Customer Segmentation Using Unsupervised Natural Language Processing

NexTech 2019

Tim vor der Brück

September 2019

Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Traditional Customer Segmentation (cf. Lynn 2011)

- Based on clustering demgraphic, geographic and psychological variables like sex, age, city or profession
- Rather unreliable in detecting people's interest
- Thus, we propose an alternative method based on natural language processing

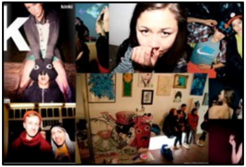
Task: assign people to marketing target groups

- Our business partner operates a Website where he organizes several contests
- in these contests, people can win certain prices like bicycles, MacBooks, pairs of sneakers
- For that, participants have to come up with a short descriptions (text snippets) what to do with their prize, or what they want to do in their dream holiday
- Based on these text snippets, the participants were distributed into one of 6 target groups

Contest text snippets provided by the participants

1. Jordan: Ride through the desert and marveling Petra during sunrise before the arrival of tourist buses
2. Cook Island: Snorkeling with whale sharks and relaxing
3. USA: Experience an awesome week at the Burning Man Festival

Target Groups (Youth Milieus)



Keywords

- Each youth milieu is described by a set of keywords
- Keywords are currently defined manually
- Examples:
    - Young Performer: rich, elite, luxury, luxurious
    - Action Sportsmen: sports, fitness, music

