Topic Modeling

Anna, Christian, Freya, Luca, Micaela, Saurabh, Sophia, Subir

Agenda

1. Introduction
2. Approach
3. Pipeline
   a. Sentence Vectorization
   b. Dimensionality Reduction
   c. Clustering
   d. Grid Search
   e. Topic Labeling
4. Results
5. Next Steps / Discussion

Introduction

- Dataset
    - Employee survey data.
    - Short comments.

Introduction

- Dataset
  - Employee survey data.
  - Short comments.

THE CUSTOMER IS THE CENTER.
we are usually seeking customer satisfaction
The CIF training is really a good chance for everyone
Training programs are good for team building and product knowledge (ie CIF program).
PLS CONTINUE THE ZUMBA SESSION.
[company] is a good company

Introduction

- Dataset
  - Employee survey data.
  - Short comments.
- Task
  - Report on topics in the dataset.

THE CUSTOMER IS THE CENTER.
we are usually seeking customer satisfaction
The CIF training is really a good chance for everyone
Training programs are good for team building and product knowledge (ie CIF program).
PLS CONTINUE THE ZUMBA SESSION.
[company] is a good company

Introduction

- Dataset
  - Employee survey data.
  - Short comments.
- Task
  - Report on topics in the dataset.

THE CUSTOMER IS THE CENTER.	Customer Service
we are usually seeking customer satisfaction	Customer Service
The CIF training is really a good chance for everyone	Training
Training programs are good for team building and product knowledge (ie CIF program).	Training
PLS CONTINUE THE ZUMBA SESSION.	Amenities?
[company] is a good company	?

Introduction

- Dataset
  - Employee survey data.
  - Short comments.
- Task
  - Report on topics in the dataset.
- Approaches
  - Supervised Topic Classification
    - Can't dynamically identify new topics
  - Unsupervised

THE CUSTOMER IS THE CENTER.	Customer Service
we are usually seeking customer satisfaction	Customer Service
The CIF training is really a good chance for everyone	Training
Training programs are good for team building and product knowledge (ie CIF program).	Training
PLS CONTINUE THE ZUMBA SESSION.	Amenities?
[company] is a good company	?

Introduction

- Dataset
  - Employee survey data.
  - Short comments.
- Task
  - Report on topics in the dataset.
- Approaches
  - Supervised Topic Classification
    - Can't dynamically identify new topics
  - Unsupervised
- Problem Statement
  - Use unsupervised learning to report on topics in the dataset.

THE CUSTOMER IS THE CENTER.	Customer Service
we are usually seeking customer satisfaction	Customer Service
The CIF training is really a good chance for everyone	Training
Training programs are good for team building and product knowledge (ie CIF program).	Training
PLS CONTINUE THE ZUMBA SESSION.	Amenities?
[company] is a good company	?

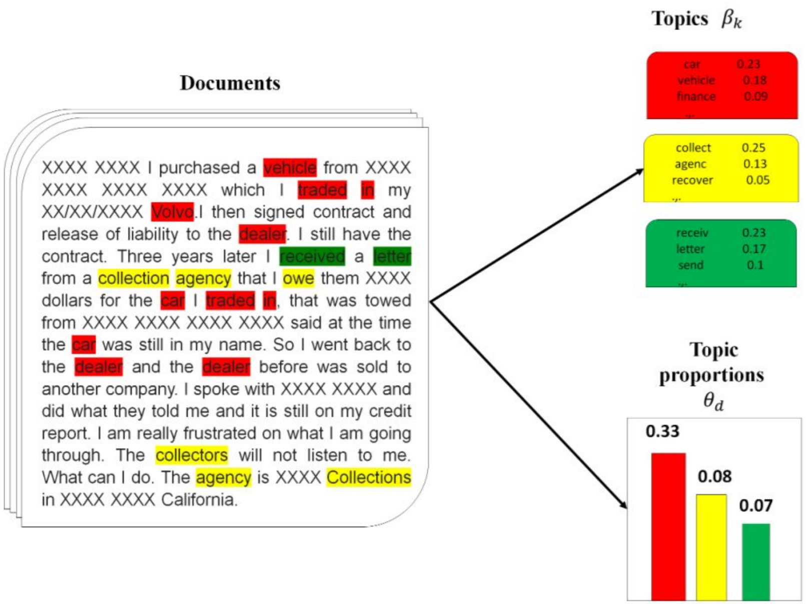
Statistical Topic Model Approach

Statistical Topic Model Approach

- Latent Dirichlet Allocation (LDA)[1]
- Main Idea
  - Assume a generative model and infer a set of 'latent' topics which could have produced a set of documents.

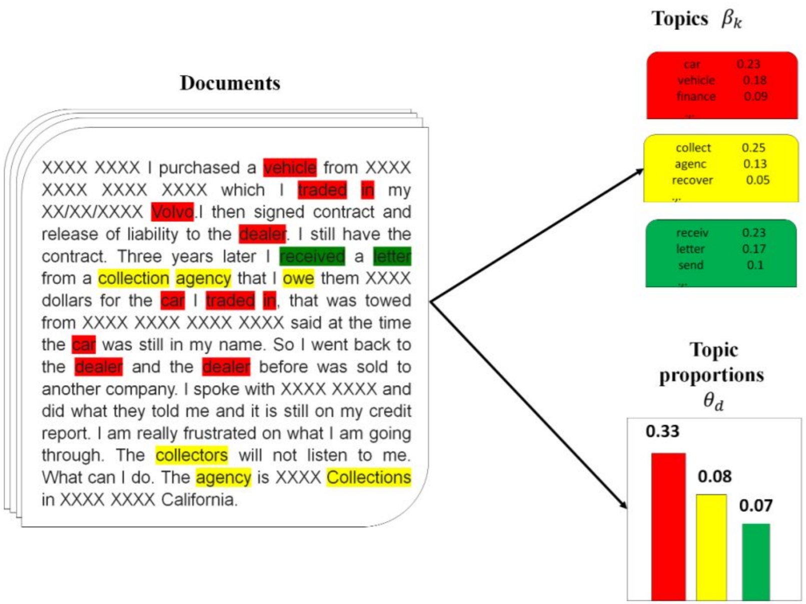
Statistical Topic Model Approach

- Latent Dirichlet Allocation (LDA)[1]
- Main Idea
  - Assume a generative model and infer a set of 'latent' topics which could have produced a set of documents.



Statistical Topic Model Approach

- Latent Dirichlet Allocation (LDA)[1]
- Main Idea
  - Assume a generative model and infer a set of 'latent' topics which could have produced a set of documents.
- Advantages
  - Language independent.
- Disadvantages
  - Hyperparameter tuning.
- Usage
  - gensim.models.ldamodel[2]



Topic Modeling

Word Length

Total words : 343549

After 10 common word remove : 275579

After all preprocessing steps : 173403

Top 10 common word

====================

to, the, and, is, of, in, a, for, are, be

Gensim LDA Model

```
# Build LDA model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word, # Dictionary
                                            chunksize=100, # Number of documents used in each training chunk
                                            alpha='auto',
                                            eta='auto',
                                            iterations=400, # Maximum number of iterations through the corpus
                                            num_topics=8,
                                            passes=passes, # Number of passes through the corpus during training
                                            eval_every=1)
```

Topic coherence measure : u_mass

Coherence Score: -6.830808283945607 where 0 is best and -14 is worst

Topic List

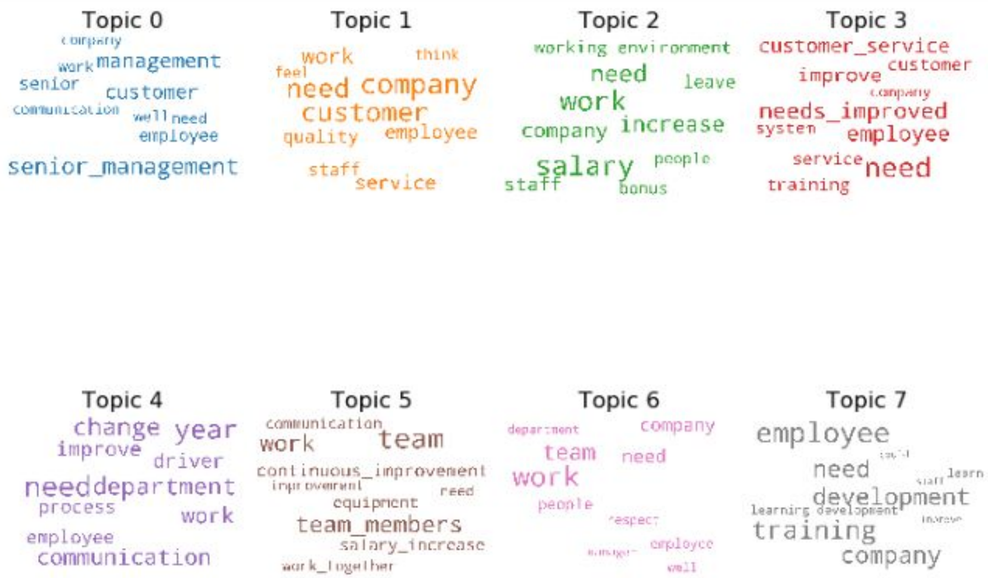


Fig: All of the topic list

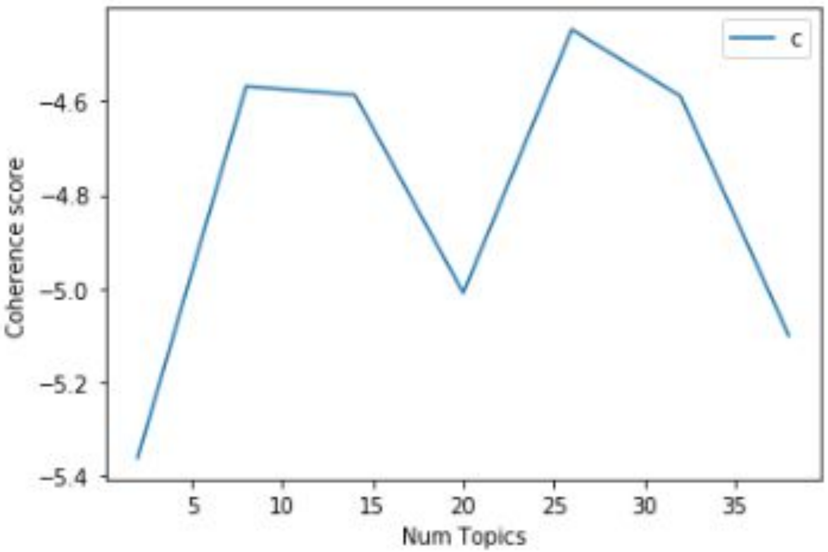


Fig : Best Coherence Score -4.5681 for number of topics 8

Assign Topic

	Row_No	Dominant_Topic	Keywords	comments
0	0	0.0	senior_management, management, customer, senio...	[customer, need, communicate, aperiodically]
1	1	6.0	work, team, need, company, people, employee, r...	[custom, business, development, continue, grow...
2	2	5.0	team, team_members, work, continuous_improveme...	[think, team, work, hard, commit, continuous, ...
3	3	0.0	senior_management, management, customer, senio...	[overall, work, towards, customer, centric, en...
4	4	1.0	company, customer, need, work, service, employ...	[customer, centricity, grow, culture, company,...
5	5	5.0	team, team_members, work, continuous_improveme...	[develop, comfortable, rapport, client, determ...
6	6	0.0	senior_management, management, customer, senio...	[customer, center]
7	7	0.0	senior_management, management, customer, senio...	[usually, seek, customer, satisfaction, help, ...
8	8	5.0	team, team_members, work, continuous_improveme...	[alignment, regional, office, country, focus, ...
9	9	0.0	senior_management, management, customer, senio...	[innovation, customer, relation, ship, custome...

