UPPSALA
UNIVERSITET

Palindromes

Never odd or even

Shifei Chen

Uppsala University
Department of Linguistics and Philology
Master Programme in Language Technology
Master's Thesis in Language Technology, 30 ECTS credits

May 18, 2020

Supervisors:
Ali Basirat, Uppsala University

Abstract

The concept of *palindromes* is introduced, and some method for finding palindromes is developed.

Contents

Preface

I want to thank Donald Knuth for making TeX, without which I wouldn't have written this.

1 Introduction

Palindromes are fun. I've tried to find some. In Chapter 2 previous work is reviewed, and Chapter 3 is about my results.

2 Previous work

2.1 Word Embeddings

2.1.1 Representing Words in Vectors

In Natural Language Processing, people need to convert the natural representation of words into form that are more effieicent for computer to process. The idea started with statistical language modelling (**bengio2003neural**). In 2013, Mikolov, Chen, et al., 2013 introudced Word2Vec, which encapsules words and their latent information into vectors. Besides the benefit that it simplifies representation and storage of words for computers, it also enables the possibilities to calcualte word and their semantic meanings just as vectors.

Take an example vocabulary $V = \{\text{king}, \text{queen}, \text{man}, \text{woman}\}$, if we convert these words into vectors such as

$$\begin{aligned}\vec{k} &= \text{vec}(\text{king})\\ \vec{q} &= \text{vec}(\text{queen})\\ \vec{m} &= \text{vec}(\text{man})\\ \vec{w} &= \text{vec}(\text{woman})\end{aligned}$$

We could have an equation of

$$\vec{q} = \vec{k} - \vec{m} + \vec{w} \tag{2.1}$$

It is meaningful from both the mathmatical prospective and the linguistic prospective. The latter can be illustrated by Figure 2.1 in a vector space that contains these four vectors. In addition, the two cosine similarity values of vectors $\vec{k}$ and $\vec{q}$, and of $\vec{m}$ and $\vec{w}$ should also be close, as the angles between each two vectors are about the same.

To turn words into vectors, one could use simple one-hot encoding. Like in the example above we could make $\vec{k} = [1, 0, 0, 0]$. But these one-hot vectors can merely

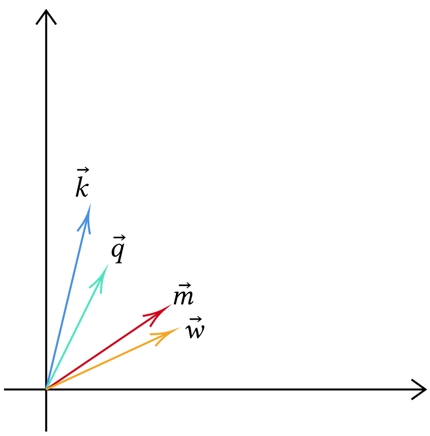

Figure 2.1: Illustraion of a vector space where Equation 2.1 exists.

capture any latent semantic meanings between different words. Recent vectorized word representations, or word embeddings, were learning through neural networks, such as Word2Vec which learns word embeddings through Continuous Bag of Words (CBOW) model and Continuous Skip-gram model (Mikolov, Sutskever, et al., 2013). Both of the models have a projection layer between the input and the output layer. The difference is that the CBOW model predicts the target word from the multiple given context words, while the Skip-gram model predicts the context words from one given center word.

3 Results

Bibliography

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient Estimation of Word Representations in Vector Space" (Jan. 2013). eprint: 1301.3781. URL: https://arxiv.org/pdf/1301.3781.pdf.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality" (Oct. 2013). eprint: 1310.4546. URL: https://arxiv.org/pdf/1310.4546.pdf.