Abstracts

1/23/2024

Say everything three times

Talks

- Say what you will say (promise)
- Say it (connect the dots)
- Say what you said (delivery)

Abstracts

- Lead Sentence
- Body
- Conclusion

Part of Speech Tagging (1988) https://aclanthology.org/A88-1019.pdf

- Lead
  - A program that tags each word in an input sentence with the most likely part of speech has been written.
- Body
  - The program uses a linear-time dynamic programming algorithm to find an assignment of parts of speech to words that optimizes the product of (a) lexical probabilities (probability of observing part of speech i given word i) and (b) contextual probabilities (probability of observing part of speech i given n following parts of speech).
- Conclusion
  - Program performance is encouraging; a 400-word sample is presented and is judged to be 99.5% correct

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding https://aclanthology.org/N19-1423.pdf

- Lead
  - We introduce a new language representation model called BERT,
  - which stands for Bidirectional Encoder Representations from Transformers.
- Body
  - Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial taskspecific architecture modifications. BERT is conceptually simple and empirically powerful.
- Conclusion
  - It obtains new state-of-the-art results on eleven natural language processing tasks,
    - including pushing the GLUE score to 80.5% (7.7% point absolute improvement),
    - MultiNLI accuracy to 86.7% (4.6% absolute improvement),
    - SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and
    - SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

Techniques for Automatically Correcting Words in Text https://dl.acm.org/doi/pdf/10.1145/146370.146380

- Lead
  - Research aimed at correcting words in text has focused on three progressively more difficult problems:
    - (1) nonword error detection;
    - (2) isolated-word error correction; and
    - (3) context-dependent word correction.
- Body
  - In response to the first problem, efficient pattern-matching and n-gram analysis techniques have been developed for detecting strings that do not appear in a given word list.
  - In response to the second problem, a variety of general and application-specific spelling correction techniques have been developed. Some of them were based on detailed studies of spelling error patterns.
  - In response to the third problem, a few experiments using natural-language-processing tools or statistical-language models have been carried out.
- Conclusion
  - This article surveys documented findings on spelling error patterns,
  - provides descriptions of various nonword detection and isolated-word error correction techniques,
  - reviews the state of the art of context-dependent word correction techniques, and
  - discusses research issues related to all three areas of automatic error correction in text.

https://people.csail.mit.edu/indyk/p117-andoni.pdf

Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions

by Alexandr Andoni and Piotr Indyk

Abstract

In this article, we give an overview of efficient algorithms for the approximate and exact nearest neighbor problem. The goal is to preprocess a dataset of objects (e.g., images) so that later, given a new query object, one can quickly return the dataset object that is most similar to the query. The problem is of significant interest in a wide variety of areas.

The goal of this article is twofold. In the first part, we survey a family of nearest neighbor algorithms that are based on the concept of *locality-sensitive hashing*. Many of these algorithm have already been successfully applied in a variety of practical scenarios. In the second part of this article, we describe a recently discovered hashing-based algorithm, for the case where the objects are points in the d-dimensional Euclidean space. As it turns out, the performance of this algorithm is provably near-optimal in the class of the locality-sensitive hashing algorithms.

1 Introduction

The *nearest neighbor* problem is defined as follows: given a collection of n points, build a data structure which, given any query point, reports the data point that is closest to the query. A particularly interesting and tice, they often provide little improvement over a linear time algorithm that compares a query to each point from the database. This phenomenon is often called "the curse of dimensionality."

In recent years, several researchers have proposed methods for overcoming the running time bottleneck by using approximation (e.g., [5, 27, 25, 29, 22, 28, 17, 13, 32, 1], see also [36, 24]). In this formulation, the algorithm is allowed to return a point whose distance from the query is at most c times the distance from the query to its nearest points; $c > 1$ is called the *approximation factor*. The appeal of this approach is that, in many cases, an approximate nearest neighbor is almost as good as the exact one. In particular, if the distance measure accurately captures the notion of user quality, then small differences in the distance should not matter. Moreover, an efficient approximation algorithm can be used to

Not-so-great leads

- Such and such is an important problem
- Restate the title:
  - This survey will survey such-and-such paper
- Introduce some irrelevant background material
  - (failing to advance the main point)
  - Assumption: there should be one (and only one) main point