Drawback :

- Some words are repeated in multiple topic.
- Display different topic words upon running LDA model multiple times.

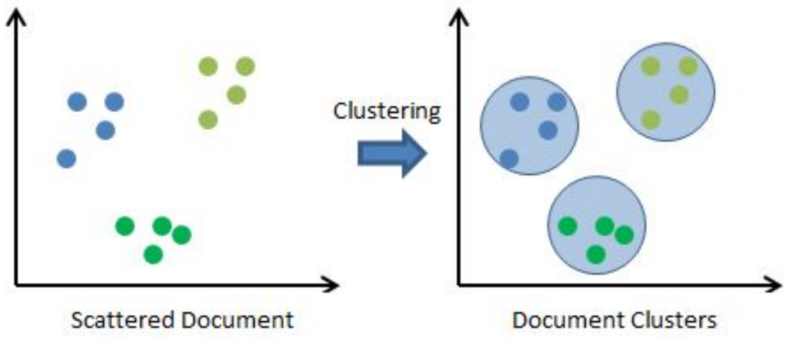
Approach: Document Clustering

Approach: Document Clustering

- Main Idea
  - Vectorize each document and perform a cluster analysis.

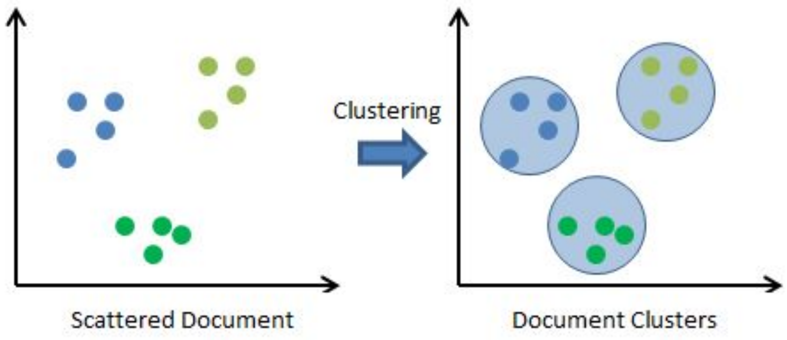
Approach: Document Clustering

- Main Idea
  - Vectorize each document and perform a cluster analysis.



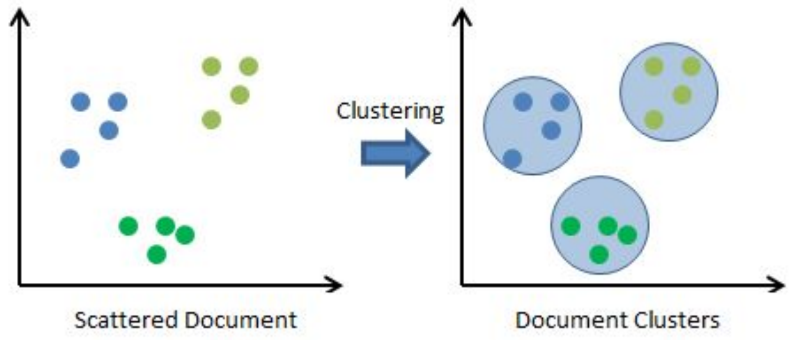
Approach: Document Clustering

- Main Idea
  - Vectorize each document and perform a cluster analysis.
- Justification
  - Explore word embeddings.
  - Hypothesis: Easier to employ transfer learning and compensate for lack of information in short texts.



Approach: Document Clustering

- Main Idea
  - Vectorize each document and perform a cluster analysis.
- Justification
  - Explore word embeddings.
  - Hypothesis: Easier to employ transfer learning and compensate for lack of information in short texts.
- Challenges
  - Calculating effective sentence vectors.
  - Choice of cluster algorithm.
  - Hyperparameter optimization.



Approach: Overview

Approach: Overview

1. Word vectorization
2. Sentence vectorization
3. Dimensionality reduction
4. Cluster analysis
   a. Cluster algorithm
   b. Hyperparameter optimization
5. Topic labeling and representative sentences

Approach: Overview

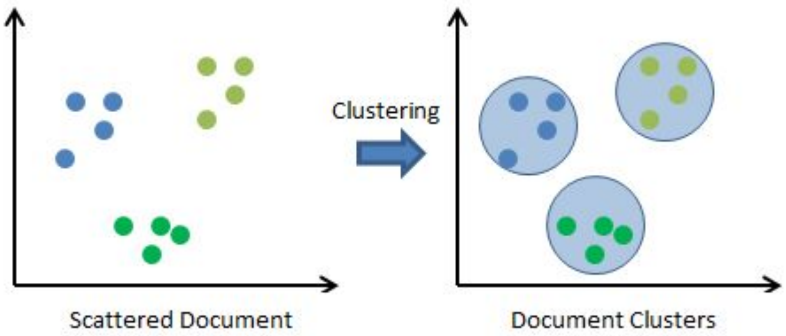
1. Word vectorization
2. Sentence vectorization
3. Dimensionality reduction
4. Cluster analysis
   a. Cluster algorithm
   b. Hyperparameter optimization
5. Topic labeling and representative sentences

6. Evaluation

Approach: Overview

1. Word vectorization
2. Sentence vectorization
3. Dimensionality reduction
4. Cluster analysis
   a. Cluster algorithm
   b. Hyperparameter optimization
5. Topic labeling and representative sentences

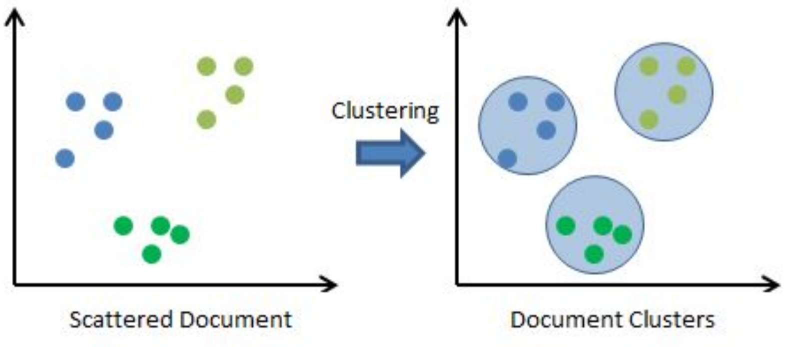
6. Evaluation

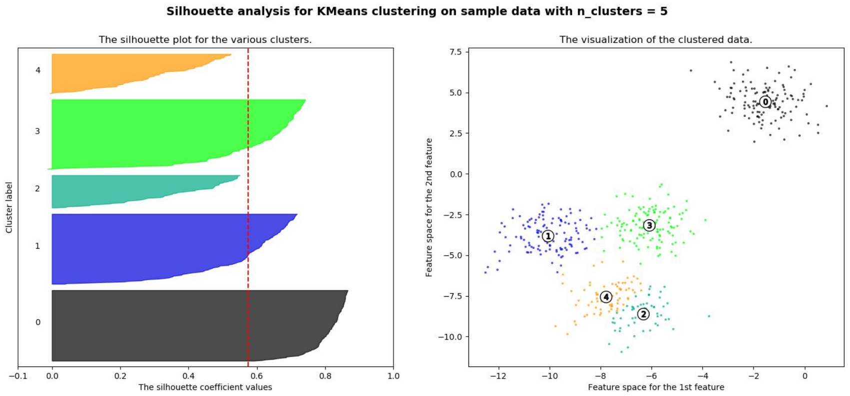


Approach: Overview

1. Word vectorization
2. Sentence vectorization
3. Dimensionality reduction
4. Cluster analysis
    a. Cluster algorithm
    b. Hyperparameter optimization
5. Topic labeling and representative sentences

6. Evaluation





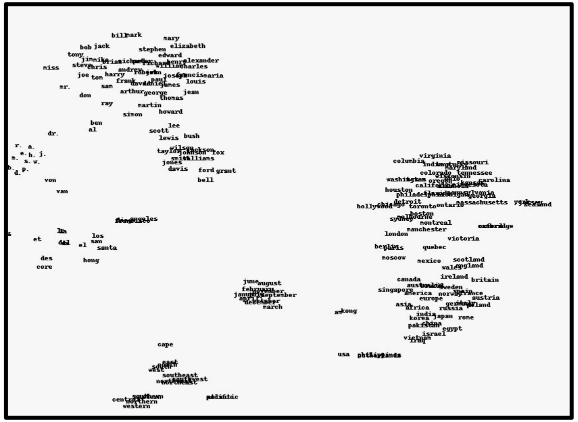
Sentence Vectorization

Sentence Vectorization

- Word Embeddings
  - FastText[5] - Word vectors with subword info
  - Bert[6] - Contextualized vectors with attention

