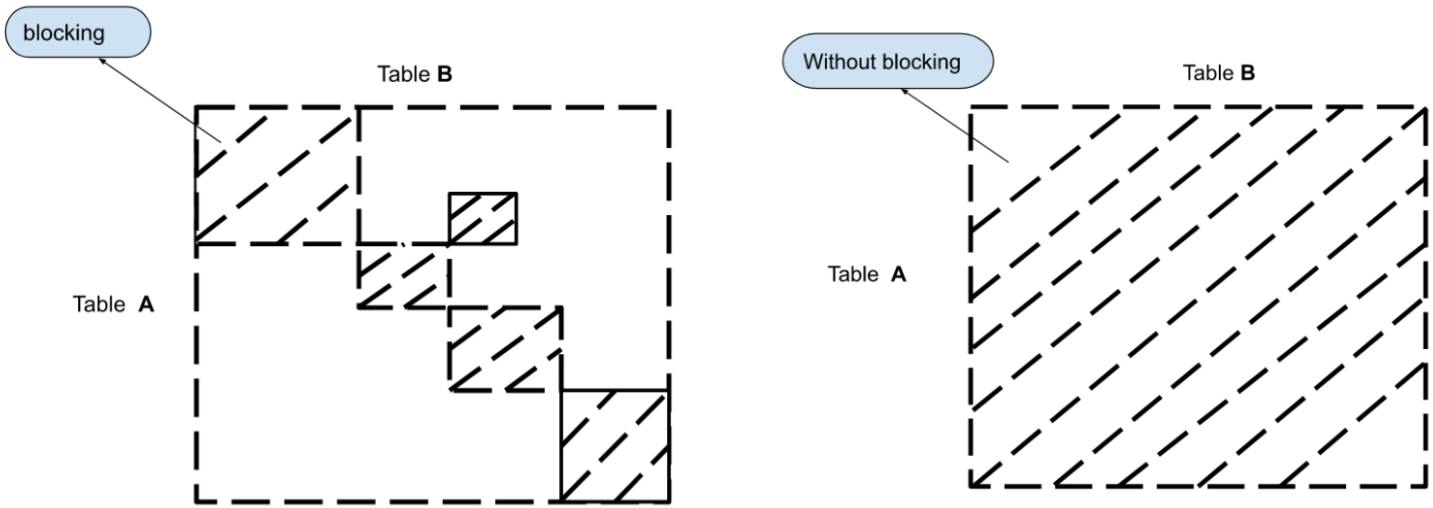


Figure: Types of blocking frameworks

Introduction: blocking techqniues

- "Blocking" techniques can be categorized into 3 types;

Figure: Types of blocking frameworks

- Rule-based methods require handcrafted features, domain knowledge & are labour intensive
- Learning-based methods have high accuracy but require labelled data (labels are not always available)
- Cluster-based methods circumvent the need of labels & handcrafted features

Introduction: blocking techqniues

- "Blocking" techniques can be categorized into 3 types;

Figure: Types of blocking frameworks

- Rule-based methods require handcrafted features, domain knowledge & are labour intensive
- Learning-based methods have high accuracy but require labelled data (labels are not always available)
- Cluster-based methods circumvent the need of labels & handcrafted features

Introduction: blocking techqniues

- "Blocking" techniques can be categorized into 3 types;

Figure: Types of blocking frameworks

- Rule-based methods require handcrafted features, domain knowledge & are labour intensive
- Learning-based methods have high accuracy but require labelled data (labels are not always available)
- Cluster-based methods circumvent the need of labels & handcrafted features

Introduction: blocking techqniues

- "Blocking" techniques can be categorized into 3 types;

Figure: Types of blocking frameworks

- Rule-based methods require handcrafted features, domain knowledge & are labour intensive
- Learning-based methods have high accuracy but require labelled data (labels are not always available)
- Cluster-based methods circumvent the need of labels & handcrafted features

Introduction: motivation for cluster blocking

- Existing solutions capture database interactions via traditional word embeddings.
- That means they assign the same vector to a word irrespective of context.
- E.g., The bank is located near the river bank.
- In contrast, context embeddings assign vectors dynamically thereby incorporating rich semantics.

Introduction: motivation for cluster blocking

- Existing solutions capture database interactions via traditional word embeddings.
- That means they assign the same vector to a word irrespective of context.
- E.g., The bank is located near the river bank.
- In contrast, context embeddings assign vectors dynamically thereby incorporating rich semantics.

Introduction: motivation for cluster blocking

- Existing solutions capture database interactions via traditional word embeddings.
- That means they assign the same vector to a word irrespective of context.
- E.g., The bank is located near the river bank.
- In contrast, context embeddings assign vectors dynamically thereby incorporating rich semantics.

Introduction: motivation for cluster blocking

- Existing solutions capture database interactions via traditional word embeddings.
- That means they assign the same vector to a word irrespective of context.
- E.g., The bank is located near the river bank.
- In contrast, context embeddings assign vectors dynamically thereby incorporating rich semantics.

Introduction: leveraging contrastive learning for cluster blocking.

- Existing contextual embeddings suffer from anisotropy.

