

Johannes Franke
IR: Query Understanding WiSe 24/25
Query Segmentation
10.12.2024

Beispiel




"new york" "times square" dance

"new york times" "square dance"

Beispiel


new york times square dance



10:53

NYC TIMES SQUARE DANCERS 2024


YouTube · Intothewoods_Dave
18.05.2024



0:40

Street Dancing in Times Square


TikTok · escobar.917
24.04.2024



26:36

Times Square New York Street dance performer ...

YouTube · YUMA 유마
07.04.2022




0:22

Weatherman Flashmob in Times Square! #NYC ...

YouTube · Times Square NYC
15.06.2023

Mehr anzeigen →

 Times Squares Square Dance Club
<https://timesquares.nyc> · [Diese Seite übersetzen](#)

Welcome to Times Squares | Times Squares

First hour is reserved for interested folks who want to take a chance on learning to square dance. No experience or partner needed. Admission FREE Then rotate ...

Weitere Fragen :

What is the New York street dance called?

▼

What is a square dance called?

▼

Which dance is famous in New York?

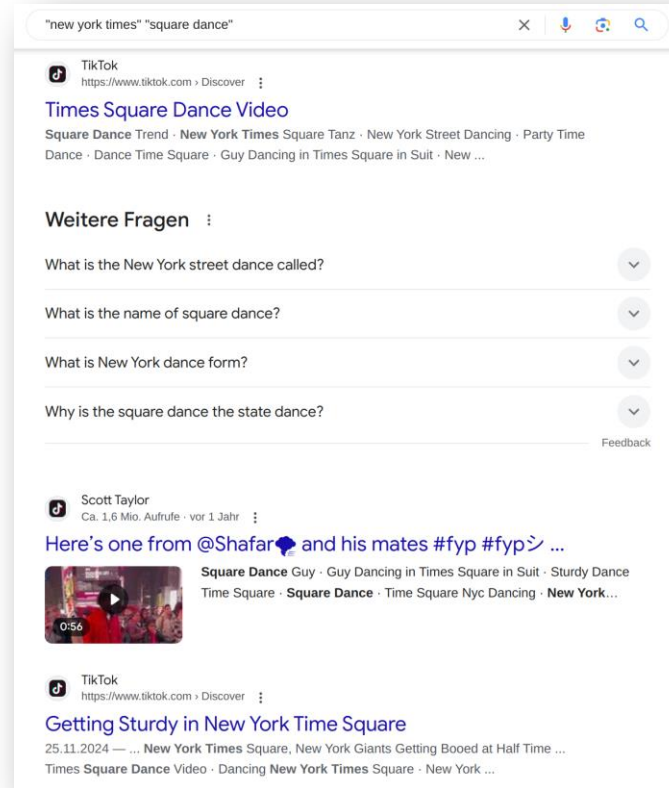
▼

FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

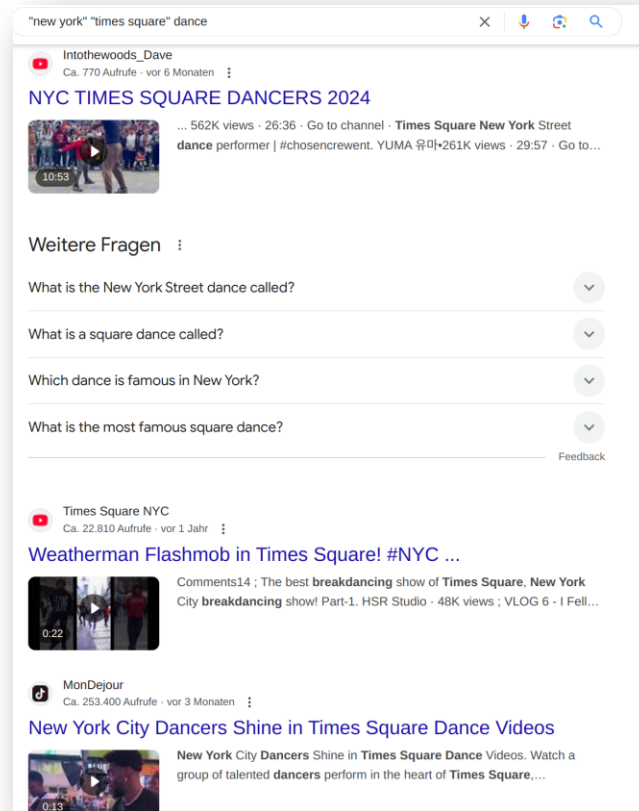
Beispiel | IR: Query Understanding WiSe 24/25 – Query Segmentation
Johannes Franke | 10.12.2024