Sentence Vectorization

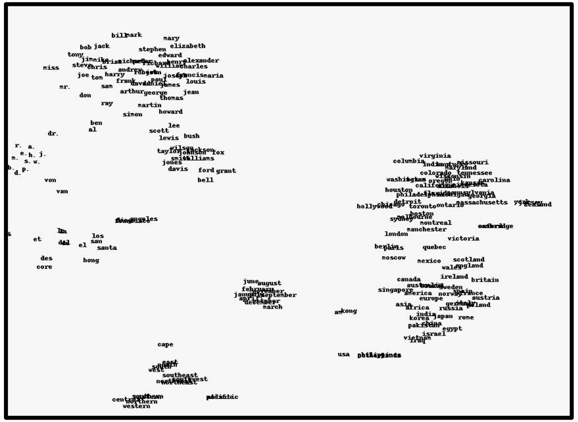
- Word Embeddings
  - FastText[5] - Word vectors with subword info
  - Bert[6] - Contextualized vectors with attention



Ideal

Sentence Vectorization

- Sentence Embeddings
  - Simple average.
  - Smooth inverse frequency weighted average.
    - Arora et al (2016)[8]



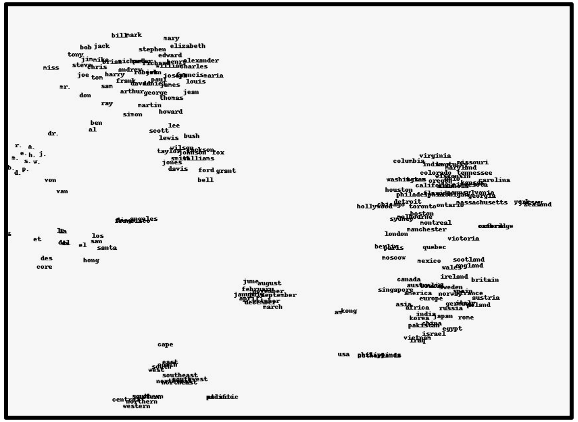
Ideal

Sentence Vectorization

- Sentence Embeddings
  - Simple average.
  - Smooth inverse frequency weighted average.
    - Arora et al (2016)[8]

SIF = a/(a + p(w))

a = 1e-5
p(w) = word frequency



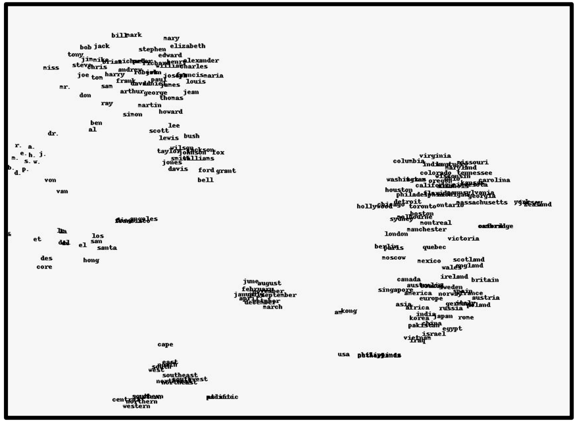
Ideal

Sentence Vectorization

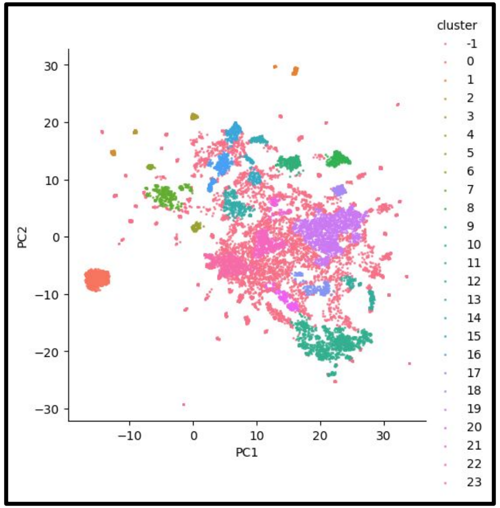
- Sentence Embeddings
  - Simple average.
  - Smooth inverse frequency weighted average.
    - Arora et al (2016)[8]

$SIF = a/(a + p(w))$

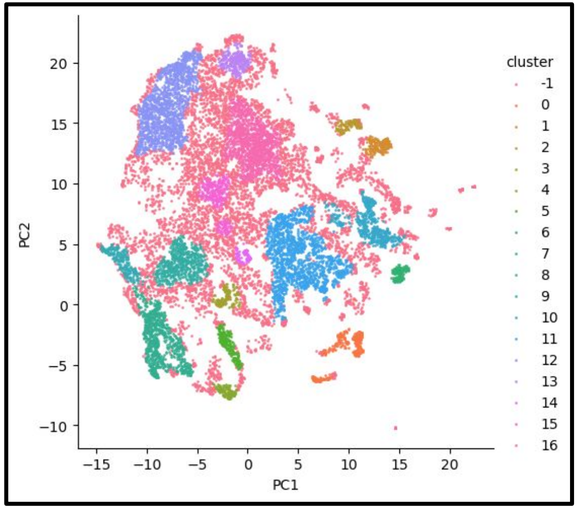
$a = 1e\text{-}5$
$p(w)$ = word frequency



Ideal



FastText



Bert

Sentence Vectorization

- Sentence Embeddings
  - Simple average.
  - Smooth inverse frequency weighted average.
    - Arora et al (2016)[8]

Sentence Vectorization

- Sentence Embeddings
  - Simple average.
  - Smooth inverse frequency weighted average.
    - Arora et al (2016)[8]
  - Smooth inverse TFIDF weighted average.
    - Zhao et al (2015)[9]
    - Correa et al (2017)[10]

Sentence Vectorization

- Sentence Embeddings
  - Simple average.
  - Smooth inverse frequency weighted average.
    - Arora et al (2016)[8]
  - Smooth inverse TFIDF weighted average.
    - Zhao et al (2015)[9]
    - Correa et al (2017)[10]

TFIDF

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Sentence Vectorization

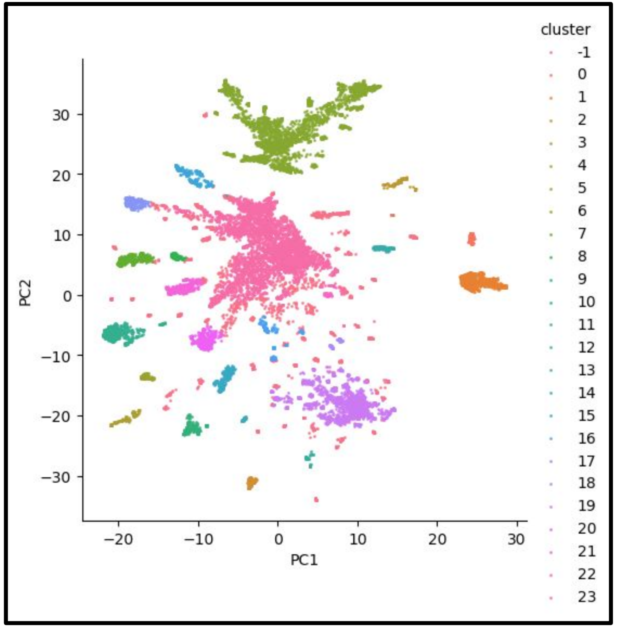
- Sentence Embeddings
  - Simple average.
  - Smooth inverse frequency weighted average.
    - Arora et al (2016)[8]
  - Smooth inverse TFIDF weighted average.
    - Zhao et al (2015)[9]
    - Correa et al (2017)[10]

TFIDF

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents



FastText

Dimensionality Reduction

- Problems with high dimensional datasets
  - training a data model usually requires vast time and space complexity
  - often leads to overfitting
  - not all the features available are relevant to our problem
  - not plottable
  - *Curse of Dimensionality*

Dimensionality Reduction

- Curse of Dimensionality
  - phenomena that occurs, when working with high dimensional data
    - *Data sparsity:* represented space grows quicker than the data

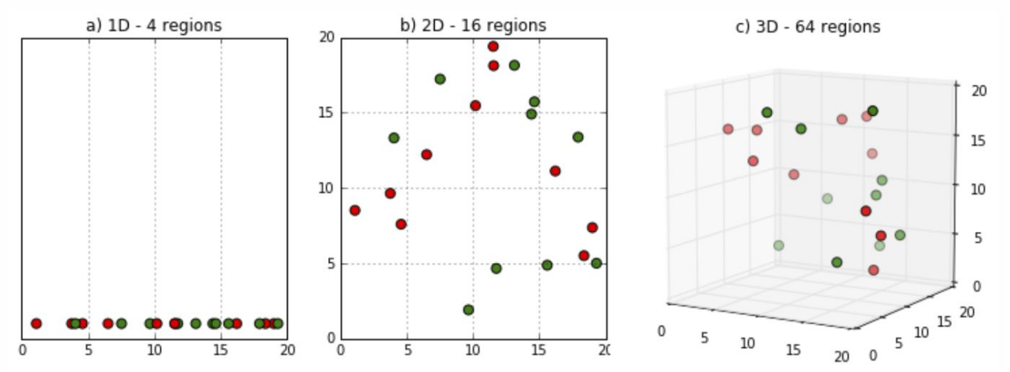


Figure 1: referring to "*Data sparsity*" [14]

Dimensionality Reduction

- Curse of Dimensionality
  - phenomena that occurs, when working with high dimensional data
    - *Data sparsity:* represented space grows quicker than the data
    - *Closeness of the data:* the higher the dimension the further data points may seem

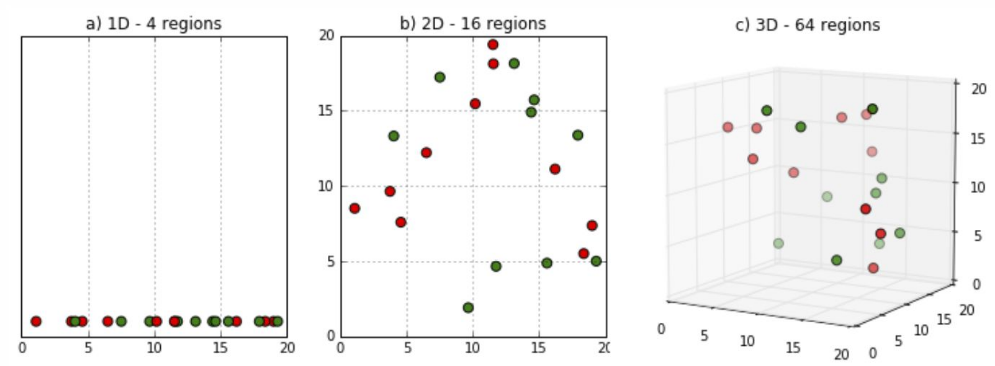


Figure 1: referring to "*Data sparsity*" [14]