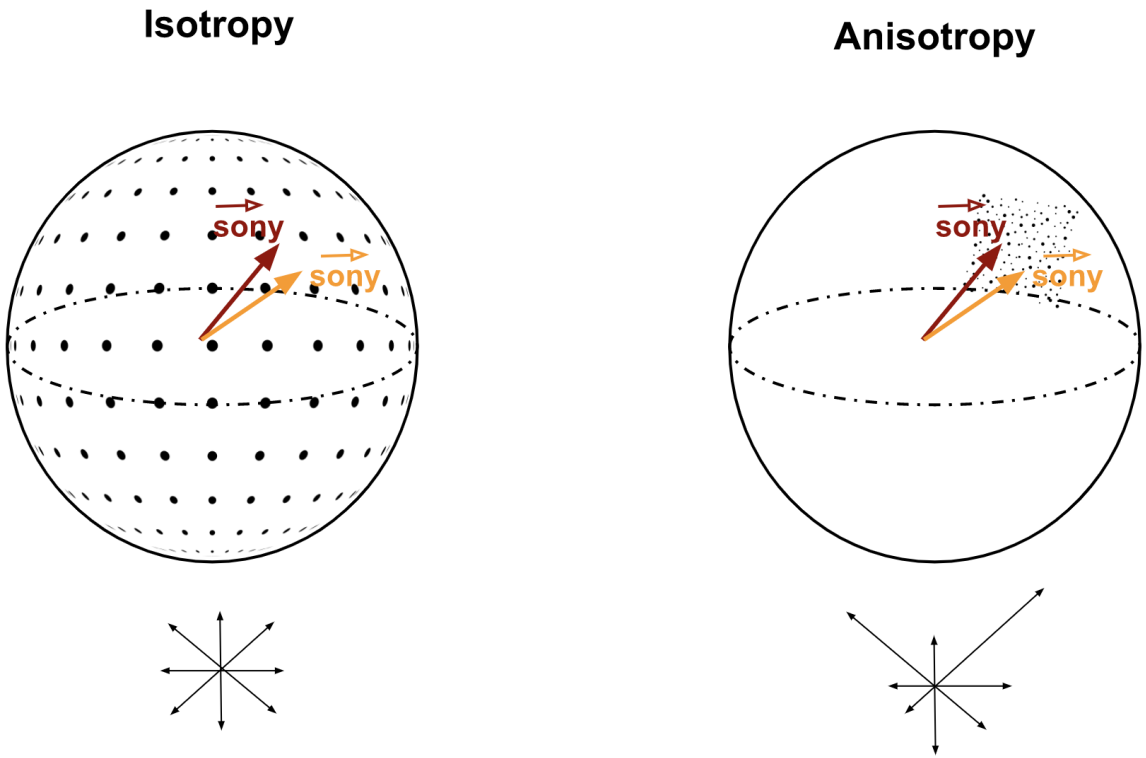


Figure: leveraging contrastive learning.

Problem definition

- Traditional clustering techniques suffer long execution times when dealing with large databases
- As a consequence, improving the efficiency of cluster-based blocking while maintaining accuracy is a major challenge
- To this end, our work exploits pre-trained language models for feature extraction, a k-nearest neighbour graph and graph clustering algorithms
- We wish to execute “blocking” in an effiecient way while maintaining accuracy

Problem definition

- Traditional clustering techniques suffer long execution times when dealing with large databases
- As a consequence, improving the efficiency of cluster-based blocking while maintaining accuracy is a major challenge
- To this end, our work exploits pre-trained language models for feature extraction, a k-nearest neighbour graph and graph clustering algorithms
- We wish to execute “blocking” in an effiecient way while maintaining accuracy

Problem definition

- Traditional clustering techniques suffer long execution times when dealing with large databases
- As a consequence, improving the efficiency of cluster-based blocking while maintaining accuracy is a major challenge
- To this end, our work exploits pre-trained language models for feature extraction, a k-nearest neighbour graph and graph clustering algorithms
- We wish to execute “blocking” in an effiecient way while maintaining accuracy

Problem definition

- Traditional clustering techniques suffer long execution times when dealing with large databases
- As a consequence, improving the efficiency of cluster-based blocking while maintaining accuracy is a major challenge
- To this end, our work exploits pre-trained language models for feature extraction, a k-nearest neighbour graph and graph clustering algorithms
- We wish to execute “blocking” in an effiecient way while maintaining accuracy

Thesis objective and contributions

- We propose a graph-based blocking technique predicated on the k-nearest neighbour (k-NN) graph algorithm for EM.
- We leverage readily available context-aware sentence embeddings from four pre-trained language models for our blocking scheme
- We show that our k-NN graph blocking transcends the existing deep learning-based cluster blocking solution in terms of time and accuracy.

Thesis objective and contributions

- We propose a graph-based blocking technique predicated on the k-nearest neighbour (k-NN) graph algorithm for EM.
- We leverage readily available context-aware sentence embeddings from four pre-trained language models for our blocking scheme
- We show that our k-NN graph blocking transcends the existing deep learning-based cluster blocking solution in terms of time and accuracy.

Thesis objective and contributions

- We propose a graph-based blocking technique predicated on the k-nearest neighbour (k-NN) graph algorithm for EM.
- We leverage readily available context-aware sentence embeddings from four pre-trained language models for our blocking scheme
- We show that our k-NN graph blocking transcends the existing deep learning-based cluster blocking solution in terms of time and accuracy.

Related works

- Earlier attempts adopted rule-based solutions, e.g., standard blocking, sorted neighbourhood blocking, Q-gram blocking, suffix blocking, & canopy blocking
- Normally, a special function (BKV) is used to map tuples to their blocks
  - However, limitations arise when dealing with long, dirty, noisy text or missing values
  - Some methods, e.g., sorted neighbourhood blocking are sensitive to parameters (the sliding window)

Related works

- Earlier attempts adopted rule-based solutions, e.g., standard blocking, sorted neighbourhood blocking, Q-gram blocking, suffix blocking, & canopy blocking
- Normally, a special function (BKV) is used to map tuples to their blocks
  - However, limitations arise when dealing with long, dirty, noisy text or missing values
  - Some methods, e.g., sorted neighbourhood blocking are sensitive to parameters (the sliding window)

Related works

- Earlier attempts adopted rule-based solutions, e.g., standard blocking, sorted neighbourhood blocking, Q-gram blocking, suffix blocking, & canopy blocking
- Normally, a special function (BKV) is used to map tuples to their blocks
  - However, limitations arise when dealing with long, dirty, noisy text or missing values
  - Some methods, e.g., sorted neighbourhood blocking are sensitive to parameters (the sliding window)

Related works

- Earlier attempts adopted rule-based solutions, e.g., standard blocking, sorted neighbourhood blocking, Q-gram blocking, suffix blocking, & canopy blocking
- Normally, a special function (BKV) is used to map tuples to their blocks
  - However, limitations arise when dealing with long, dirty, noisy text or missing values
  - Some methods, e.g., sorted neighbourhood blocking are sensitive to parameters (the sliding window)

Related works

- Later the paper of Azzalini[1] develops a system for "blocking" based on the RNN architecture.
  - However, clustering large data sets proves to be resource-intensive
  - Morever, vectors have to be down-sampled via the t-SNE algorithm, in their work, which scales poorly on big data sets
  - The RNN architecture relies on simple word embeddings that neglect context

[1] F Azzalini, et al. 2020. Blocking Techniques for Entity Linkage: A Semantics-Based Approach.

Related works

- Later the paper of Azzalini[1] develops a system for "blocking" based on the RNN architecture.
  - However, clustering large data sets proves to be resource-intensive
  - Morever, vectors have to be down-sampled via the t-SNE algorithm, in their work, which scales poorly on big data sets
  - The RNN architecture relies on simple word embeddings that neglect context

[1] F Azzalini, et al. 2020. Blocking Techniques for Entity Linkage: A Semantics-Based Approach.

Related works

- Later the paper of Azzalini[1] develops a system for "blocking" based on the RNN architecture.
  - However, clustering large data sets proves to be resource-intensive
  - Morever, vectors have to be down-sampled via the t-SNE algorithm, in their work, which scales poorly on big data sets
  - The RNN architecture relies on simple word embeddings that neglect context

[1] F Azzalini, et al. 2020. Blocking Techniques for Entity Linkage: A Semantics-Based Approach.