2

Beispiel

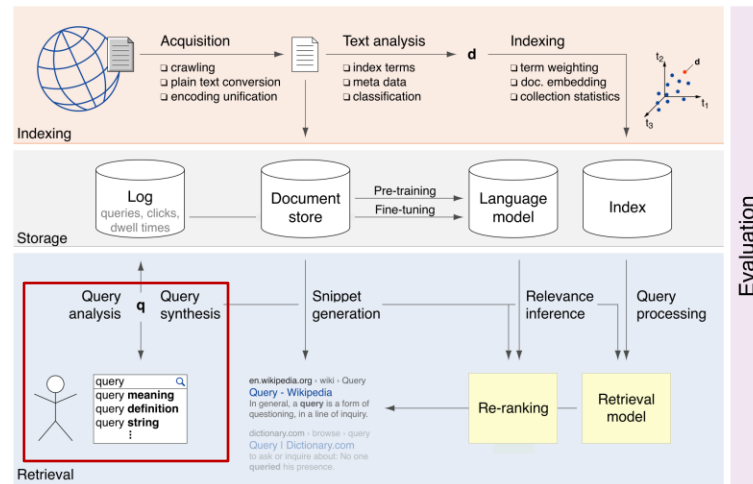


Beispiel



Grundlagen

- Aufteilung in Token-Sequenzen
- Query mit n Token hat $n - 1$ Breakpoints
- Segmentierungen haben Einfluss auf Bedeutung & Retrieval



Paper

- **Query Segmentation for Web Search**

- Risvik et al., 2003
- Publisher: The Web Conference

- **Learning Noun Phrase Query Segmentation**

- Bergsma et al., 2007

- Publisher: EMNLP-CoNLL

(Empirical Methods in Natural Language Processing and Computational Natural Language Learning)

Learning Noun Phrase Query Segmentation

Shane Bergsma and Qin Iris Wang
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
(bergsma, wqin}@cs.ualberta.ca

Query Segmentation for Web Search

Knut Magne Risvik
Fast Search & Transfer ASA
P.O. Box 4452 Høyrisleikskan
NO-7418 Trondheim, Norway
kmr@fast.no

Tomasz Mikolajewski
Fast Search & Transfer ASA
Friedmark 7
D-80334 München, Germany
tomasz@fast.no

Peter Boros
Fast Search & Transfer ASA
P.O. Box 4452 Høyrisleikskan
NO-7418 Trondheim, Norway
boros@fast.no

ABSTRACT

This paper describes a query segmentation method for search engines supporting inverse lookup of words and phrases. Data mining in query logs and document corpora is used to produce segment candidates and compute consistency measures. Candidates are considered in context of the whole query, and a list of the most likely segmentations is generated, with each segment annotated with a consistency value. For each segmentation a segmentation score is computed from consistency values of non-trivial segments, which can be used as a scoring criterion for the segmentation. We also point to a relevancy improvement in query evaluation model by means of consistency penalty.

Keywords

web search, query processing, data mining, query segmentation, query evaluation

1. INTRODUCTION

Web search engines are rapidly emerging into the most important application of the World Wide Web. Several challenges arise when trying to make the web search useful.

Search engines like AltaVista[1], Google[2] and AllTheWeb[3] are usually based on a keyword matching inverse lookup of words and phrases. On top of their keyword matching, techniques such as detection of proper phrases (e.g. new york <new york>) and removal of stopwords or synonyms (e.g. how can i get information about X - Y).

There are techniques reduce the query into a term that is more likely to express the topic that is asked for, and in a suitable manner for a word-based or phrase-based inverse lookup, and thus improve precision of the search. For instance, a query like where can I find pizza but in new york will most likely have better precision in a word/phrase match intersection when rewritten into a form like "pizza but new york".

Difficult with this query rewriting occur when there are ambiguities in phrasing and non-phrasing disambiguation. For instance the query free computer wallpaper downloads will be rewritten into "free computer wallpaper downloads" if phrasing were done by a reference-language approach (which is a common approach) instead of more natural free "computer wallpaper" downloads.

In this paper we will describe how we use data mining in query logs and document corpora to derive information that can be used to segment queries into words and phrases with a number indicating consistency.

Copyright held by the author(s).

WWW-07, May 20-26, 2007, Banff, Alberta, Canada.

ACM 955.