Figure 2: referring to "*Closeness of the data*" [14]

Dimensionality Reduction

- Curse of Dimensionality
  - phenomena that occurs, when working with high dimensional data
    - *Data sparsity:* represented space grows quicker than the data
    - *Closeness of the data:* the higher the dimension the further data points may seem

  - problem to achieve statistical significance from the high dimensional data
    - the amount of data needed to support a sound and reliable result often grows exponentially with the dimensionality
      - common data organization strategies become inefficient

Dimensionality Reduction

- Problems with high dimensional datasets
  - training a data model usually requires vast time and space complexity
  - often leads to overfitting
  - not all the features available are relevant to our problem
  - not plottable
  - *Curse of Dimensionality*

- What is dimensionality reduction about?
  - discovering non-linear, non-local relationship in data
  - reducing noise by reducing the dimensions
  - easier to apply simple learning algorithms to smaller subset

Dimensionality Reduction

- PCA (Principal Component Analysis)
    - most tried technique
    - linear DR technique
    - transforms variables into a new set of features (principle components)
    - which are a linear combination of the original variables
    - and converts the correlations among all of the features into a 2D-Graph
    - features that are highly correlated cluster together
    - differences along the PC1 axis are more important than the differences along the PC2 axis

Dimensionality Reduction

- PCA (Principal Component Analysis)
  - most tried technique
  - linear DR technique
  - transforms variables into a new set of features (principle components)
  - which are a linear combination of the original variables
  - it projects the original data onto a direction which maximizes variance
  - and converts the correlations among all of the feature into a 2D-Graph
  - features that are highly correlated cluster together
  - differences along the PC1 axis are more important than the differences along the PC2 axis

- UMAP (Uniform Manifold Approximation and Projection)
  - relatively new technique (2018)
  - non-linear DR technique
  - has a solid theoretical mathematical background as a manifold approximation technique
  - algorithm balances between emphasizing local versus global structure
  - first models the high dimensional set with a fuzzy topological structure
  - searching for a low dimensional projection of the data that has the closest possible equivalent fuzzy topological structure

Dimensionality Reduction

- Our approach
  - Using PCA and UMAP in combination
  - Expectations:
    - improves computation time

Dimensionality Reduction

- Our approach
  - Using PCA and UMAP in combination
  - Expectations:
    - ~~improves computation time~~

→ no big difference in computation time

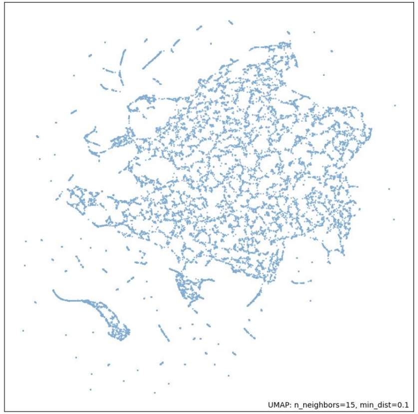
Dimensionality Reduction

- Our approach
  - Using PCA and UMAP in combination
  - Expectations:
    - ~~improves computation time~~
    - removes noise

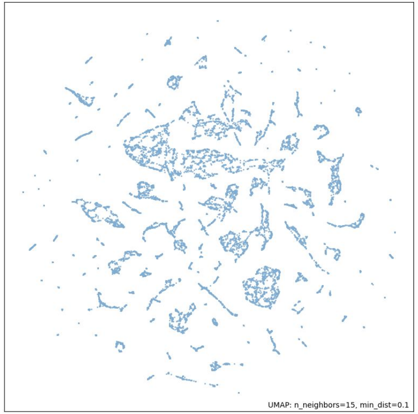
Dimensionality Reduction

- Our approach
  - Using PCA and UMAP in combination
  - Expectations:
    - ~~improves computation time~~
    - **removes noise**

running dimensionality reduction only with UMAP (without PCA)



running dimensionality reduction with UMAP and PCA



Clustering

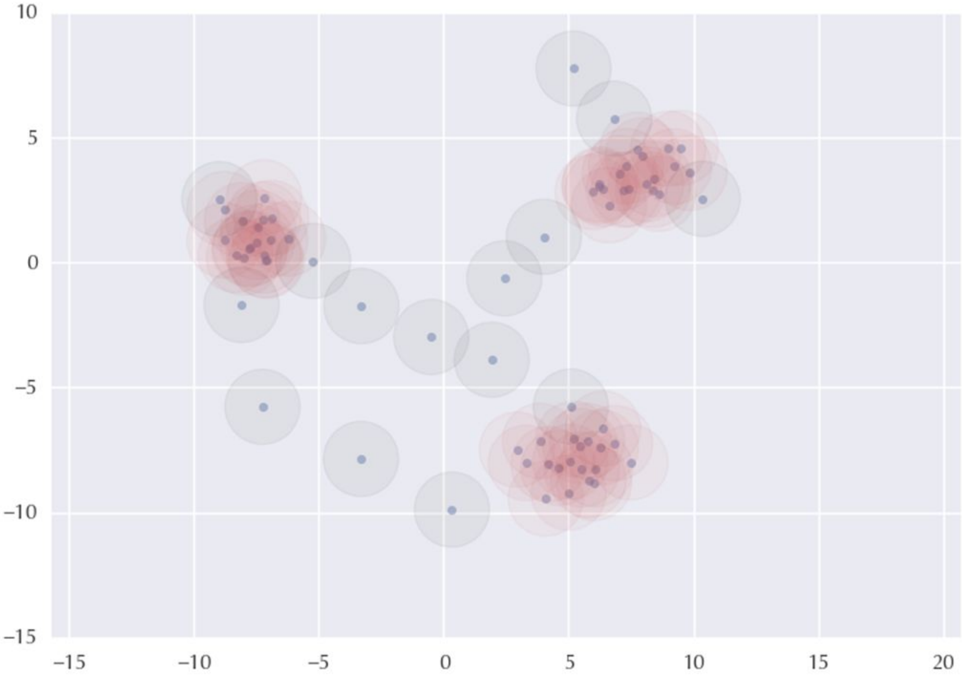
- HDBSCAN (Hierarchical Density Based Clustering)

	Flat	Hierarcical
Centroid / Parametric	k-means GMM	Ward Complete-linkage
Density/ Non-Parametric	DBSCAN Mean shift	HDBSCAN

[11]

HDBSCAN

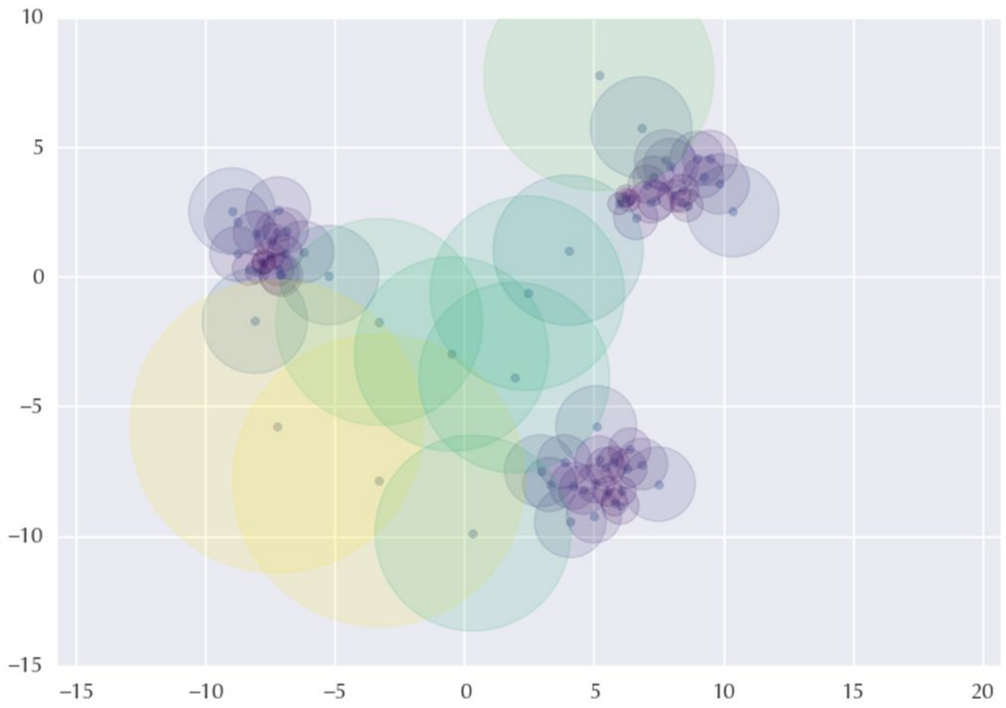
- Works on DBSCAN
  - Density ??
- Circles of radius : ε
- Nearest neighbours : K



[11]

HDBSCAN

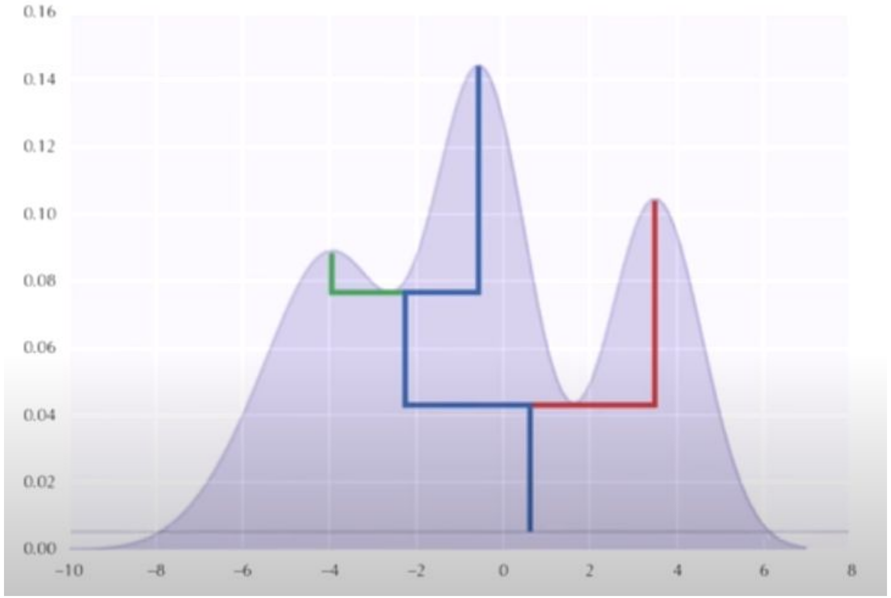
- Fix K
- Small circles - dense region
- With this we have PDF at each point.
- Smaller ε the denser the points