Related works

- Later the paper of Azzalini[1] develops a system for "blocking" based on the RNN architecture.
  - However, clustering large data sets proves to be resource-intensive
  - Morever, vectors have to be down-sampled via the t-SNE algorithm, in their work, which scales poorly on big data sets
  - The RNN architecture relies on simple word embeddings that neglect context

[1] F Azzalini, et al. 2020. Blocking Techniques for Entity Linkage: A Semantics-Based Approach.

Proposed approach: system overview

An overview of the system is as follows;

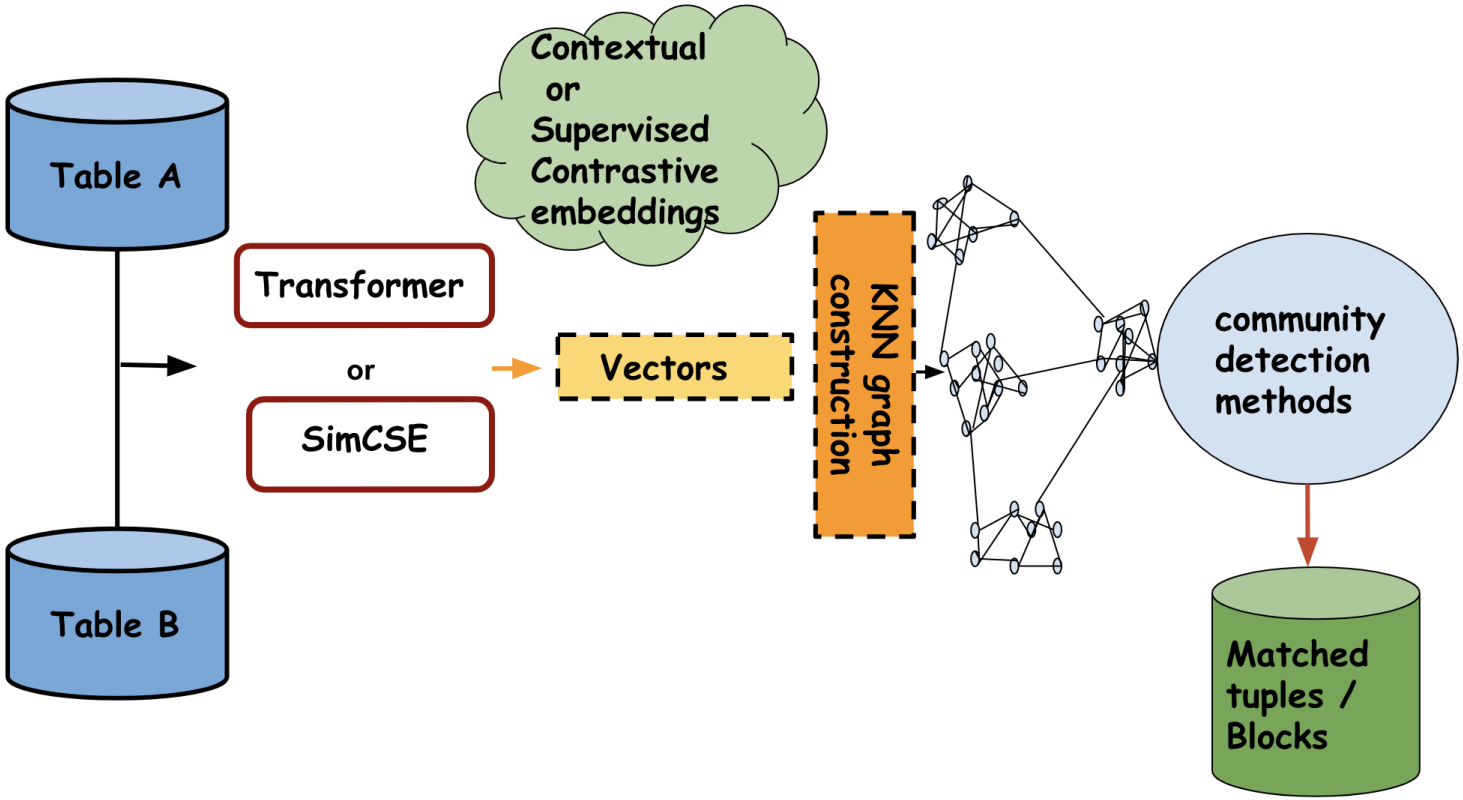


Figure: Our blocking system

Proposed approach: pipeline step 1

First, attributes of data sets to be integrated are concatenated into a string

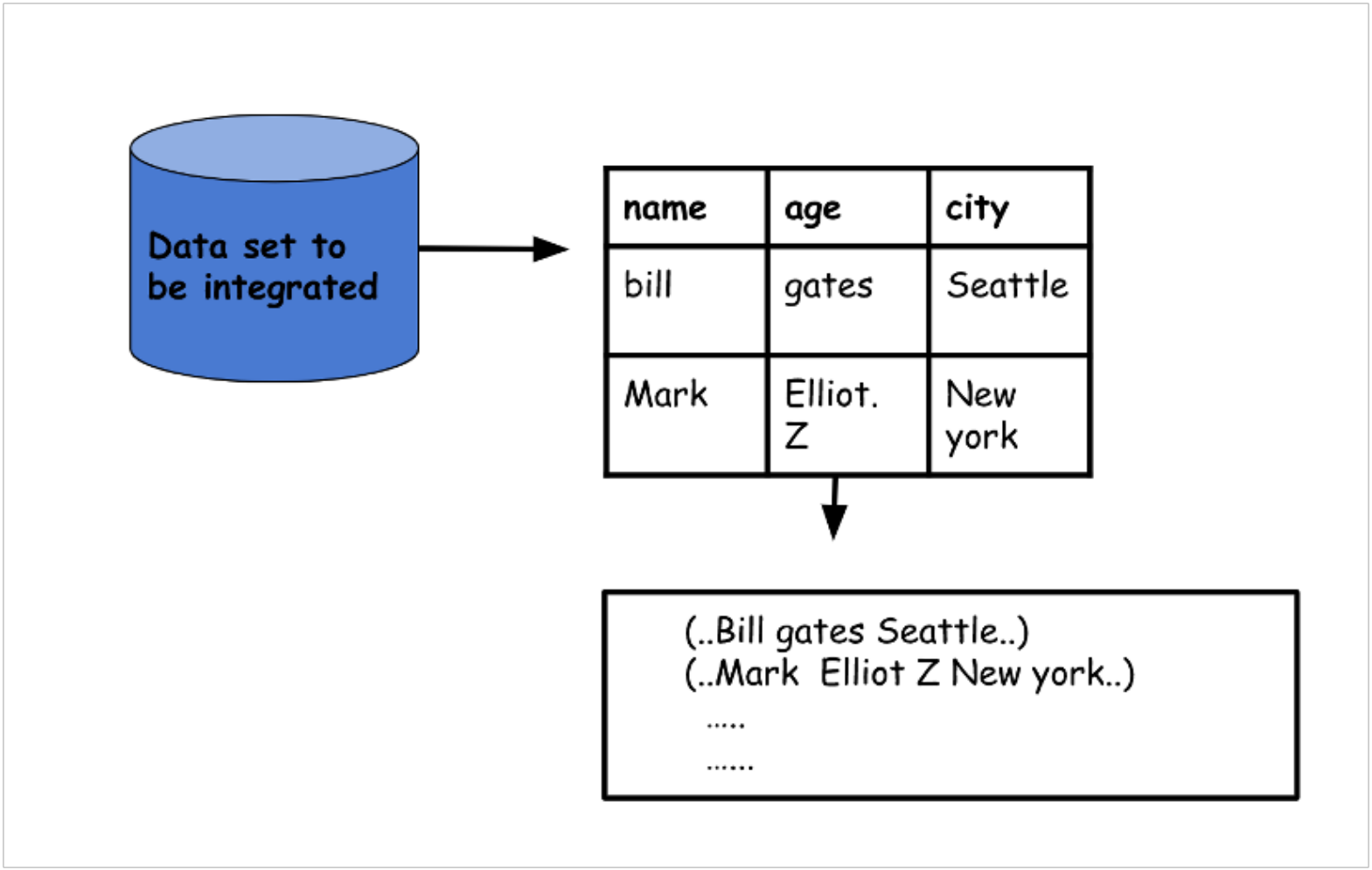


Figure: Textual representation from table A or B

Proposed approach:pipeline step 2

Next, each tuple is then input to a pre-trained transformer language model producing context embeddings

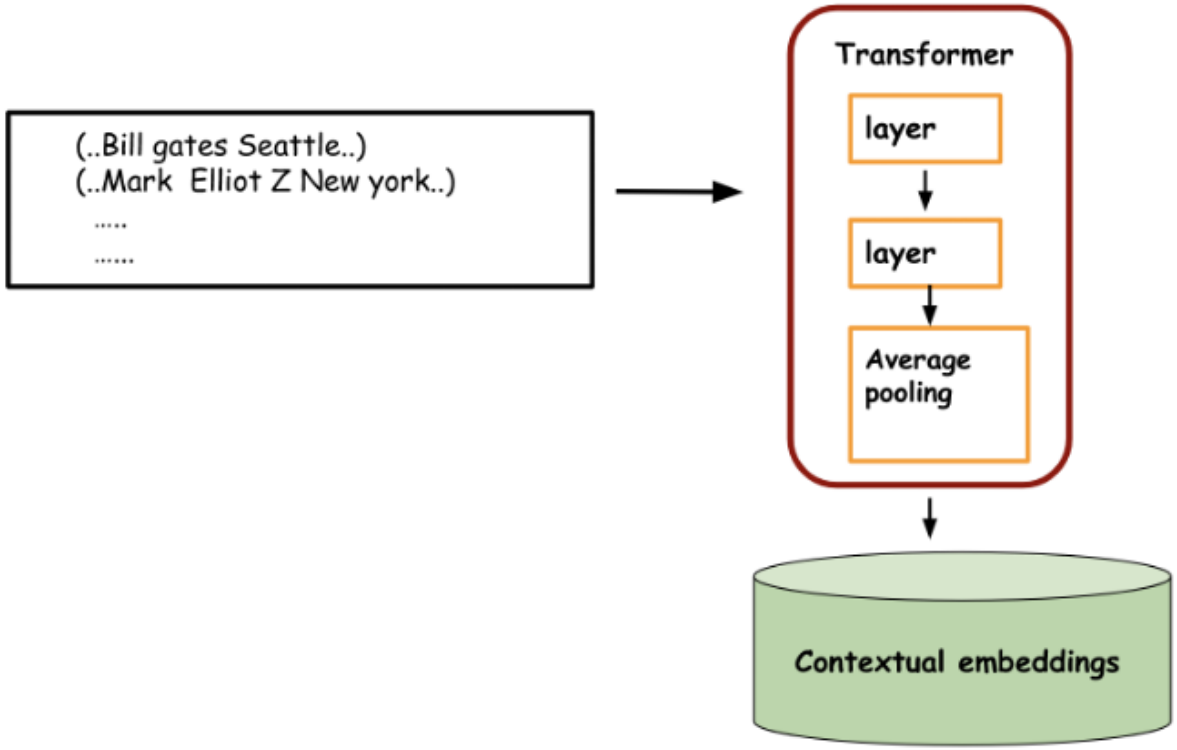


Figure: Feature extraction (generating embeddings)

