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. At first glance, such simple methods capturing only unordered word use should have little capacity in the rich and highly structured nature of human language as we linguistically understand it. However at a second glance, similar surface information has been used in information retrieval with great success since the inception of the field (**maron1961automatic**). Linear combinations of embeddings can be considered as a dimensionality reduction of a bag of words, with various weightings. Dimensionality reduction can be characterised as finding the best lower dimensional representation of an input according to some quality criterion. In the case of word embeddings, that quality criterion is generally along the lines the ability to predict the co-occurring words – an salient quality of lexical semantics. As such, linear combinations of embeddings take bag of words which is a strong surface form representation, take reduce it to a dense representation that captures lexical semantics.

acl2018bleuopposedmeaning found that taking a mean of word embeddings outperformed almost all their more sophisticated machine-translation-based sentence representations when used on classification and paraphrase detection tasks. This is not to say that linear combinations of embeddings are ideal models for all tasks. They clearly can not truly handle all the complexities of language. But rather that the occurrence of the complexities they can not handle is rarer in practice in many tasks than is often expected.

ac2018probingsentencevectors constructed 10 probing tasks to isolate the some of the information captured by sentence representations. They found the strong performance of averaged word embeddings on sentence level tasks to be striking. They attribute it to the sentence level information being redundantly encoded in the word-forms: the surface level information is surprisingly useful for what at first looks like a sophisticated tasks. With the exception of their word-content task, they did find more sophisticated models able to perform better the the averaged word embeddings. However, when correlating the performance of their probing task against real world tasks, they found that the word-content probing task was by far the most positively correlated with the real word tasks. This makes it clear how valuable surface information is in practical tasks.

In the work presented this dissertation, we find that that even in tasks where it would seem that non-surface information incorporating word-order is required, we find in practice other issues cause the more powerful models that are (theoretically) able to handle these situations correctly may them to be never-the-less outperformed. This is particularly the case where the theoretical improvement from incorporating this information is small, relative to the practical complexity of the techniques required to leverage it. Such a case where where word order matters but the error from ignoring it is small, is particular illustrated in **??**.

At a high-level the success of these techniques comes down to most human language being easy to understand and simple. This expectation of language being easily understood is highlight by (**grice1975logic**), which brings the expectation the communication is conducted following the cooperative principle The overall supermaxim for Grice's cooperative principle is the speakers should "be perspicuous" or more perspicuously, should use speech that is clearly expressed and easily understood. The particular relevant maxims within the principle are: the maxim of quantity, that one should should make contributions no more nor less informative than required; and the maxim of manner: to avoid ambiguity and obscurity of expression, and to make contributions that are brief and orderly. While Grice originally proposed these are exceptions upon conversation, the general principle applies more broadly to natural language communication. This general principle being that language used is normally expected to be understood easily – thus fulfilling the goal of communicating.

Adversarial examples are reasonably easy to construct. An adversarial example to a linear combination of word embeddings is any text where the word order significantly effects that meaning; and where multiple possible word orders exist. For such an adversary to be significant, both word orders must be reasonably likely to occur. However; such cases are rarer than one might expect, as was found indirectly in ??. Particularly when punctuation is included, which it reasonably can be as a token embedding. As such, while these cases certainly exist, we find that for real applications they are sufficiently rare that the simplicity of the linear combinations of embeddings type approach can work very well.

The when applied in sentence/phrase representation contexts, such as discussed in ??, and ?? this gives support to the notion that word order is often not a very significant feature in determining meaning. It seems clear that word order, and other factors of linguistic structure must contribute to the meaning of the phrase. However, our result suggest that it is often in a minor way, and that for many tasks these linear combinations are superior due to their simplicity and effectiveness. While taking into account greater linguistic structure may be the key to bridging the between ”almost perfect” and ”true perfection”, the current state of the field for many tasks has not reached ”almost perfect”, and as such simpler methods still form an important part.