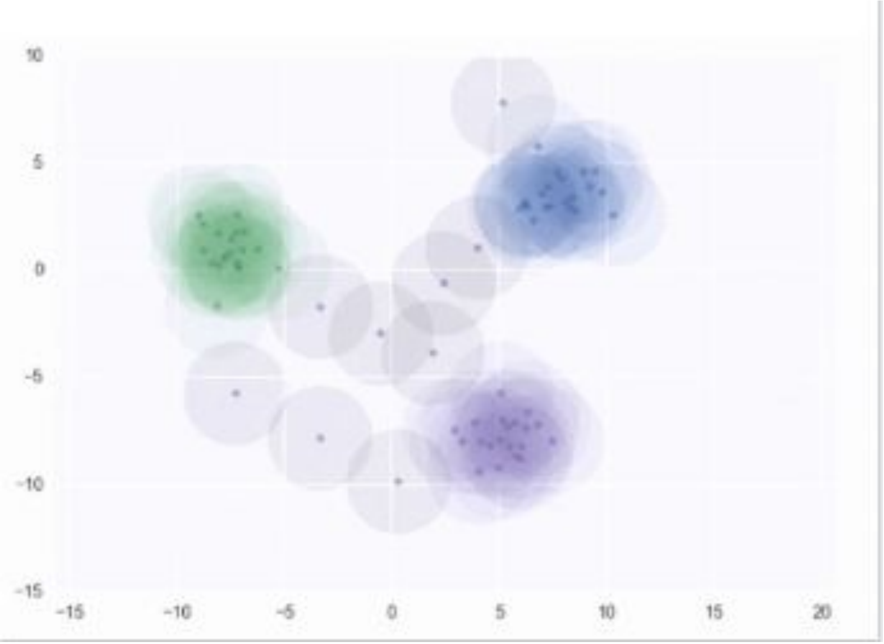


[11]

HDBSCAN

- As we lower down from the PDF hill, points get added.

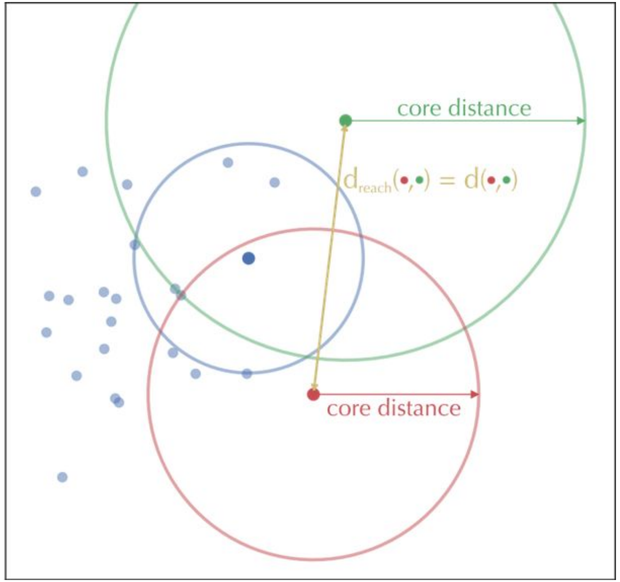


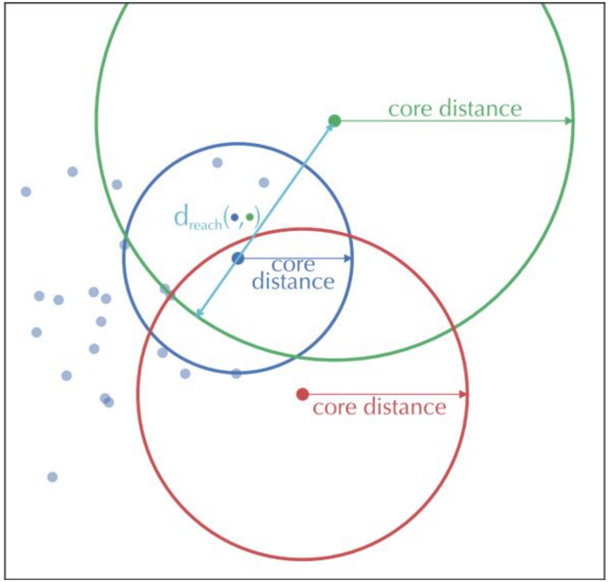


[11]

HDBSCAN

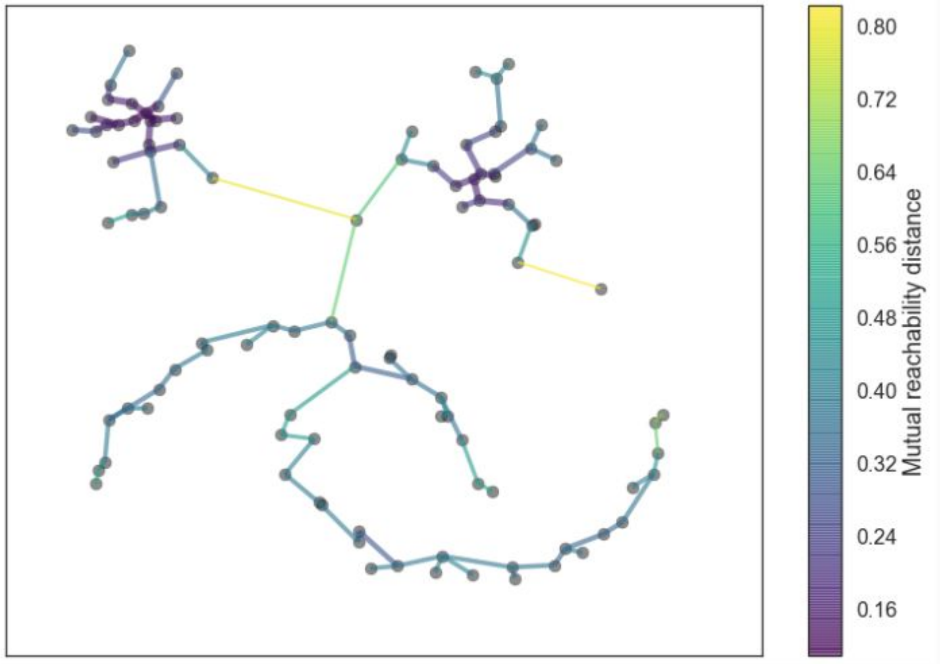
Mutual Reachability Distance

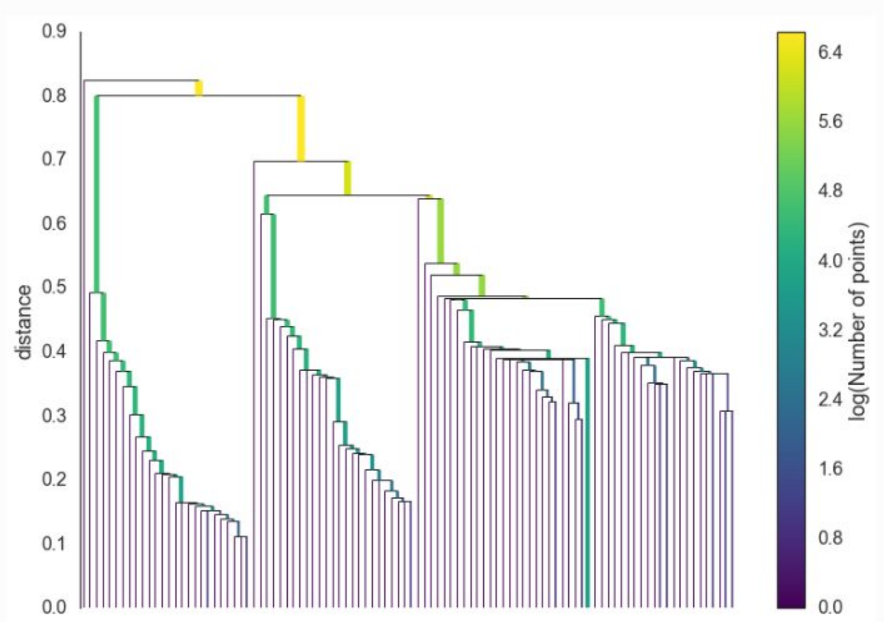




$$d_{\mathrm{mreach}}(X_i, X_j) = \begin{cases} \max\{\kappa(X_i), \kappa(X_j), d(X_j, X_j)\} & X_i \neq X_j \\ 0 & X_i = X_j \end{cases}$$ [11]