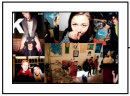

NITRO CIRCUS

Keywords

Keywords

Keywords

Keywords



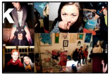
Keywords

Keywords

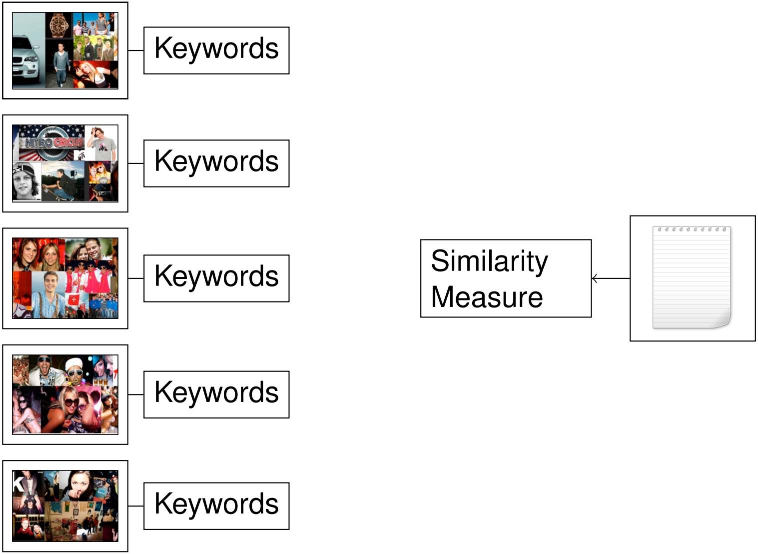
Keywords

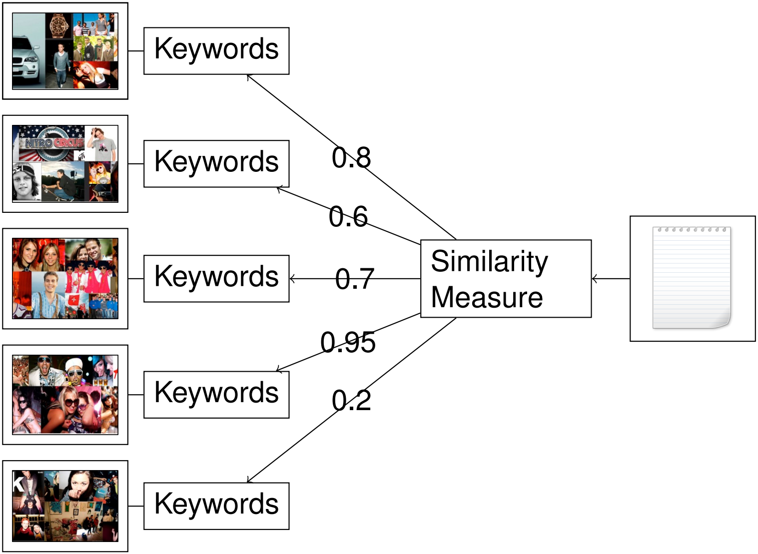
Keywords

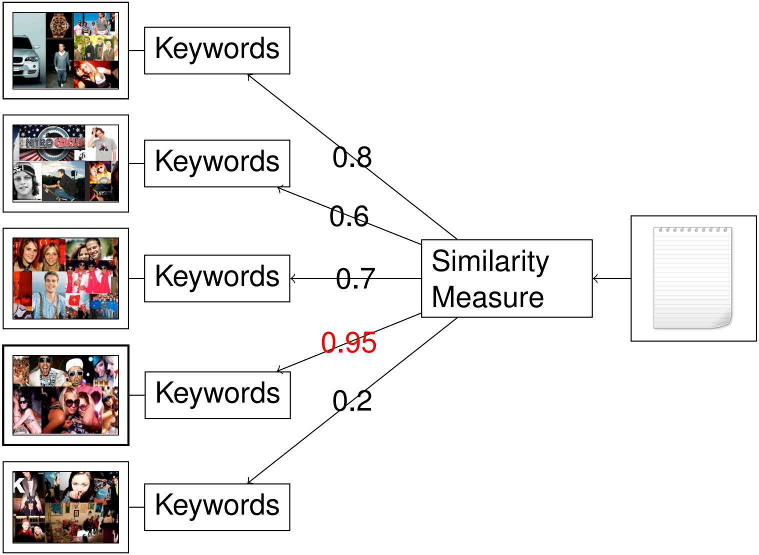
Keywords



Keywords


Keywords
Keywords
Keywords
Keywords
Keywords
Similarity Measure

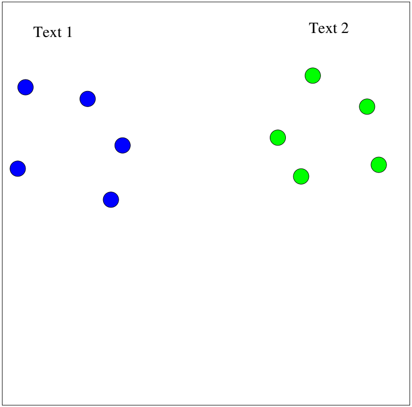

Keywords
Keywords
Keywords
Keywords
Keywords
0.8
0.6
0.7
0.95
0.2
Similarity Measure


NITRO CIRCUS
Keywords
Keywords
Keywords
Keywords
Keywords
0.8
0.6
0.7
0.95
0.2
Similarity Measure

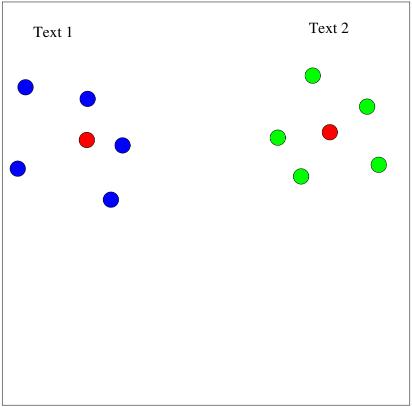
Word Vectors / Embeddings

- Idea: Each word w is associated a fixed-length numerical vector emb(w) in a semantic space
- Semantic similar words have similar vectors
- These vectors are determined either by a neural network or co-occurrence statistics
- You can use these vectors for calculation: emb(king)-emb(man)+emb(woman)=emb(queen)
- Words can be semantically compared by taking the cosine of the angle between these vectors (cosine measure)

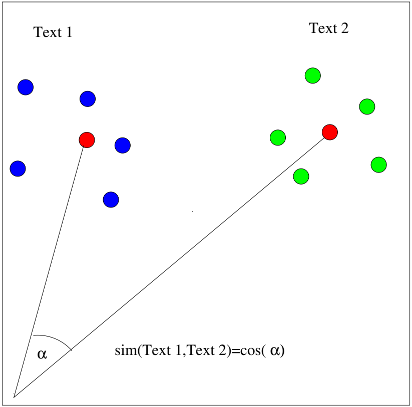
Standard approach - Centroid of Embeddings (CE)



Standard approach - Centroid of Embeddings (CE)



Standard approach - Centroid of Embeddings (CE)



Standard approach (CE)