To further understand the relationship between SOWE and BOW, and the extent to which word order matters the capacity to reverse the conversion from phrase to SOWE in investigated in ?? and ??. The results in ?? show that it is indeed largely possible to reconstruct bags of words from SOWE, suggesting that when considered as a a dimensionality reduction technique SOWE does not lose much information. This is extended in ?? to order those bags of words back to sentences via a simple tri-gram language model. This had some success at outright reconstructing the sentences. This highlights the idea that for many bags of words (which can be reconstructed form a sum of word embeddings) there may truly be only one reasonable sentence from which they might have come. This would explain why SOWE, and BOW, ignorance of word order does not prevent them from being useful representation of sentence.

The successes of the sums of word embeddings discussed in ??, and ?? leads us to consider other uses of linear combinations for representation. ?? and ?? consider tasks well outside of phrase representation where the order clearly does not matter.

On the complexity of models. One of the attractive features of these linear combinations is there simplicity This is true both in an implementation sense, and in the sense of gradient descent. For example, the vanishing gradient problem in deep networks, especially RNNs and RvNNs simply does not exist for a sum of word embeddings due to it not being as an input structure. This in contrast to RNNs which are deep in time, and RvNNs which are deep in structure. Deep networks can be placed upon the input processing as represented by a RNN, RvNN or linear combination of embeddings, but for the RNN, and RvNN the network is already depth even with only one hidden layer on top.

TODO: INSERT RNN FAIL PAPER REFERENCE HERE

1.1 Works Within this dissertation

To break this down by structure, we investigate using linear combinations of embeddings to represent:

Sentences ??

Phrases ??

Word Senses ??

Contexts ??

With ???? providing supporting motivational evidence of the capacity of the models with particular reference to the use in sentences.

1.1.1 ??

We begin by examining methods for representing sentences. Sentences are a fundamental unit of communication – a sentence is a single complete idea.

The core goal is to determine if an sentence embedding clearly separated the different ideas. Paraphrases are defined by a bidirectional entailment relationship between two sentences. This is a equivalence relationship, it thus gives rise to a partitioning of all sentences in natural language space. If a sentence embedding is of high quality, it will be easy to define a corresponding partitioning of the embedding space. One way to determine how easy it is to define the corresponding partitioning it to attempt to do just that as a supervised classification task using a weak classifier. A weak classifier, such as the linear SVM we used, is required as a more powerful classifier (such as a deep neural network) could learn arbitary transforms. The classification task is to take in a sentence embedding and predict which group of paraphrases it belongs to. Where the target paraphrase group is defined using other paraphrases with the same meaning as the candidate.

Under this course of evaluation it was found that the sum and mean of word embeddings performed very well as a sentence representation. These LCOWEs were the best performing models under evaluation. They were closely followed by the bag of words, which is advantaged by being much higher dimensionality than other models. The LCOWEs outperform the bag of words as they also capture synonyms and other features of lexical relatedness. Slightly worse than the bag of words was the bag of words with PCA dimensionality reduction to 300 dimensions. This confirms our expectation that LCOWEs are a better form of dimensionality reduction for preserving meaning from a bag of words than PCA.

A limitation of this work is that it does not include the examination of any encoder-decoder based methods, such as Skip-Thought (**DBLP:journals/corr/KirosZSZTUF15**), or machine translation models. Another limmitation of the work is that the URAE use a pretrained model with only 200 dimensions, rather than 300 dimensions as was used in the other evaluations.

Paraphrases provide one source of groundingm for evaluation of sentences. Color names are a subset of short phrases which also have a ground truth for meaning – the color. They are thus useful for evaluating the performance of LICOWE on short phrases.

1.1.2 ??

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 due to the ease of training.

1.1.3 ??

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.

1.1.4 ??

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.

A limitation of this study was that it did not perform the evaluation on state of the art word-sense embeddings. As such, while it's comparisons between these embeddings are valid and shows improvement, on the similarity tasks, it can not be readily compared to the current state-of-the-art on that task.

1.1.5 ??

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.

1.1.6 ??

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.

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:

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

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

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

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 $\alpha(\tilde{x}, w) = \frac{1}{\left\|\tilde{x}\right\|_1}$.)

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

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

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

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