HDBSCAN



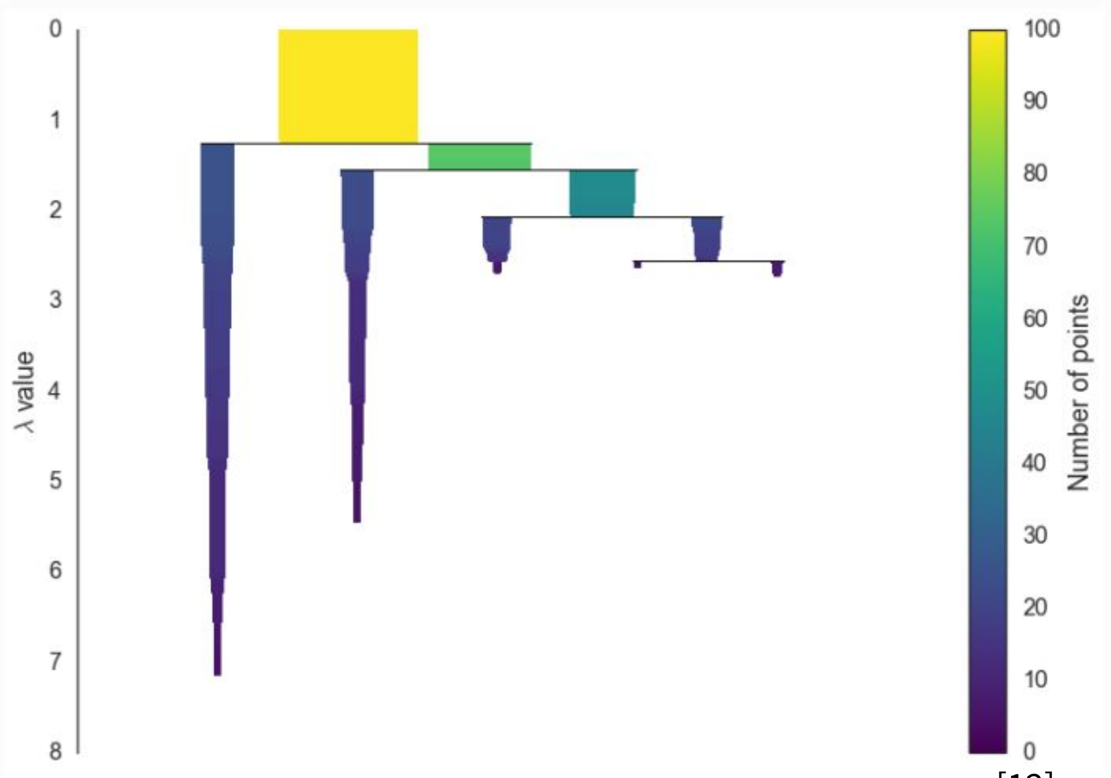


[12]

Minimum spanning tree (prim's algo)

HDBSCAN

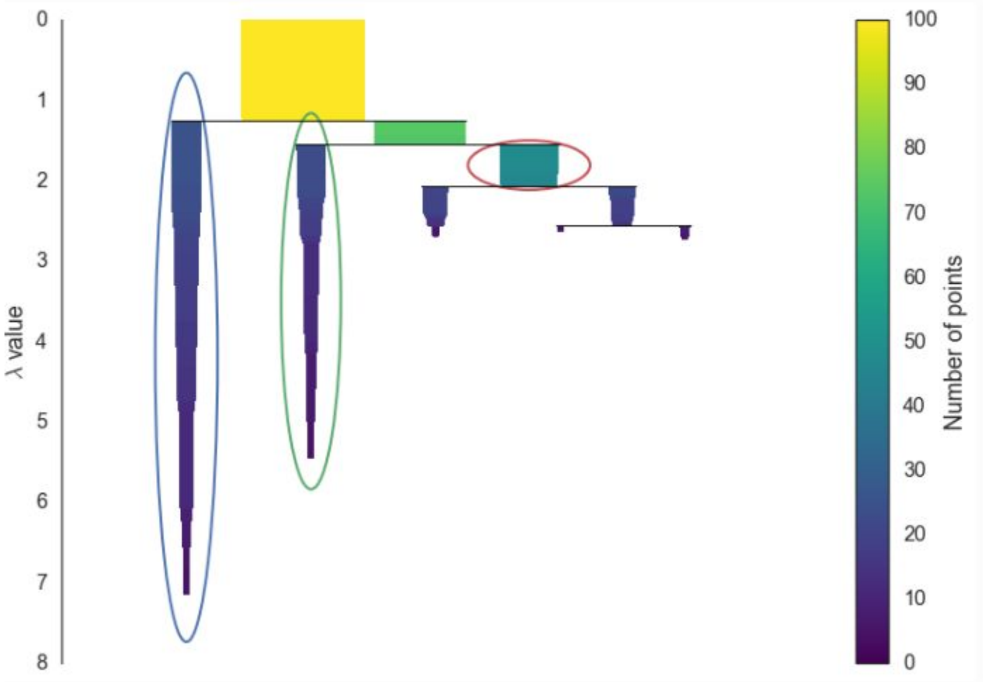
- Parameter - minimum cluster size
- Check whether cluster has fewer points.
  - Yes -> points falling out of a cluster
  - Else true cluster persist
- Looping hierarchy → end up small tree.



[12]

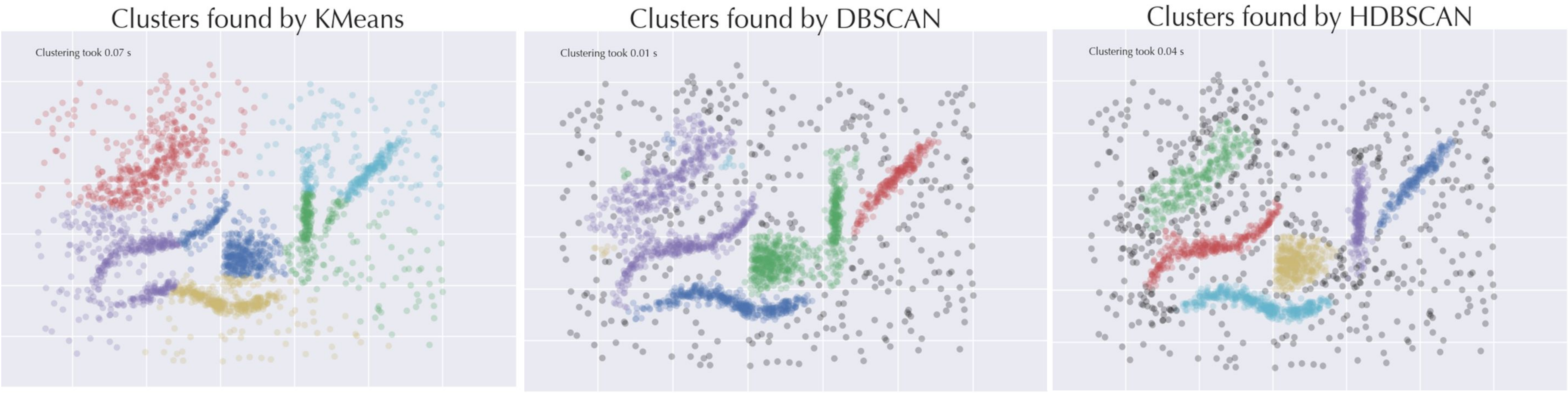
HDBSCAN

- Last step is to Extract the clusters
- Choose the cluster which will give most number of points.
  - Idea is if parent is chosen then , child nodes can't be chosen
- So it's cut on the hierarchical tree



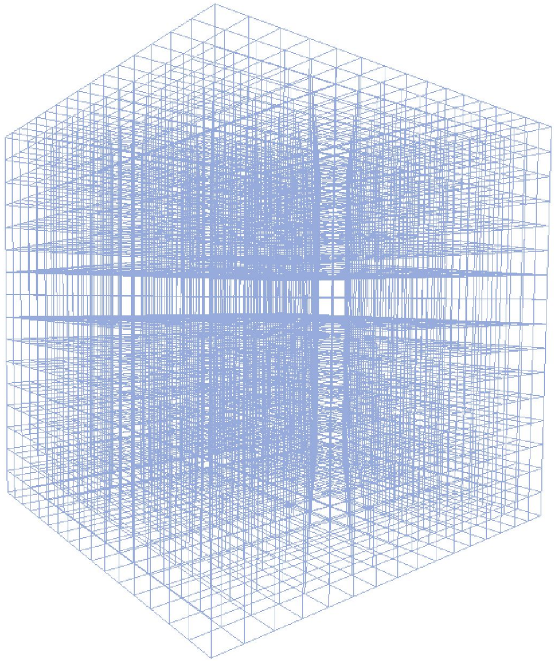
[12]

HDBSCAN


Clusters found by KMeans
Clustering took 0.07 s
Clusters found by DBSCAN
Clustering took 0.01 s
Clusters found by HDBSCAN
Clustering took 0.04 s

Grid Search

- Building a grid out of all input parameters for each algorithm used in the pipeline
- Running pipeline for every parameter configuration (with same tuning set)
- Define metrics and pick n configurations with highest scores on tuning set
- Evaluate the picks by running the pipeline on a validation set
- Log the results to later search for the best suiting configuration



Grid Search - Parameters

```
pipeline = {
    'normalization':
        {'function': normalize_data,
         'parameters': {},
         'name': 'MinMaxScaler'},
    'dim_reduction':
         {'function': reduce_dimensions,
          'parameters': {},
          'name': 'UMAP'},
    'cluster_algorithm':
        {'function': get_cluster_ids,
         'parameters': {},
         'name': cluster_algorithm}
}
```

Defining a grid search pipeline
(in our case: 1. normalization, 2. dimensionality reduction, 3. clustering) → Call python function by pipeline entry 'function'

Produce grid search output (scores) dynamically
We designed each pipeline component in a way that they all get the same type of inputs

```
grid_search_data = self.pipeline_components[pipeline_step](grid_search_data,
                                                             self.algorithms[pipeline_step],
                                                             node[pipeline_step])
```

Grid Search - Logging

Create a connection to MongoDB via pymongo library

Makes it really easy to…
… read data from database
… write results to database
… iterate through collections (Mongo specific)
… convert python objects to mongo objects
… manipulate database entries

We wrote a wrapper for easier handling:
Class using pymongo's MongoClient with custom methods like 'write', 'get_grid_id', etc.

```
DB_NAME = 'pnlp'
URL = f'mongodb+srv://{username}:{password}' \
      f'@cluster0-8ejtu.azure.mongodb.net/{DB_NAME}' \
      f'?retryWrites=true&w=majority&ssl=true&ssl_' \
      f'cert_reqs=CERT_NONE'

class MongoAccessor:
    """
    Object for data base access
    """
    def __init__(self):
        self.collection_name = ""
        self.client = MongoClient(URL)
        self.database = self.client[DB_NAME]
        self.collection = None
        self.collection_items = None
        self.grid_id = None
```

Representative Sentences

- Aim
  - explore cluster
  - use as basis for label

Representative Sentences

- Aim
  - explore cluster
  - use as basis for label
- Statistical approach:
  - Sentences are assigned value based on sum of all frequencies of tokens
  - Highest values are used to find representative sentences

Representative Sentences

- Aim
  - explore cluster
  - use as basis for label
- Statistical approach:
  - Sentences are assigned value based on sum of all frequencies of tokens
  - Highest values are used to find representative sentences
- Word embedding approach:
  - Mean embedding of all sentences in cluster is calculated
  - Closest neighbours based on cosine distance are extracted as representative sentences