Proposed approach:pipeline step 3

Projection of embeddings to lower dimension is possible via UMAP or CVAE

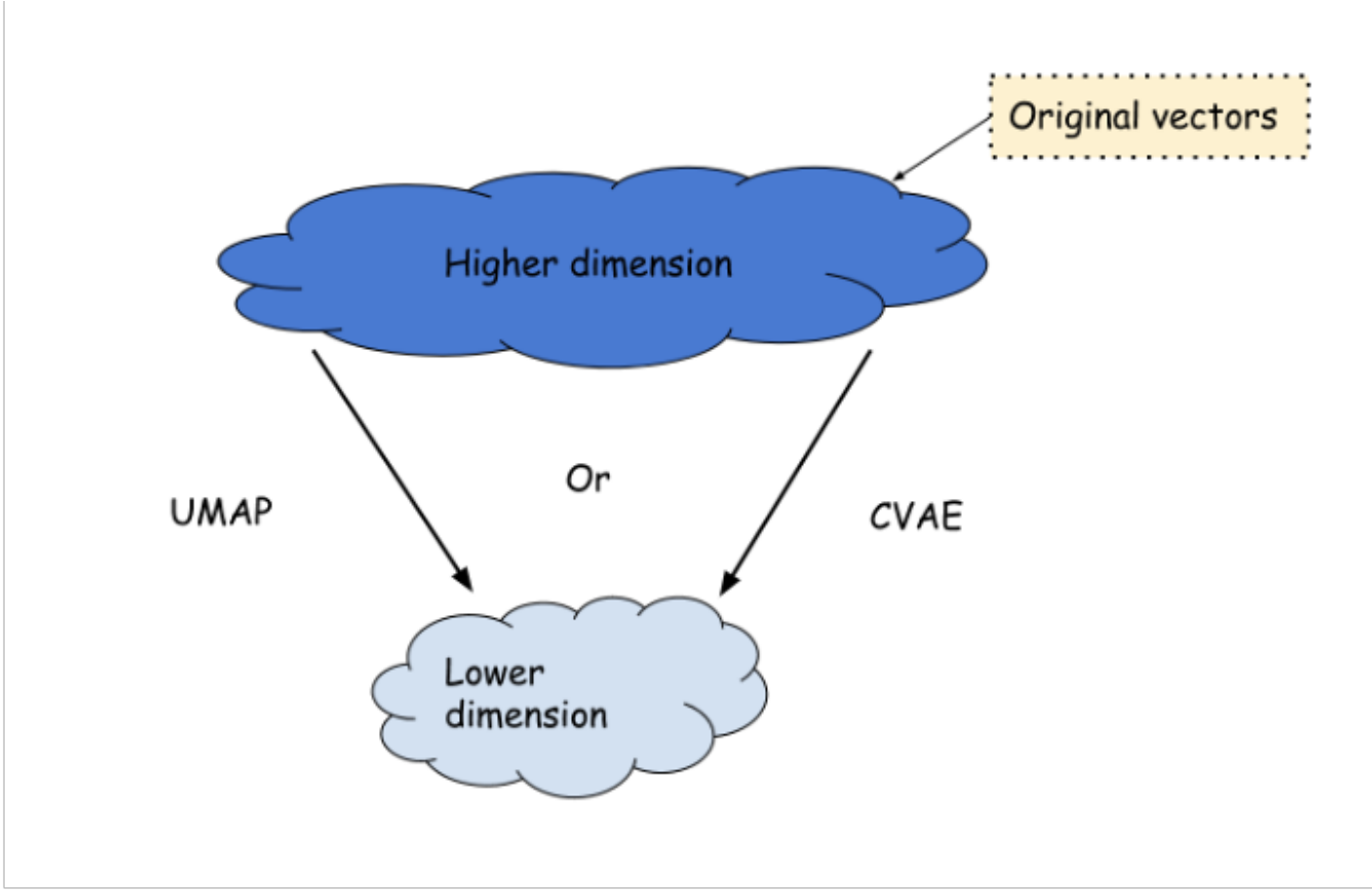


Figure: elaborating the vector processing in case of dimensionality reduction

Proposed approach: pipeline step 4

Next, we apply knn graph algorithm on embedding vectors to construct a graph followed by unsupervised community detection algorithms

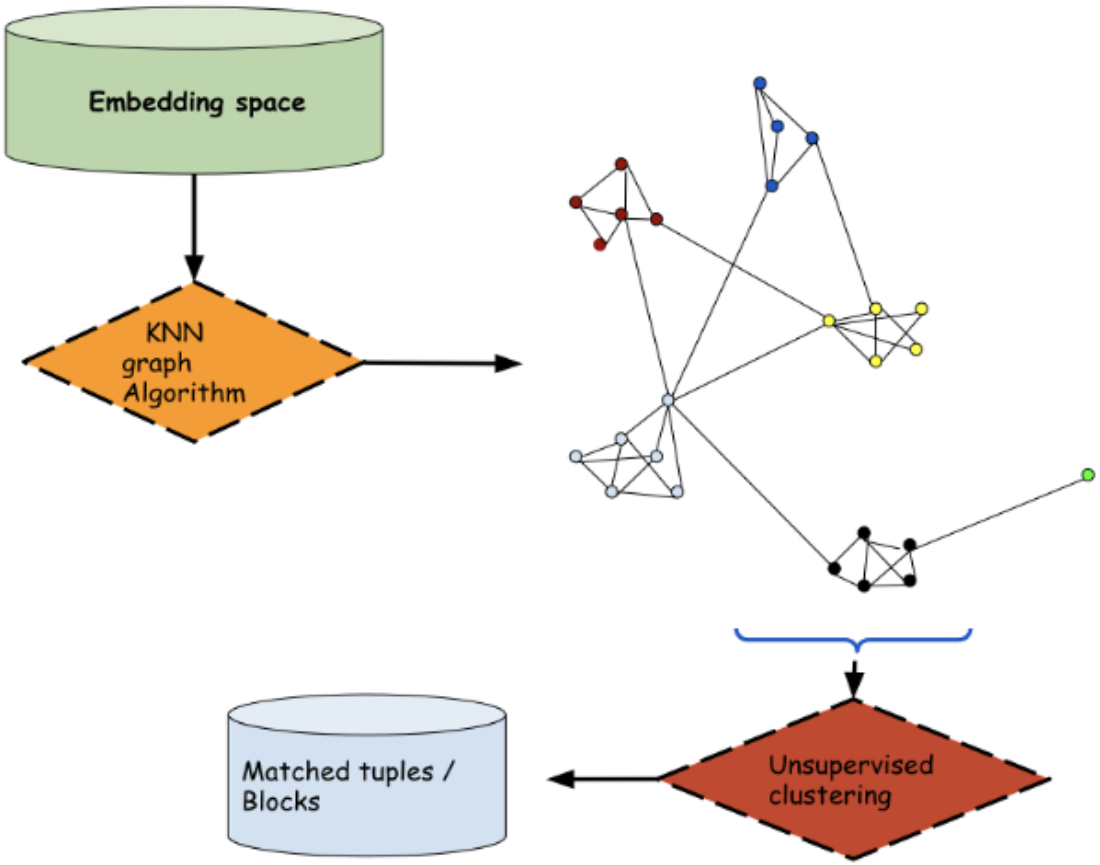


Figure: KNN-graph based blocking

Experimental work: data sets

- Each data set has the format of **Table A-Table B**
- Each pair has more than 6 million record comparisons

Table 5: Dataset statistics.

Data	Domain	#Tuples	#Matches	Attr	Size (M)
DBLP-Scholar,	citation	2616-64263	5347	4	168
iTunes-Amazon	music	6907-55923	132	8	386
Walmart-Amazon	electronics	2554-22074	962	5	56
GoogleScholar-DBLP	citation	2616-64263	5347	4	168

Figure: Experimental datasets for entity matching

Experimental work: data sets

- Each data set has the format of **Table A-Table B**
- Each pair has more than 6 million record comparisons

Table 5: Dataset statistics.

Data	Domain	#Tuples	#Matches	Attr	Size (M)
DBLP-Scholar,	citation	2616-64263	5347	4	168
iTunes-Amazon	music	6907-55923	132	8	386
Walmart-Amazon	electronics	2554-22074	962	5	56
GoogleScholar-DBLP	citation	2616-64263	5347	4	168

Figure: Experimental datasets for entity matching

Experimental work: computing environment & key parameters

- For the transformer based models, we choose the attention spans to be 200 tokens
- Batch size is chosen to be 32 & mean-pooling for summarising input tokens
- A single workstation equipped with Intel(R) Core(TM) i7-4820K quad-core CPU encompassing 48 GB RAM running Ubuntu 18.04
- We use pre-trained models based on Hugging-face [2] & all programs are executed in python version 3.7.6

[2] T. Wolf et al. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:cs.CL/1910.03771

Experimental work: computing environment & key parameters

- For the transformer based models, we choose the attention spans to be 200 tokens
- Batch size is chosen to be 32 & mean-pooling for summarising input tokens
- A single workstation equipped with Intel(R) Core(TM) i7-4820K quad-core CPU encompassing 48 GB RAM running Ubuntu 18.04
- We use pre-trained models based on Hugging-face [2] & all programs are executed in python version 3.7.6