Task: Estimate semantic similarity between text t text u

- Compute word embeddings for all words occurring in text t and u
- Compute the two centroids $\mathbf{C_t}$ and $\mathbf{C_u}$ of the word embeddings
- Similarity is given by: $cos(\angle(\mathbf{C_t}, \mathbf{C_u}))$

Drawback of standard approach

Let

- $x_1, \ldots, x_m$ embedding vectors of document t,
- $y_1, \ldots, y_n$ embedding vectors of document u
- C_t the centroid for document t
- C_u the centroid for document u

$$cos(\angle(\mathbf{C_t}, \mathbf{C_u})) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \langle x_i, y_j \rangle}{nm\|\mathbf{C_t}\| \cdot \|\mathbf{C_u}\|}$$

Small cosine similarity values can have in aggregate a considerable impact on the result $\rightarrow$ Method is vulnerable to noise

Noise reduction techniques

Noise reduction techniques are important, since we are dealing with short text snippets.

- Stop word list
- Weighted embeddings (e.g., tf-idf)
- Outlier robust centroids
- Use of Similarity Matrix

Stop Word List

- Manually specified list of words that are automatically removed from the text (here snippet)
- Usually contains function and very common words
- Pro: very fast
- Cons: crisp decision, no weighting function

Conclusion: Stop word filtering should be done but only with very common words

Weighted Embeddings

(cf. Brokos 16)

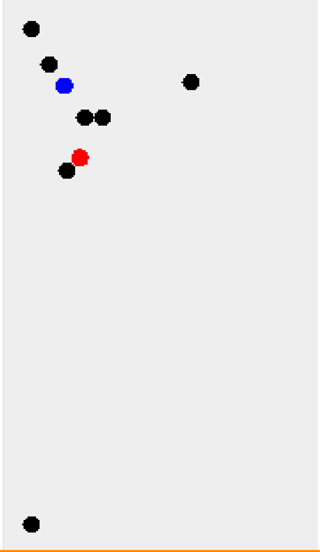
- word vectors are weighted according to the words relevance
- very common words are weighted less
- rather rare words occuring often in the given text are weighted strong
- most popular weighting scheme: tf-idf
- tf-idf(w,d): $tf(w,d) \cdot log(N/df(w))$
  - $tf(w,d)$ (term frequency): how often does word w occur in document d
  - $df(w)$ (document frequency): how often occurs word w in entire corpus
  - N: corpus size

Outlier Robust Centroid

(cf. I. Ilea et al. 2016)

- Instead of comparing centroids of word embeddings, one can compare outlier robust centroids
- Ordinary centroid: Linear combination of input vectors, each vector is weighted identically
- In Contrast: an outlier robust centroid weights outliers less strong than nearby vectors
- See talk in special session: SemaNLP

Outlier Robust Centroid



- ordinary centroid
- outlier robust centroid

Similarity Matrix

- In the following, we will focus on methods using the word similarity matrix F.
- Assuming the first text has n words, the second m
- Then the similarity matrix has $n\times$ m entries
- An entry F_{ij} specifies the similarity of word i of text 1 to word j of text 2
- We propose to use the matrix norm of this similarity matrix as similarity estimate

Basic Definitions: Text similarity

Let t, u be two text documents. Then $sn(t, u)$ is a normalized similarity estimate (measure): $\Leftrightarrow$

- Reflexivity: $sn(t, t) = sn(u, u) = 1$
- Symmetry: $sn(t, u) = sn(u, t)$
- Boundedness: $sn(t, u) \leq 1$

Basic Definitions: matrix norm

- Generalization of vector norm to matrices
- A measure how large the values of a matrix are
- Inherits usual vector norm properties
  - Positive definite: $\|\mathbf{A}\| \geq 0$ and $\|\mathbf{A}\| = 0 \Leftrightarrow \mathbf{A} = 0$
  - Subadditive: $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
  - Absolutely homogeneous: $\|(a\mathbf{A})\| = |a| \cdot \|\mathbf{A}\|$
- Submultiplicative: $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$
- Spectral radius: $\rho(\mathbf{A})$: largest absolute eigenvalue of **A**, not a matrix norm itself but lower bound of all matrix norms

Examples of matrix norms

Examples of matrix norms; $\mathbf{A}$ is an $m \times n$ matrix; $\rho(\mathbf{X})$ denotes the largest absolute eigenvalue of a squared matrix $\mathbf{X}$.