Topic Labeling

- Final step in pipeline: give clusters meaningful titles
- Difficult problem:
  - Unsupervised task
  - What does a good solution look like?
  - Several equally good solution possible
  - Labels often contain some kind of abstract concept
- Several approaches:
  - Find nearest neighbour of mean embedding of cluster in vector space
  - TF-IDF of all comments in cluster → important keywords for each topic
  - Use keywords and calculate nearest neighbour
  - Use representative sentences of the cluster, calculate mean embeddings of these sentences, take the average of the means and find nearest neighbour

Gensim Library - “Generate Similar”

- Free library for unsupervised semantic modeling from plain text [13]
- Contains number of pretrained models like Fasttext, Word2Vec, GloVe
- Broad range of statistical (e.g. LDA, BOW) and semantic (word, sentence and document embeddings) analysis tools
- Easy to use: e.g. model.most_similar(word_list) gives list of most similar words to given words
- Runtime optimized implementation
- Caveat: the documentation is quite unstructured, google directly what you need ;)

Gensim - Mini-Demo

- Load pretrained word embedding model:

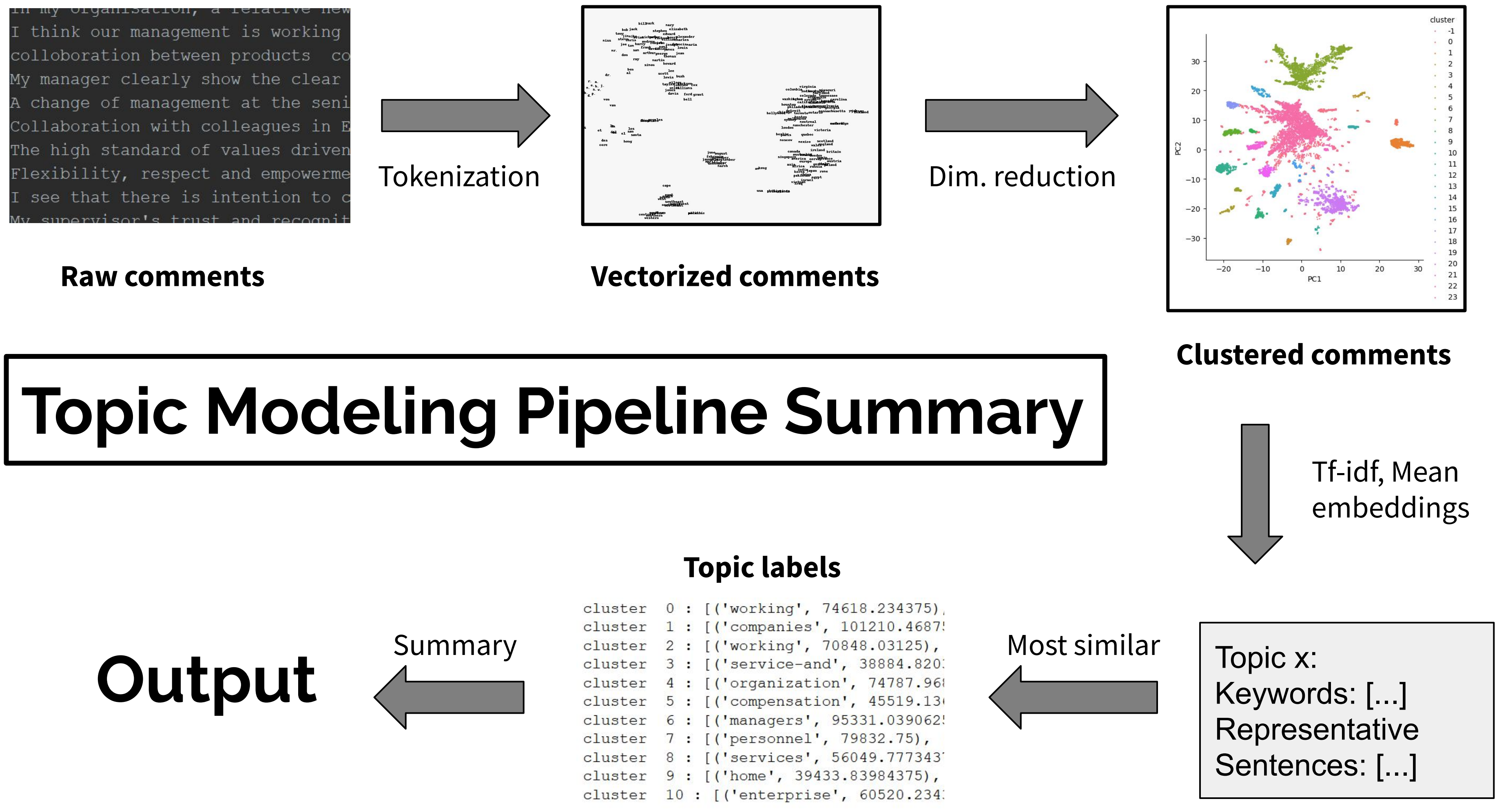
```
# Load Word2Vec news model.
vector_path = 'GoogleNewsVectors300.bin'
model = gensim.models.KeyedVectors.load_word2vec_format(vector_path, binary=True)
```

- Get embeddings for tokenized comments:

```
    embeddings = []
    for token in tokens:
        try:
            embeddings.append(model[token])
```

- Calculate cluster labels via keywords:

```
cluster_name, _ = model.most_similar(positive=keywords, topn=1)[0]
return cluster_name
```



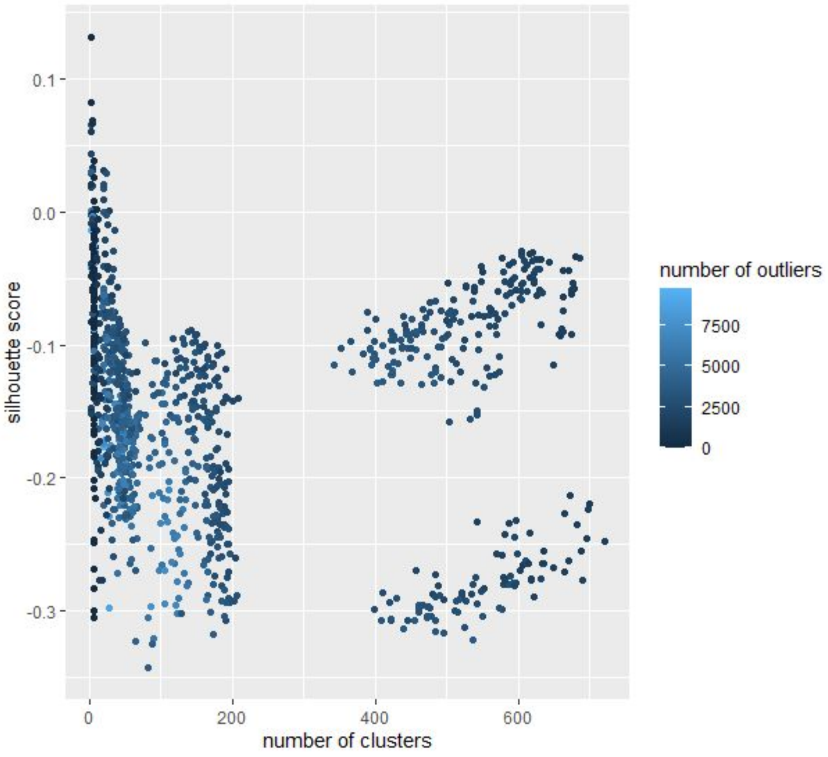
Topic Modeling Pipeline Summary
I think our management is working
colloboration between products co
My manager clearly show the clear
A change of management at the seni
Collaboration with colleagues in E
The high standard of values driven
Flexibility, respect and empowerme
I see that there is intention to c
Raw comments
Tokenization
Vectorized comments
Dim. reduction
cluster
PC1
PC2
Clustered comments
Tf-idf, Mean embeddings
Topic x:
Keywords: [...]
Representative Sentences: [...]
Most similar
Topic labels
cluster 0 : [('working', 74618.234375),
cluster 1 : [('companies', 101210.4687
cluster 2 : [('working', 70848.03125),
cluster 3 : [('service-and', 38884.820
cluster 4 : [('organization', 74787.96
cluster 5 : [('compensation', 45519.13
cluster 6 : [('managers', 95331.039062
cluster 7 : [('personnel', 79832.75),
cluster 8 : [('services', 56049.777343
cluster 9 : [('home', 39433.83984375),
cluster 10 : [('enterprise', 60520.234
Summary
Output

Results

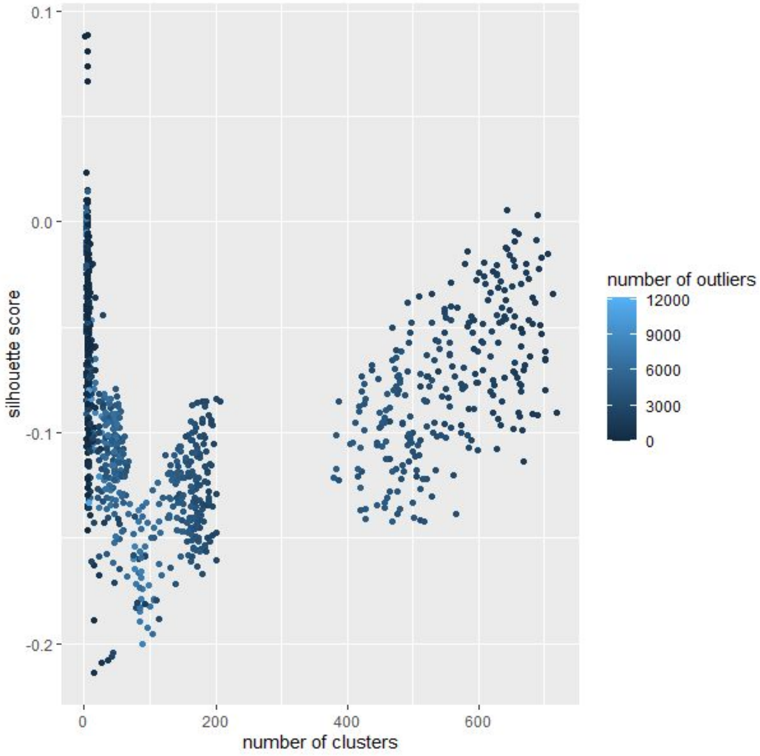
1. Analyse the grid search
    - 1085 nodes (ran over night locally)
2. Identify best configurations
    - FastText
    - BERT
3. Inspect the best configuration
    - Visualizations
    - Labeling