2. MINING LOGS AND CORPORA

Query logs yield a highly interesting data that may be useful for various tasks. Whenever query content specific application (statistical modeling, generation of related queries or triggering relevant feedback) is considered, a segmentation of meaningful and recognizable phrases in each individual query remains one of its core parameters.

A sequence $S = w_1 \dots w_n$ (with $2 \leq n \leq 4$) of query tokens (words, numbers, special symbols) is considered a potentially meaningful phrase if the following conditions hold:

1. S is significantly frequent in all resources.

2. S has a "good" natural information.

Both above conditions make up a central criterion for the segmentation of queries. A consistency of a sequence $S = w_1 \dots w_n$, which is defined as a product of the global frequency of the segment $freq(S)$ and the natural information I between longest but complete subsequence of S :

$$cons(S) = freq(S) \cdot I(w_1 \dots w_n, w_1 \dots w_n) \quad (1)$$

It is assumed that consistency of a single token is equivalent to its frequency i.e. $cons(w_i) = freq(w_i)$.

The consistency value presented here is computed from a selected sample of our query logs where characteristics is approx. 400 million original query lines and 100 million lines in its normalized frequency log $Q_{normalized}$. Most of the operations related to the computation of consistency are carried out on $Q_{normalized}$ or on its representation.

For the sake of a brief characterization of $Q_{normalized}$ we split it into values according to the number of tokens in a line. Table 1 shows how many lines each subset of $Q_{normalized}$ consists of and how many new lines were in other subsets segments $S = w_1 \dots w_n$ (with $1 \leq n \leq 4$) are contained in each subset.

tokens in line	1	2	3	4	5	6	7	8	9	10
new lines	10	10	10	10	10	10	10	10	10	10
new lines	10	10	10	10	10	10	10	10	10	10

Table 1: Tokens and segment numbers in $Q_{normalized}$

The total number of 1-4 token segments that appear in $Q_{normalized}$ is estimated to $|S_{total}| \geq 544,197$. A minimum subset S_{subset} of S_{total} may be chosen to satisfy feasibility conditions on a single processing block (i.e. search node) as defined in [13].

• Query processing speed ≥ 5000 queries/second.

• disk access excluded - full database that keeps the segment S_{subset} in their connections must be large in memory.

tokens are not indexed but matched on a web search and phrases with sample. Zhai (1997) single-word symbols is for "bank terminated by bank". The reader current search engine does recognize the meaning in some way, by semantics; also demonstrates the reader the query "two is a number of possible and these can be excellent segmentations are.

of these interpretations also marks around the search engine to only matches. If, as seems pages about the large, on saws used by luring trees, then the first vol., a physical search Google does find the he second interpretation relevant pages displaying "two-man handsets,

and Computational Natural Languages

Query Segmentation for Web Search

- Segmente = Bedeutungsvolle Phrasen mit
 - signifikanter Frequenz im Corpus
 - hoher "Mutual Information"

$$\text{conn}(S) = \text{freq}(S) \cdot I(w_1 \dots w_{n-1}, w_2 \dots w_n)$$

- Berechnung des Connexity-Scores für alle 2^{n-1} Segmentierungen
- Sortierung nach max. kumulativen Scores

Query Segmentation for Web Search

$$\text{conn}(S) = \text{freq}(S) \cdot I(w_1 \dots w_{n-1}, w_2 \dots w_n)$$

$S_1 = \text{"new york" "times square" dance}$

- $\text{freq}(\text{"new york"}) = 7.500$
- $\text{freq}(\text{"times square"}) = 100$
- $I(\text{"new"}, \text{"york"}) = 0.04$
- $I(\text{"times"}, \text{"square"}) = 0.02$

$$\begin{aligned}\text{Score}(S_1) &= \text{conn}(\text{"new york"}) + \text{conn}(\text{"times square"}) \\ &= 300 + 2 \\ &= 302\end{aligned}$$

$S_2 = \text{"new york times" "square dance"}$

- $\text{freq}(\text{"new york times"}) = 3.400$
- $\text{freq}(\text{"square dance"}) = 200$
- $I(\text{"new york"}, \text{"york times"}) = 0.03$
- $I(\text{"square"}, \text{"dance"}) = 0.01$