Name	Definition
Frob. norm	$\|\mathbf{A}\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n \|\mathbf{A}_{ij}\|^2}$
2-norm	$\|\mathbf{A}\|_2 := \sqrt{\rho(\mathbf{A}^\top \mathbf{A})}$
$L_{1,1}$-norm	$\|\mathbf{A}\|_{L_{1,1}} := \sum_{i=1}^m \sum_{j=1}^n \|\mathbf{A}_{ij}\|$
1-norm	$\|\mathbf{A}\|_1 := \max_{1\leq j\leq n} \sum_{i=1}^m \|\mathbf{A}_{ij}\|$
∞-norm	$\|\mathbf{A}\|_\infty := \max_{1\leq i\leq m} \sum_{j=1}^n \|\mathbf{A}_{ij}\|$

Basic definitions: Similarity matrix between documents

- $E(t)$: normalized embedding matrix of document t, column i is the embedding vector of word i of document t
- Similarity matrix (slightly simplified) $\mathbf{F} = E(t)^\top E(u)$
- F_{ij}: cosine similarity of word i of document t and j of document u

Example

- Document 1 contains two words
- Document 2 contains three words

$$\mathbf{F} := \begin{bmatrix} 0 & 0.5 & 0.8 \\ 0.1 & 0.7 & 0.2 \end{bmatrix}$$

Estimated similarity between word 1 of document 1 and word 3 of document 2 is 0.8

Similarity Measure Induced by Matrix Norm

Apply matrix norm on similarity matrix and use result as similarity estimate

$$sn_i(t,u) := \frac{\|F(t,u)\|_i}{\sqrt{\|F(t,t)\|_i \cdot \|F(u,u)\|_i}} \qquad (1)$$

Research Questions

For which matrix norms $\|\|\|_i$

- is sn_i a normalized similarity measure?
- is sn_i a valid SVM kernel?
- is sn_i independent of word order
- noise resilient

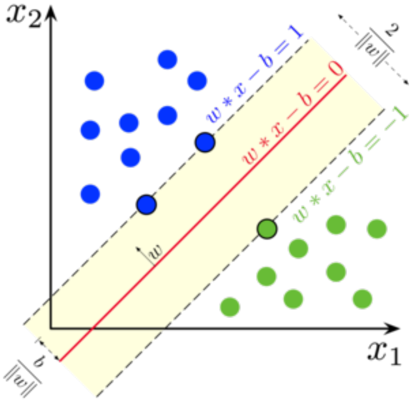
Additional question: How to deal with negative cosine similarity values, since matrix norms treat positive and negative values alike? In the following, we assume of cosine measure values are non-negative.

SVM

Support Vector Machine

- Supervised machine learning method
- Separates data by a hyperplane that maximizes the margin to the nearest vectors (called support vectors)
- Can transform the data prior to separation to higher dimensional space
- This transformation can be accomplished implicitly using a kernel function
- A kernel function is a similarity measure with certain properties (symmetry and positive-semidefiniteness)
- kernel matrix K: an item i,j of the kernel matrix K is the kernel function value of item i and j
- If a function is not a valid kernel (lacks one of the properties above), it is not guaranteed that the global optimum is found

SVM



	sim. measure	SVM kernel	indep word order	Noise resilient
sn_{sr}	✗	✗	✗	✓
sn_2	✓	?	✓	✓
sn_F	✓	?	✓	✓
sn_1	✗	✗	✓	✓
$sn_{L_{1,1}}$	✓	?	✓	✗

sn_{sr}: similarity estimate induced by spectral radius.

Normalized Similarity Measure - Recap

- Reflexivity: $sn(t, t) = 1$
- Symmetry: $sn(t, u) = sn(u, t)$
- Bounded by one: $sn(t, u) \leq 1$

Reflexivity

$$
\begin{aligned}
\mathbf{A} :=& E(t) \\
\mathbf{B} :=& E(u) \\
sn(t,t) =& \frac{\|\mathbf{A}^\top \mathbf{A}\|}{\sqrt{\|\mathbf{A}^\top \mathbf{A}\| \cdot \|\mathbf{A}^\top \mathbf{A}\|}} \\
=& \frac{\|\mathbf{A}^\top \mathbf{A}\|}{\sqrt{\|\mathbf{A}^\top \mathbf{A}\|^2}} \\
=& \frac{\|\mathbf{A}^\top \mathbf{A}\|}{\|\mathbf{A}^\top \mathbf{A}\|} \\
=& 1
\end{aligned}
\tag{2}
$$