Results - Grid Search

FastText



BERT



Results - Grid Search

Grid with 1085 nodes (ran over night locally)

Scores:

- Number of outliers
- Number of clusters
- Silhouette Score

Best results:

	node	score	n_clusters	n_outliers	MinMaxScaler feature_range	UMAP metric	UMAP random_state	UMAP spread	UMAP n_neighbors	UMAP min_dist	hdbscan alpha	hdbscan leaf_size	hdbscan min_samples	hdbscan metric	hdbscan min_cluster_size	hdbscan cluster_selection_epsilon
BERT	3	0.06674	5	3	[0, 1]	cosine	42	4	20	0.0	0.1	40	9	canberra	100	0.3
FASTTEXT	648	0.03822	5	9	[0, 1]	canberra	42	5	40	0.0	0.1	40	9	canberra	100	0.3

Results - Labels

BERT configuration:

Label (Top 5 key words):

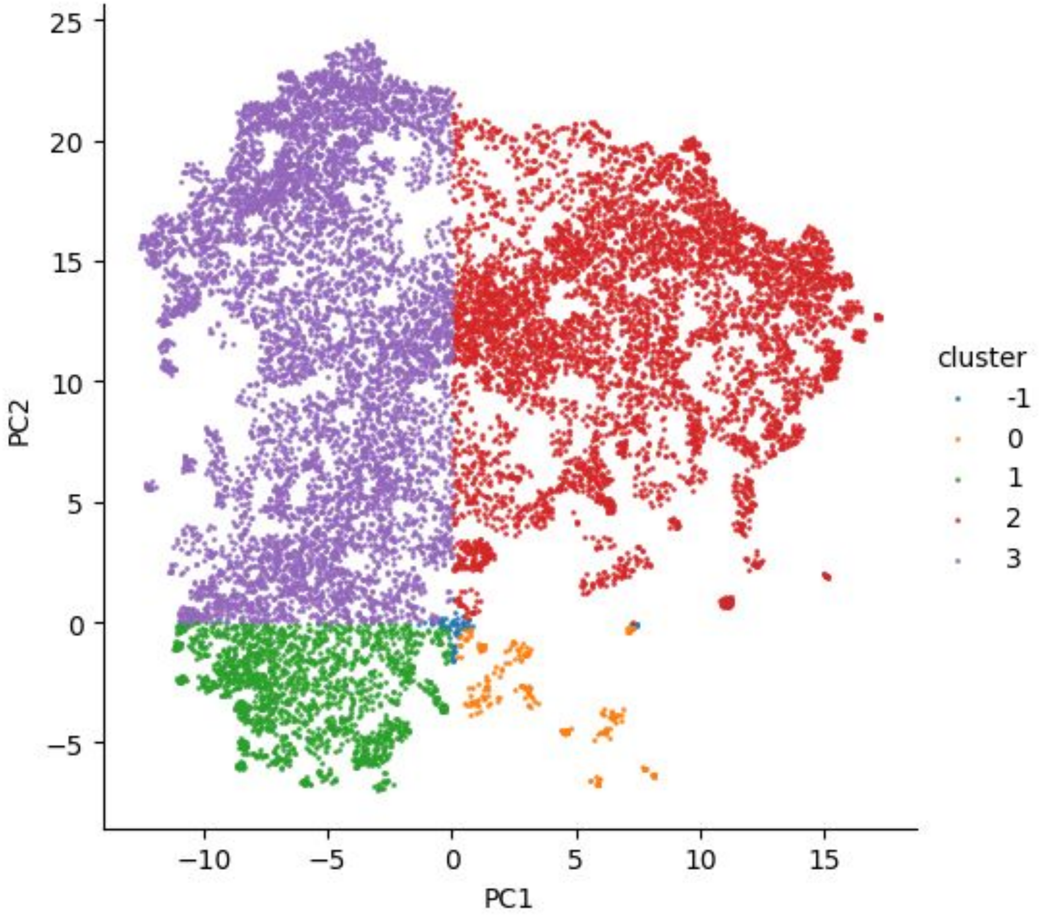
Cluster -1: 'team', 'work', 'working', 'job', 'done'

Cluster 0: 'nothing', 'need', 'work', 'time', 'comment'

Cluster 1: 'team', 'communication', 'work', 'employee', 'good'

Cluster 2: 'team', 'company', 'work', 'need', 'well'

Cluster 3: 'employee', 'need', 'company', 'work', 'team'



Results - Representative Sentences

Cluster 3 (Keywords: 'employee', 'need', 'company', 'work', 'team'):

1. 'Sales force and strategy, customer focus and extra care to be improved. Process dvlpt plan to be done touching the basic aspects and getting common employee feedback not just by reports and presentations. Right people should be chosen in recruitment',
2. 'The working environment needs to be improved , Less Pressure (Mgt monitor every single action of employees). More team building (only one EOY is not enough).No 2 ways communication , SMT seems to always impose rather than encourage',
3. 'Pay band and Salary levels not correct/low comparing to market makes much easier for competition to headhunt our talented and high potential team members. Need to consider different actions to multiply/strengthen employee engagement and loyalty.',
4. "The management team needs to be more interested in seeing employees' well-being needs, benefits have been removed without seeing investments in other minimum things to make the employee feel well, comparing the sites, BCS is in demotivating conditions.",
5. 'Operation and customer service have to be improved to the level what Key account managers are currently doing so that we can convert Kay Account Manager to business development rather than solving problem of daily issue.'

Next Steps

We want a 'real' product

Input: comments
→ Output: something that can be used

Improve topic labeling

Improve topic labels with cluster mean nearest neighbor search

No optimal clusters

- Clusters are not really separable
- Data from grid search is not analyzed in detail

Implement soft clustering

Get possibility instead of fixed cluster IDs and use insights from grid search data to improve clustering pipeline

Frequent words appearing in each cluster

We have words like 'work' or 'team' which are seen as representative for all clusters

Get rid of redundant representations

Increase the filtering or come up with a smart way to detect real important representations

Discussion

Label (Top 5 key words):

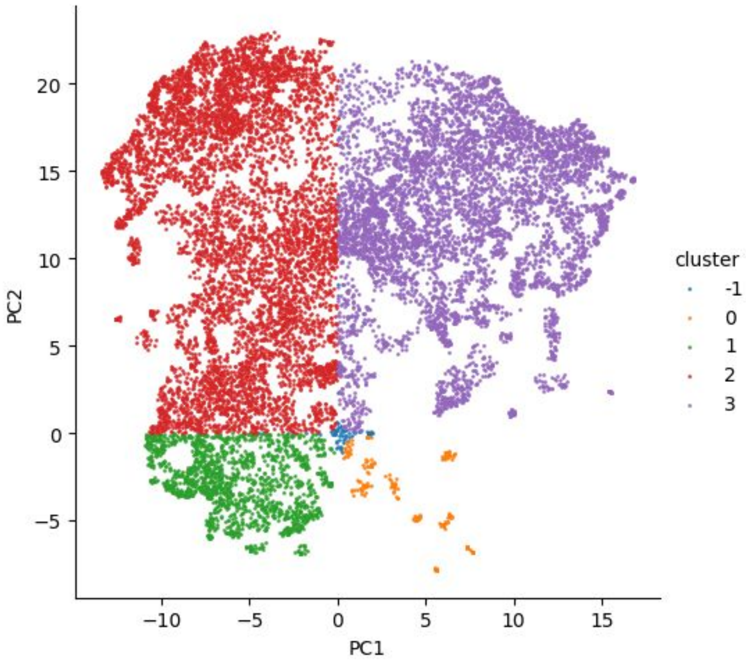
Cluster -1: 'team', 'work', 'working', 'job', 'done'

Cluster 0: 'nothing', 'need', 'work', 'time', 'comment'

Cluster 1: 'team', 'communication', 'work', 'employee', 'good'

Cluster 2: 'team', 'company', 'work', 'need', 'well'

Cluster 3: 'employee', 'need', 'company', 'work', 'team'



Potential Discussion Points

- How would you approach further analysis of the data?
- Have you used other topic modeling mechanisms?
- How would you deal with the shared words per class?

Thank you for your attention.

References

- [1]David M. Blei, Andrew Y. Ng, Michael I. Jordan: Latent dirichlet allocation. In: Journal of Machine Learning Research, Jg. 3 (2003), S. 993–1022.
- [2]Hurek, Radim. Gensim: LDA Model. https://radimrehurek.com/gensim/auto_examples/tutorials/run_lda.html
- [3]Bastani, Kaveh & Namavari, Hamed & Shaffer, Jeffry. (2018). Latent Dirichlet Allocation (LDA) for Topic Modeling of the CFPB Consumer Complaints.
- [4]https://www.codeproject.com/Articles/439890/Text-Documents-Clustering-using-K-Means-Algorithm
- [5]https://fasttext.cc/
- [6]Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need.
- [7]https://towardsdatascience.com/a-friendly-introduction-to-text-clustering-fa996bcefd04

References

- [8]Arora, S., Liang, Y., & Ma, T. (2016). A simple but tough-to-beat baseline for sentence embeddings. Paper presented at 5th International Conference on Learning Representations, ICLR 2017, Toulon, France.
- [9]Zhao, Jiang & Lan, Man & Tian, Jun. (2015). ECNU: Using Traditional Similarity Measurements and Word Embedding for Semantic Textual Similarity Estimation. 117-122. 10.18653/v1/S15-2021.
- [10]Edilson Anselmo Correa Junior, Vanessa Marinho, and Leandro Santos. (2017). NILC-USP at SemEval-2017 Task 4: A Multi-view Ensemble for Twitter Sentiment Analysis. In Proceedings of the 11th International Workshop on Semantic Evaluation. Vancouver, Canada, SemEval '17, pages 610–614.

References

- [11] https://www.youtube.com/watch?v=dGsxd67IFiU&t=1151s
- [12] How HDBSCAN Works
- [13] https://radimrehurek.com/gensim/index.html
- [14] https://deepai.org/machine-learning-glossary-and-terms/curse-of-dimensionality