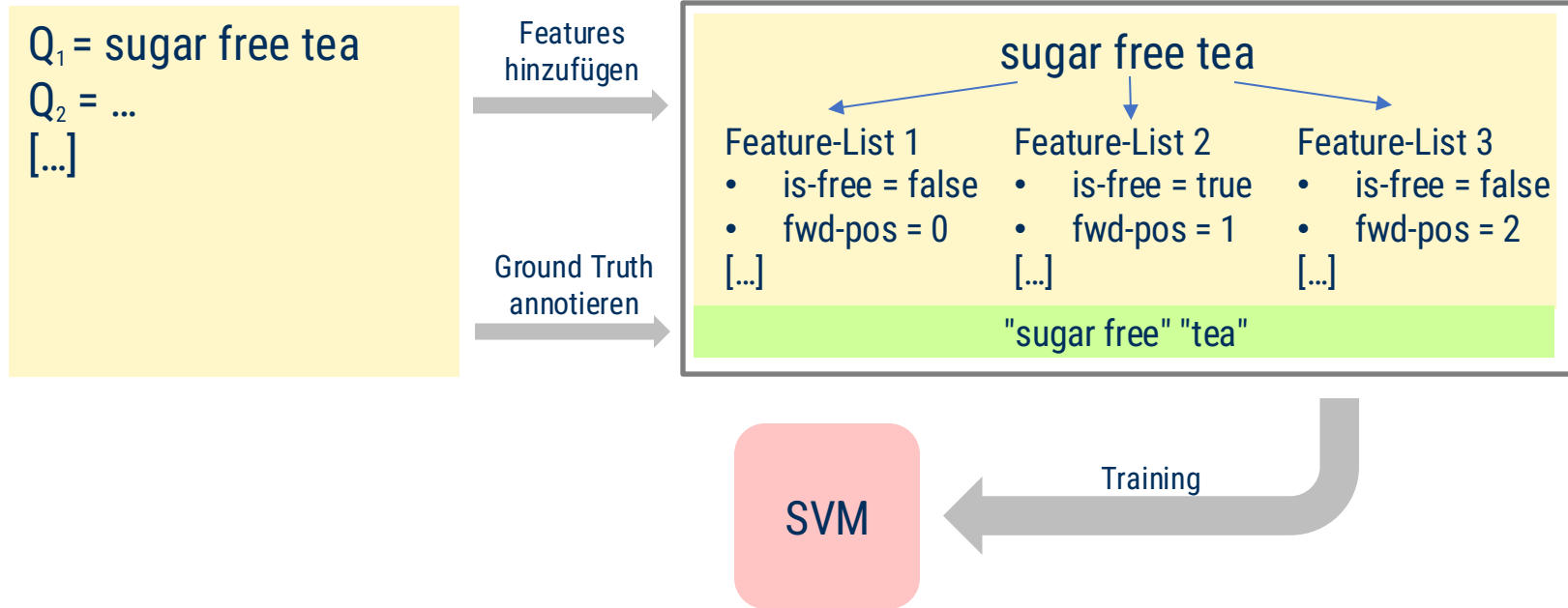
$$\begin{aligned}\text{Score}(S_2) &= \text{conn}(\text{"new york times"}) + \text{conn}(\text{"square dance"}) \\ &= 102 + 2 \\ &= 104\end{aligned}$$

Learning Noun Phrase Query Segmentation

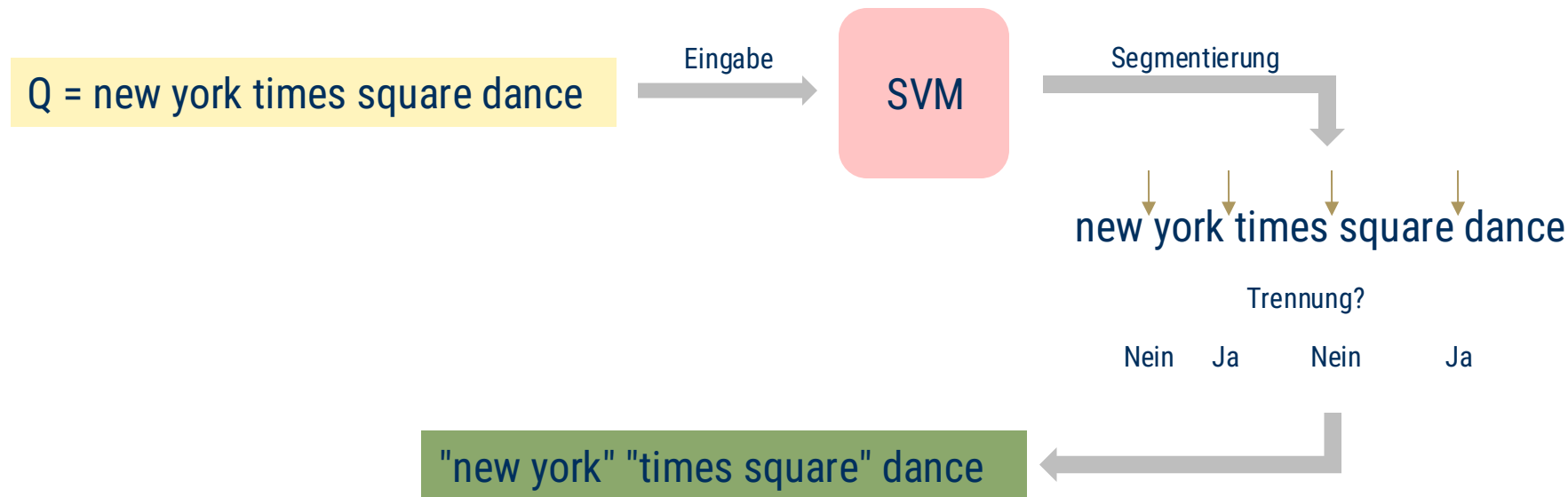
- Segmentation als Classification Task mittels
 - supervised Machine Learning
 - Support Vector Model (SVM)
- Trennung wird an jedem Breakpoint einer Query entschieden
- Einbezug des Kontext in Form eines Token-Fensters
 - 3 Token links/rechts von Breakpoint (falls vorhanden)
 - genannt "Decision Boundary"