$E(t)$: Embedding matrix of document t, which contains normalized embedding vectors stacked together

Symmetry

For showing symmetry it is sufficient to verify: $\|M^\top\| = \|M\| \forall M$

Proof.

$$
\begin{aligned}
sn(t,u) &= \frac{\|\mathbf{A}^\top \mathbf{B}\|}{\sqrt{\|\mathbf{A}^\top \mathbf{A}\| \cdot \|\mathbf{B}^\top \mathbf{B}\|}} \\
&= \frac{\|((\mathbf{A}^\top \mathbf{B})^\top)^\top\|}{\sqrt{\|\mathbf{A}^\top \mathbf{A}\| \cdot \|\mathbf{B}^\top \mathbf{B}\|}} \\
&= \frac{\|(\mathbf{A}^\top \mathbf{B})^\top\|}{\sqrt{\|(\mathbf{A}^\top \mathbf{A})^\top\| \cdot \|\mathbf{B}^\top \mathbf{B}\|}} \quad \text{(use the assumption above)} \\
&= \frac{\|(\mathbf{B}^\top \mathbf{A})\|}{\sqrt{\|(\mathbf{A}^\top \mathbf{A})\| \cdot \|(\mathbf{B}^\top \mathbf{B})\|}} \quad = sn(u,t)
\end{aligned}
$$

$\square$

Boundedness by 1

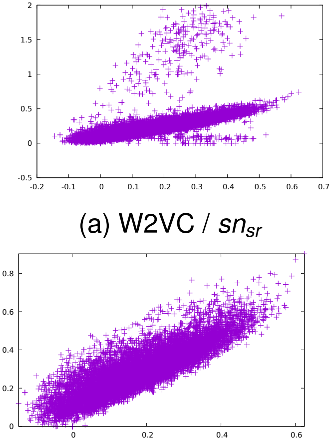
- Usually most difficult to prove
- Needs advanced knowledge of linear algebra (trail, eigenvalues)
- Easier is to prove that boundness is violated, can be done just by a counter-example

Evaluation

- Evaluation on three contests
  - Contest 1: Participants elaborated on their dream holiday
  - Contest 2: Participants elaborated what they would do with a pair of sneakers
  - Contest 3: Participants explained for what they needed one of 4 potential prices
- Each answer was labeled by 3 marketing experts
- Unique label was obtained by majority voting over expert answers

Table: Obtained accuracy values for semilarity estimates induced by several norm and baseline methods. W2VC=Centroid of Word Embeddings.

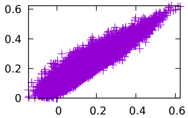
Method	Contest			
	1	2	3	all
Random	0.167	0.167	0.167	0.167
ESA	0.357	0.254	**0.288**	0.335
W2VC	0.347	**0.328**	0.227	0.330
Skip-Thought-Vectors	0.162	0.284	0.273	0.189
sn_2	0.370	0.299	**0.288**	**0.350**
sn_F	0.367	0.254	0.242	0.337
sn_1	**0.372**	0.299	0.212	0.343
sn_{sr}	0.353	0.313	0.182	0.326
$sn_{sr} + W2VC$	0.357	0.299	0.212	0.334



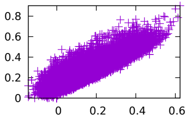
(a) W2VC / sn_{sr}

(b) W2VC / sn_2

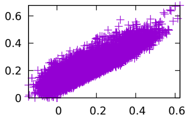
Figure: Scatter Plots of W2VC (cos. of word2vec centr.) and sn_{sr} / sn_2



(a) W2VC / $sn_{L_{11}}$



(b) W2VC / sn_2



(c) W2VC / sn_F

Figure: Scatter plots of cosine between centroids of Word2Vec embeddings (W2VC) vs *sn*.

Conclusion

- We presented an novel method to customer segmentation based on unsupervised natural language processing
- The prevalent approach to compare documents by cosine measure values of centroids is noise-vulnerable
- We described four methods that aim to reduce noise in the data
- One of these methods (matrix norms applied on similarity matrix) was evaluated and obtained overall superior results on three different contests