[2] T. Wolf et al. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:cs.CL/1910.03771

Experimental work: computing environment & key parameters

- For the transformer based models, we choose the attention spans to be 200 tokens
- Batch size is chosen to be 32 & mean-pooling for summarising input tokens
- A single workstation equipped with Intel(R) Core(TM) i7-4820K quad-core CPU encompassing 48 GB RAM running Ubuntu 18.04
- We use pre-trained models based on Hugging-face [2] & all programs are executed in python version 3.7.6

[2] T. Wolf et al. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:cs.CL/1910.03771

Experimental work: computing environment & key parameters

- For the transformer based models, we choose the attention spans to be 200 tokens
- Batch size is chosen to be 32 & mean-pooling for summarising input tokens
- A single workstation equipped with Intel(R) Core(TM) i7-4820K quad-core CPU encompassing 48 GB RAM running Ubuntu 18.04
- We use pre-trained models based on Hugging-face [2] & all programs are executed in python version 3.7.6

[2] T. Wolf et al. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:cs.CL/1910.03771

Results: blocking time

Table 5: iTunes-Amazon.

method	algo_{best}	emb'_{sec}	bk_{sec}	total_{sec}	F1
R-BERT	l'vian	**91.8**	461.8	553.6	85.2
DeBERTa	l'vian	311.2	557.2	868.4	89.2
RoBERTa	l'vian	253.6	**58.1**	**311.7**	89.7
BART	l'vian	324.0	433.6	757.6	**91.7**
RNN	birch	2329.8	dnf	dnf	dnf
SimCSE	l'vian	64.5	160.9	225.4	92.8
R-BERT_d	l'den	127.7	**328.2**	**455.9**	56.2
DeBERTa_d	l'den	470.0	607.8	1077.8	56.4
RoBERTa_d	l'vian	391.5	368.2	759.7	64.0
BART_d	l'den	642.9	347.5	990.4	**68.0**
SimCSE_d	l'den	125.8	164.4	290.2	89.7

Figure: Performance on iTunes-Amazon(62,830 tuples)

Results:blocking time

Table 4: GoogleScholar-DBLP-1.

method	algo_{best}	emb'_{sec}	bk_{sec}	total_{sec}	F1
R-BERT	l'vian	111.4	203.8	315.2	**93.5**
DeBERTa	l'vian	439.0	215.3	654.3	89.5
RoBERTa	l'vian	370.9	226.2	597.0	90.7
BART	l'vian	451.2	189.0	640.2	92.4
RNN	birch	2563.0	dnf	dnf	dnf
SimCSE	l'vian	110.4	156.8	267.2	97.8
R-BERT_d	l'vian	131.6	210.3	342.0	**86.8**
DeBERTa_d	l'vian	463.2	211.0	674.2	80.2
RoBERTa_d	l'vain	380.6	463.7	844.3	73.8
BART_d	l'vian	501.0	194.5	695.4	83.5
SimCSE_d	l'vian	90.8	245.5	336.3	93.8

Note: R-BERT is a short form for ReviewBERT, l'vian for louvian and l'den for leiden.

Figure: Performance on DBLP-Scholar(66,879 tuples)

Results:blocking time

Table 7: GoogleScholar-DBLP-2.

method	algo_{best}	emb'_{sec}	bk_{sec}	total_{sec}	F1
R-BERT	l'vian	**108.0**	283.0	**391.0**	**91.6**
DeBERTa	l'vian	293.0	283.3	576.3	89.1
RoBERTa	l'vian	328.6	**229.8**	558.4	89.6
BART	l'vian	289.8	261.4	551.2	91.4
RNN	birch	2787.1	dnf	dnf	dnf
SimCSE	Louvain	127.8	173.4	301.2	95.6

Figure: Performance on GoogleScholar-DBLP(66,879 tuples)

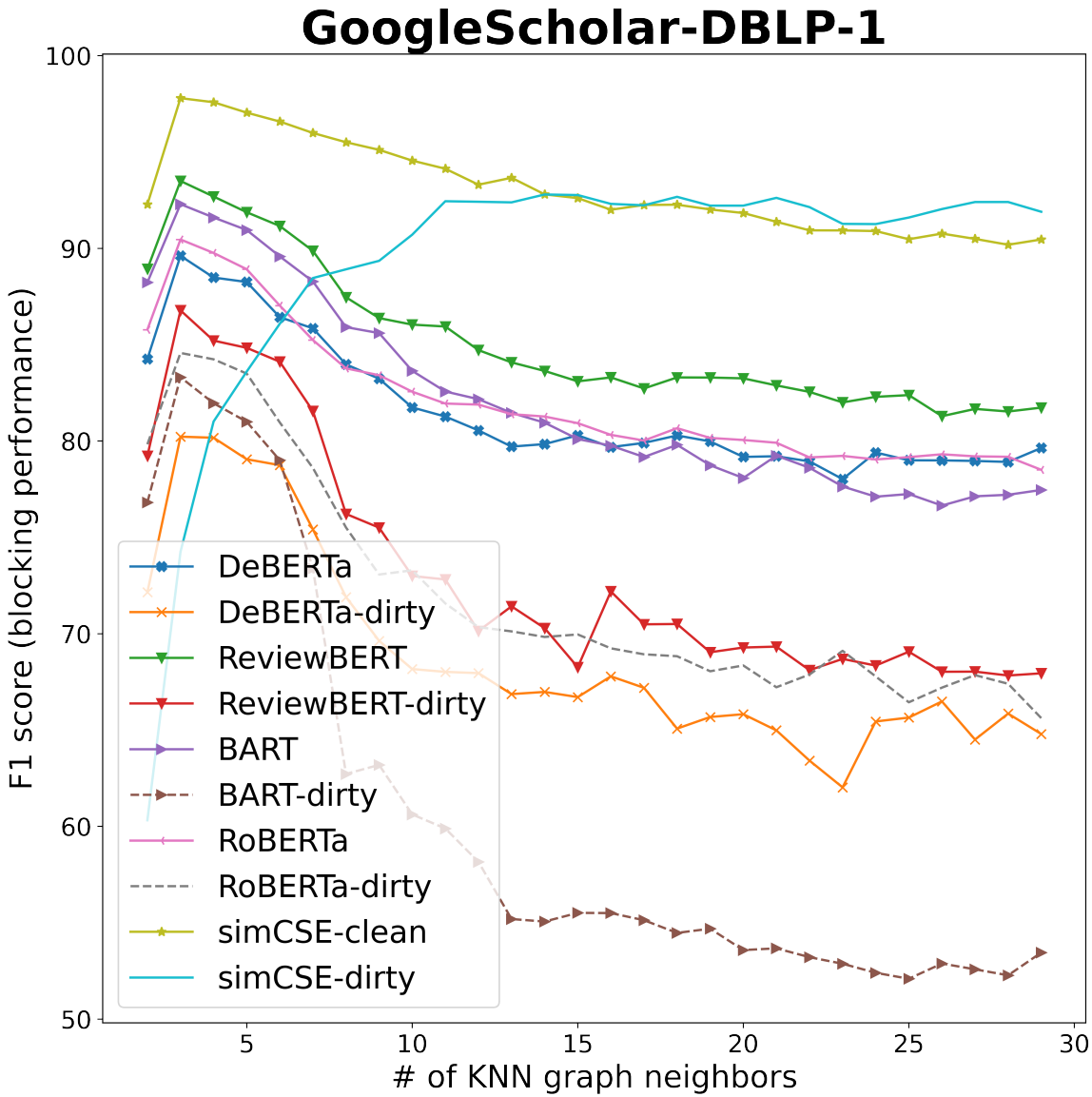
Results:blocking time

Table 6: Walmart-Amazon.

