Chapter 1

Introduction

It has been a continual surprise, that simple combinations of embeddings performs so well for a variety of tasks in natural language processing.

1.1 Works Within this Thesis

- **??** examines a variety of sentence embedding methods on how well they create a space that can be partitioned according to the true partitioning of the sentences according to their meaning. It finds the surprising result that the linear combinations of embeddings (SOWE and Mean of Word Embeddings) significantly outperform the more sophisticated methods.
- **??** proposes as method to extract the original bag of words from a sum of word embeddings. Thus placing a bound on the information lost in the transformation of BOW to SOWE. This is done via a simple greedy algorithm with a correction step.
- **??** extends this, investigating to what extend can the BOW generated by **??** be reordered to determine the original sentence. This placing a bound on the information lost from Sentence to SOWE. This is carried out by transforming the word ording problem into a mixed integer programming problem of finding the most likely sequence according to a trigram language model.
- **??** considers the use of word embeddings as features to a binary classifier to detect named entities are the Point of View character. The mean of the word embeddings of the adjacent words for each occurrence of each named-entity is used as a feature vector for a binary classifier. We contrast it's performance to that of hand-engineered lexical features. It is concluded that the difference in performance is primarily related to dimensionality and the amount of training data for the binary task.
- **??** also considers the use of word embeddings as features, here to predict probability distributions and point estimates for colors given the color name. A Sum of Word Embeddings as an input layer is contrasted with processing the input (as word embeddings) through an RNN or CNN, as well as to a baseline. While this is a task where word order is significant, ("Brownish Green" is a slightly different shade to "Greenish Brown") never the less SOWE do very well, likely do to the ease of training.
- **??** defines a method to generate new word sense embeddings from an existing set of sense embeddings by using their a linear combination of the existing embeddings with weight based on how likely those senses are to occur in an example sentence. It finds the sense embeddings generated this way are more useful for similarity comparison, but not competitive for use in word sense disambiguation.

1.2 Some Math

1.2.1 Word Embeddings

Given a word represented by an integer w, from a vocabulary $\mathbb{V} \subset \mathbb{Z}$, and a matrix of embeddings, represented as C: its embedding can be found by slicing out the wth column: $C_{:,w}$. For $\tilde{e}_w$ the elementary unit vector, i.e. the one-hot vector representation of w it can be seen that the word embedding can be represented as the product of the embedding matrix with the one-hot vector.

$$C_{:,w} = C\,\tilde{e}_w$$

1.2.2 Sum of Word Embeddings

For some sequence of (not necessarily unique words) words $\mathcal{W} \in \mathbb{V}^{\mathbb{Z}_0}$ represented $\mathcal{W} = \left(w^1, w^2, \ldots, w^n\right)$, where w^i is an integer representing which word the ith word is.

The sum of word embeddings (SOWE) representation is written as:

$$\sum_{i=1}^{i=n} C_{:,w^i}$$

The bag of word (BOW) representation is written as a vector from $\mathbb{Z}^{|\mathbb{V}|}$.

$$\tilde{x} = \sum_{i=1}^{i=n} \tilde{e}_{w^i}$$

,

Using this the sum of word embeddings can be seen to be the product of the embedding matrix with a BOW vector.

$$\sum_{i=1}^{i=n} C_{:,w^i} = \sum_{i=1}^{i=n} C\,\tilde{e}_{w^i} = C \sum_{i=1}^{i=n} \tilde{e}_{w^i} = C\,\tilde{x}$$

1.2.3 Mean of Word Embeddings

The mean of word embeddings representation is written as:

$$\frac{1}{n} \sum_{i=1}^{i=n} C_{:,w^i}$$

Note that n is equal to the element-wise sum of the BOW vector (x), i.e. to its l1-norm:

$$n = \|\tilde{x}\|_1 = \sum_{\forall j \in \mathbb{V}} \tilde{x}_j$$

Thus the mean of word embeddings can be seen as the product of the embedding matrix with the l1-normalized BOW vector.

$$\frac{1}{n} \sum_{i=1}^{i=n} C_{:,w^i} = \frac{1}{n} \sum_{i=1}^{i=n} C\,\tilde{e}_{w^i} = \frac{1}{n} C \sum_{i=1}^{i=n} x = C\,\frac{\tilde{x}}{\|\tilde{x}\|_1}$$

1.2.4 Linear Combination of Embeddings

The full generalisation of this is that any linear combination of embeddings can be seen as product of the embedding matrix, the a weighted bag of words.

A weighting function for linear combination scheme can be defined, mapping from a given bag of words, and a word, to the weighting of that word. $\alpha : \mathbb{Z}^{|\mathbb{V}|} \times \mathbb{V} \to \mathbb{R}$.

(For example for the mean of word embeddings $\alpha(\tilde{x}, w) = \frac{1}{\|\tilde{x}\|_1}$.)

From the weighting function, we can evaluated it for a given BOW, for each word in the vocabulary to define a weighting vector $\tilde{\alpha}^{\tilde{\mathrm{x}}}$:

$$\tilde{\alpha}^{\tilde{\mathrm{x}}} = \left[\alpha(\tilde{x}, w)\right]_{w \in \mathbb{V}}$$

Using $\odot$ as the Hadamard (i.e. element-wise) product, we can thus write:

$$\sum_{i=1}^{i=n} \alpha(\tilde{x}, w^i) C_{:,w^i} = C \sum_{i=1}^{i=n} \alpha(\tilde{x}, w^i) \tilde{e}_{w^i} = C\,\left(\tilde{\alpha}^{\tilde{\mathrm{x}}} \odot \tilde{x}\right)$$