↓
 $\{..., w_{L2}, w_{L1}, w_{L0}, w_{R0}, w_{R1}, w_{R2}, ...\}$

Learning Noun Phrase Query Segmentation



Learning Noun Phrase Query Segmentation



Learning Noun Phrase Query Segmentation

Annotation

- Aufteilung in Training, Validation und Test (je 500 Queries)
- "Ground Truth" von 3 Annotatoren manuell erstellt
- Agreement $\kappa = 0.69$

Naive Baselines

Ansatz	Seg.-Acc.	Qry.-Acc.
"always split"	0.44	0.04
"never split"	0.55	0.04

Learning Noun Phrase Query Segmentation

Decision Boundary

- Features für x_{L0} , x_{R0}
- Aufgeteilt in
 - Indikator F.
 - Statistische F.

Context

- Wie Decision Boundary
- Features für ganzes Fenster x_{L2} bis x_{R2}

Dependency

- Nimmt Rücksicht auf Einfluss zwischen entfernten Token
- Count von
 - x_{L0} und x_{R1}
 - x_{L1} und x_{R0}

Learning Noun Phrase Query Segmentation

Indikator	Beschreibung
is-free	Token x = "free"
fwd-pos	Position von Anfang
rev-pos	Position von Ende
[...]	

Statistik	Beschreibung
web-count	Häuf. von x im Web
Qcount-1	Häuf. von x in Query-Log
[...]	

Learning Noun Phrase Query Segmentation

Feature Type	Feature Span	Seg.-Acc.	Qry.-Acc.
MI	Decision Boundary	0.68	0.26
Basic	Decision Boundary	0.71	0.29
Basic	Decision Boundary, Context	0.80	0.52
Basic	Decision Boundary, Context, Dependency	0.81	0.53
All	Decision Boundary	0.84	0.57
All	Decision Boundary, Context	0.86	0.63
All	Decision Boundary, Context, Dependency	0.85	0.61

Takeaways & Fragen

- Query Segmentation: Grundlagen & Motivation
- Connexity-Score: Bedeutung & Berechnung
- Segmentation als Classification-Task
- Feature-Engineering und Training eines SVMs

Dankeschön!

Mutual Information

$$I(X; Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{(X,Y)}(x, y) \log \left(\frac{P_{(X,Y)}(x, y)}{P_X(x) P_Y(y)} \right)$$

$$\Leftrightarrow \log C(x_{L0}x_{R0}) + \log K - \log C(x_{L0}) - \log C(x_{R0})$$

- Ein Maß für die gegenseitige Abhängigkeit zweier Variablen
- Auch bekannt als "Information Gain"
- Benötigte Größen im IR Kontext:
 - rel. Wahrscheinlichkeiten von Token "x", "y" separat &
 - rel. Wahrscheinlichkeit von Token "x y" zusammen

Support Vector Machine (SVM)

- Überwacher, feature-basierter Klassifikationsalgorithmus
- Gilt für viele Probleme als guter Default-Ansatz
- Maximiert den Abstand zwischen den Trainingsinstanzen
 - Margin:
max. Abstand zwischen Support Vektoren
 - Support Vektoren:
Trainingsinstanzen am nächsten zur Margin