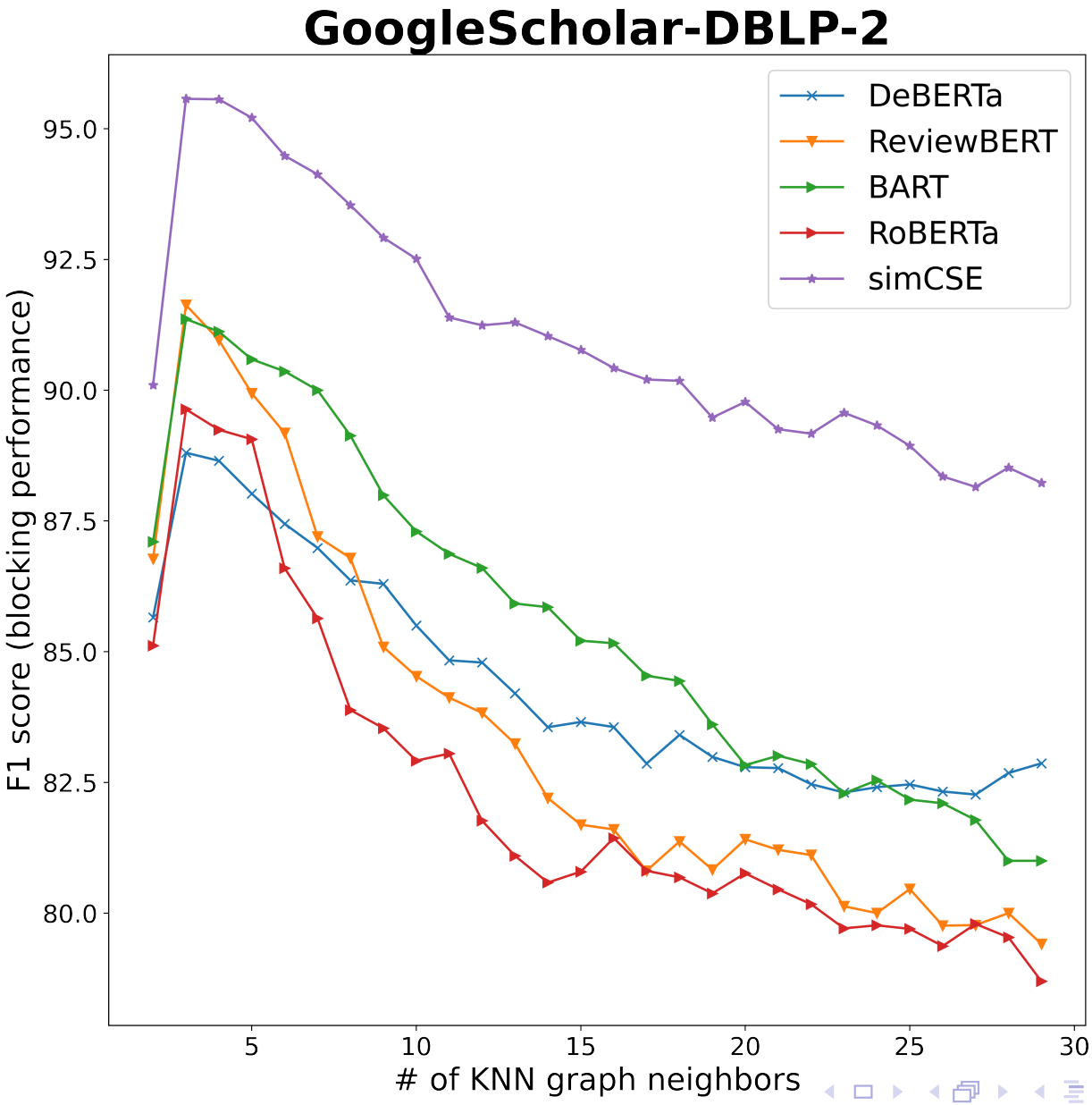
method	algo_{best}	emb'_{sec}	bk_{sec}	total_{sec}	F1
R-BERT	l'vian	**39.1**	58.9	**98.0**	**91.6**
DeBERTa	l'den	134.5	47.3	181.9	90.1
RoBERTa	l'vian	111.3	58.2	168.6	89.7
BART	l'vian	132.1	48s	180.1	90.2
RNN	birch	835.9	**12.6**	848.4	90.1
SimCSE	l'vian	42.1	36.46	78.5	92.5
R-BERT_d	l'vian	**52.9**	45.6	**98.51**	90.3
DeBERTa_d	l'vian	162.2	48.08	210.2	**90.5**
RoBERTa_d	l'den	283.2	56.28	339.4	87.6
BART_d	l'vian	160.5	**30.4**	190.9	89.4
SimCSE_d	l'vian	47.3	32.98	80.3	92.5

Figure: Performance on Walmart-Amazon(22,628 tuples)

