SiLi Index: Data Structure for Fast Vector Space Searching

RASLAN 2019

Ondřej Herman Pavel Rychlý

Natural Language Processing Centre
Faculty of Informatics, Masaryk University

December 7, 2019

Contents

Dictionaries define words

beagle

Dictionaries define words

beagle
The beagle is a breed of small hound that is similar in appearance to the much larger foxhound. The beagle is a scent hound, developed primarily for hunting hare.

Single word definition

beagle

Knowledge for people and/or computers

beagle

Hypoallergenic No

Life expectancy 12 – 15 years

Weight Male: 10–11 kg, Female: 9–10 kg

Colors Lemon+White, Tri-color, White+Tan, Chocolate Tri, White+Chocolate, Orange+White, Red+White

Temperament Amiable, Determined, Intelligent, Even Tempered, Excitable, Gentle

Height Male: 36–41 cm, Female: 33–38 cm

Definition for computers: Word embeddings

High dimension (100-500) vector space
beagle 0.29608 -0.23048 -0.19875 -0.48953 -0.50818 -0.26509 0.0694 0.44668 0.56284 -0.19437 -0.42638 0.031915 0.19004 0.29487 0.12378 0.15903 -0.33378 0.25208 0.12684 -0.36154 -0.16752 0.02755 -0.28174 -0.34246 -0.64904 -0.16328 0.19421 0.0011659 0.21844 -0.2479 0.35647 0.073234 . . .

Definition for computers: Word embeddings

High dimension (100-500) vector space
beagle 0.29608 -0.23048 -0.19875 -0.48953 -0.50818 -0.26509 0.0694 0.44668 0.56284 -0.19437 -0.42638 0.031915 0.19004 0.29487 0.12378 0.15903 -0.33378 0.25208 0.12684 -0.36154 -0.16752 0.02755 -0.28174 -0.34246 -0.64904 -0.16328 0.19421 0.0011659 0.21844 -0.2479 0.35647 0.073234 . . .
dog 0.32606 -0.078246 -0.12248 -0.27463 -0.18233 -0.14209 0.19938 0.35211 0.35153 -0.13979 -0.65495 -0.34541 0.14729 -0.23703 -0.066852 0.29443 -0.23985 -0.15383 0.24648 -0.62856 -0.42847 -0.057002 0.15087 -0.066191 -0.92777 -0.17826 0.39743 -0.38183 0.30123 . . .

Definition for computers: Word embeddings

High dimension (100-500) vector space
beagle 0.29608 -0.23048 -0.19875 -0.48953 -0.50818 -0.26509 0.0694 0.44668 0.56284 -0.19437 -0.42638 0.031915 0.19004 0.29487 0.12378 0.15903 -0.33378 0.25208 0.12684 -0.36154 -0.16752 0.02755 -0.28174 -0.34246 -0.64904 -0.16328 0.19421 0.0011659 0.21844 -0.2479 0.35647 0.073234 . . .
dog 0.32606 -0.078246 -0.12248 -0.27463 -0.18233 -0.14209 0.19938 0.35211 0.35153 -0.13979 -0.65495 -0.34541 0.14729 -0.23703 -0.066852 0.29443 -0.23985 -0.15383 0.24648 -0.62856 -0.42847 -0.057002 0.15087 -0.066191 -0.92777 -0.17826 0.39743 -0.38183 0.30123 . . .
Computer: I see, they are close (0.823)

Operations

1. **Pairwise similarity** – given two elements of the vector space, quantify their similarity.
2. **k-nearest neighbor queries** – given an element of the vector space, retrieve k most similar elements.
3. **Analogy queries** – given three elements, a, $a*$, and b, retrieve k candidate elements $b*$ which satisfy the following criterion: a is to $a*$ as b is to $b*$.

Operations calculations

1. **Pairwise similarity** – easy, fast
2. **k-nearest neighbor queries** – pairwise similarity to every other element, then choose k best
3. **Analogy queries** – k-nearest neighbor to $v = b - a + a*$

Data size

Dimension	Datatype	Total size	Note
100	float32	2.540 GiB	
300	float32	7.620 GiB	
500	float32	12.700 GiB	
500	float16	6.350 GiB	non-native datatype, slow

Data size

Dimension	Datatype	Total size	Note
100	float32	2.540 GiB	
300	float32	7.620 GiB	
500	float32	12.700 GiB	
500	float16	6.350 GiB	non-native datatype, slow

20 queries per second by memory transfers alone

SiLi Index

Main idea – similarity lists

- many operations does not need vectors
- store k-nearest neighbors

Structure of SiLi Index

1. lexicon: list of words (implicit ID)
2. main record array: nearest neighbors for every word
3. mapping from numerical IDs to record array positions

k-nearest neighbor queries

- lexicon: word $\rightarrow$ ID
- index: ID $\rightarrow$ position in main file
- read list of nearest neighbors

k-nearest neighbor queries

- lexicon: word $\rightarrow$ ID
- index: ID $\rightarrow$ position in main file
- read list of nearest neighbors

lexicon + 1 memory/disk seek

Analogy Queries

$$argmax_{b^*}(sim(b^*, b - a + a^*))$$

$$argmax_{b^*}(b^* \cdot b - b^* \cdot a + b^* \cdot a^*))$$

- nearest neighbor lists for a, b, a^*
- set oprations: intersection + difference
- sort result

lexicon + 3 memory/disk seeks

Conclusion

- SiLi index is fast
- currently, we store 500 nearest neighbors
- more evaluation needed