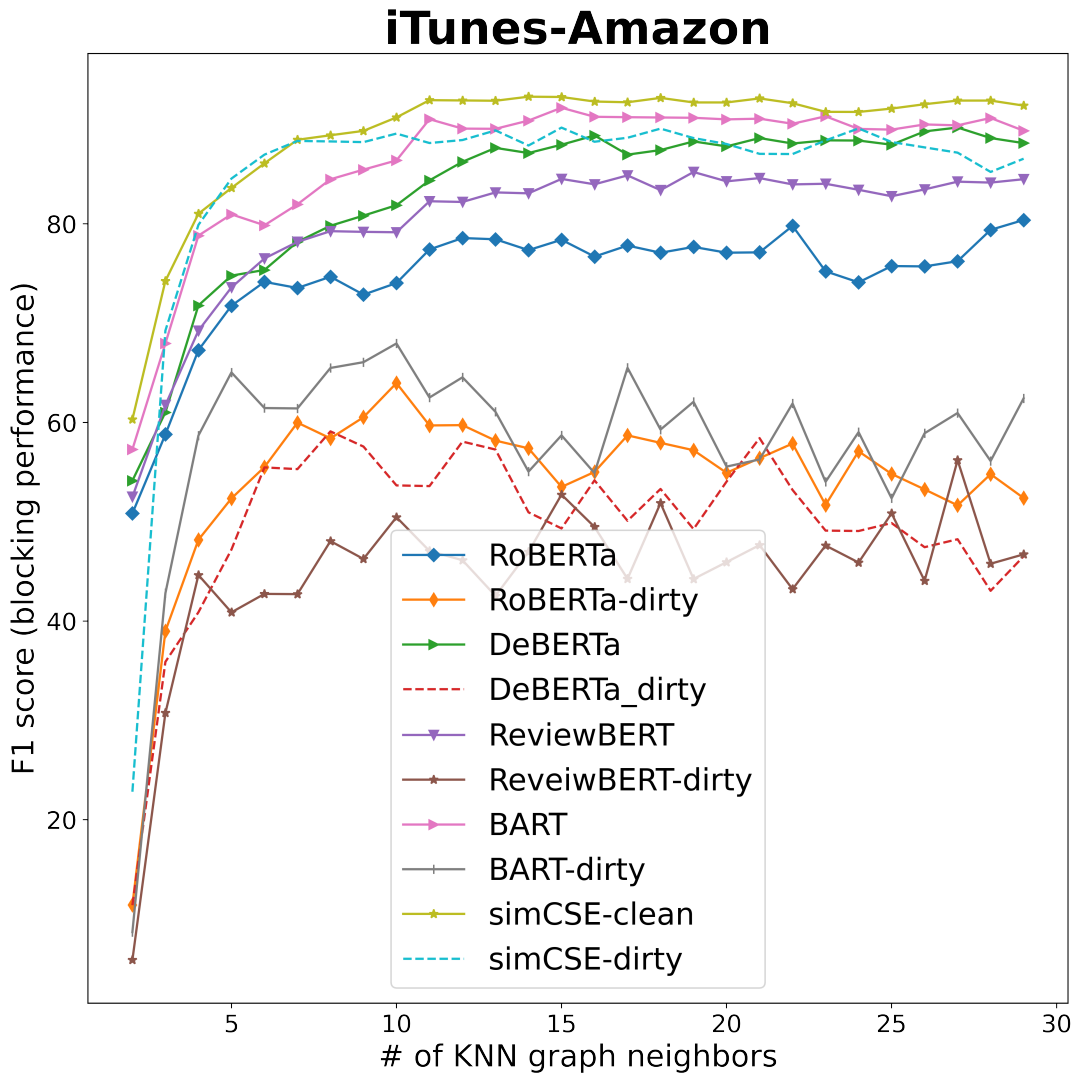
Comparison of embeddings as a function of parameter k



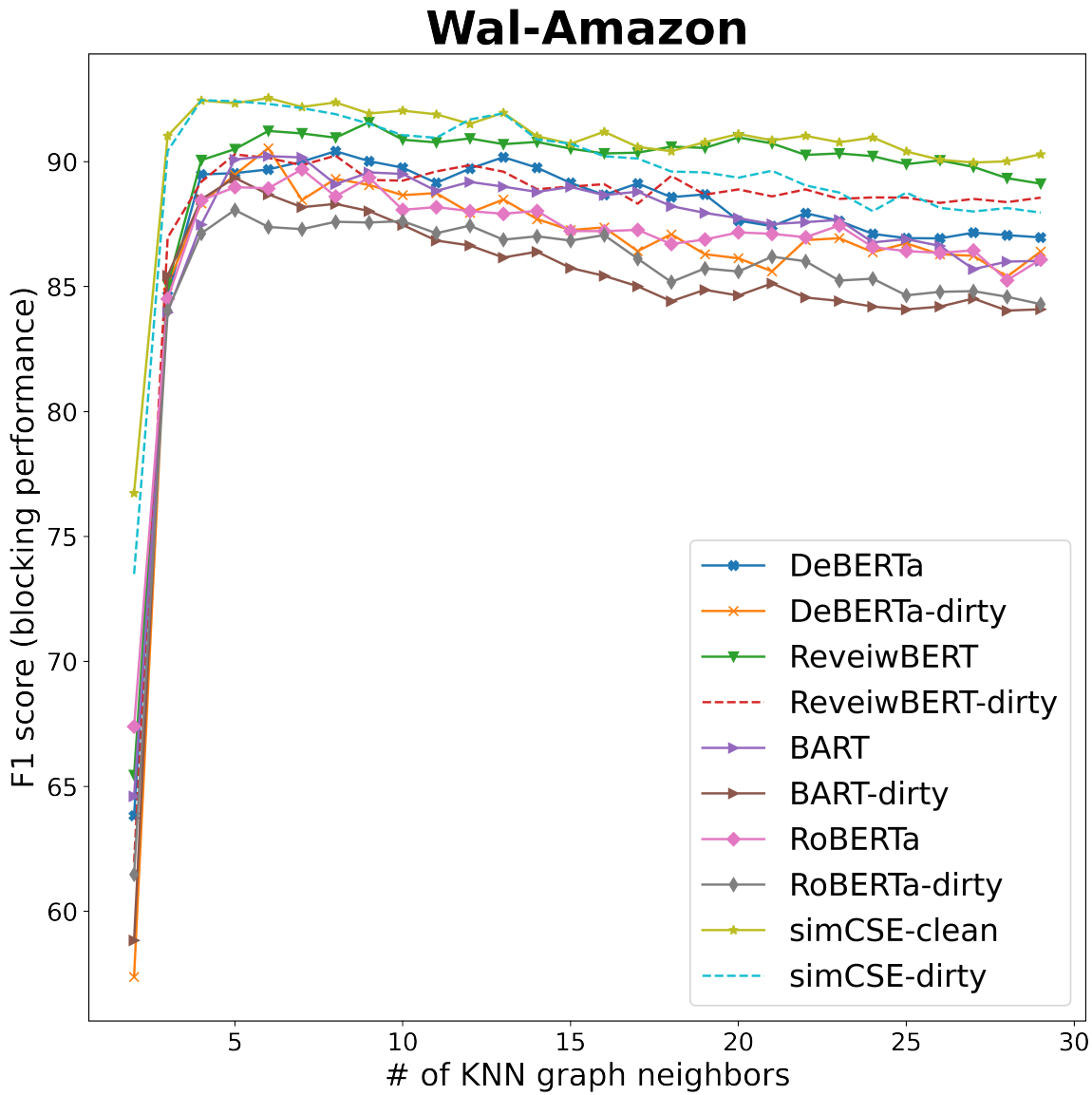
Comparison of embeddings as a function of parameter k



Comparison of embeddings as a function of parameter k



Comparison of embeddings as a function of parameter k



Conclusion

- As future work, we plan to improve representation learning using task domain data as well combining our approach with a supervised system for Entity Matching.

The End