Learning Neural Word Salience Scores

Krasen Samardzhiev **Andrew Gargett** **Danushka Bollegala**
Department of Computer Science, University of Liverpool
Science and Technology Facilities Council.
krasensam@gmail.com, danushka@liverpool.ac.uk
andrew.gargett@stfc.ac.uk

Abstract

Measuring the salience of a word is an essential step in numerous NLP tasks. Heuristic approaches such as tfidf have been used so far to estimate the salience of words. We propose *Neural Word Salience* (NWS) scores, unlike heuristics, are learnt from a corpus. Specifically, we learn word salience scores such that, using pre-trained word embeddings as the input, can accurately predict the words that appear in a sentence, given the words that appear in the sentences preceding or succeeding that sentence. Experimental results on sentence similarity prediction show that the learnt word salience scores perform comparably or better than some of the state-of-the-art approaches for representing sentences on benchmark datasets for sentence similarity, while using only a fraction of the training and prediction times required by prior methods. Moreover, our NWS scores positively correlate with psycholinguistic measures such as concreteness, and imageability implying a close connection to the salience as perceived by humans.

1 Introduction

Humans can easily recognise the words that contribute to the meaning of a sentence (i.e. content words) from words that serve only a grammatical functionality (i.e. functional words). For example, functional words such as *the, an, a* etc. have limited contributions towards the overall meaning of a document and are often filtered out as *stop words* in information retrieval systems (Salton and Buckley, 1983). We define the *salience* $q(w)$ of a word w in a given text T as the semantic contribution made by w towards the overall meaning of T. If we can accurately compute the salience of words, then we can develop better representations of texts that can be used in downstream NLP tasks such as similarity measurement (Arora et al., 2017) or text (e.g. sentiment, entailment) classification (Socher et al., 2011).

As described later in section 2, existing methods for detecting word salience can be classified into two groups: (a) lexicon-based filtering methods such as stop word lists, or (b) word frequency-based heuristics such as the popular term-frequency inverse document frequency (tfidf) (Jones, 1972) measure and its variants. Unfortunately, two main drawbacks can be identified in common to both stop words lists and frequency-based salience scores.

First, such methods do not take into account the semantics associated with individual words when determining their salience. For example, consider the following two adjacent sentences extracted from a newspaper article related to the visit of the Japanese Prime Minister, *Shinzo Abe*, to the White House in Washington, to meet the US President *Donald Trump*.

(a) *Abe visited Washington in February and met Trump in the White House.*

(b) *Because the trade relations between US and Japan have been fragile after the recent comments by the US President, the Prime Minister's visit to the US can be seen as an attempt to reinforce the trade relations.*

In Sentence (a), the Japanese person name *Abe* or American person name *Trump* would occur less in a corpus than the US state name *Washington*. Nevertheless, for the main theme of this sentence, *Japanese Prime minister met US President*, the two person names are equally important as the location they met. Therefore, we must look into the semantics of the individual words when computing their saliences.

Second, words do not occur independently of one another in a text, and methods that compute

*Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 33–42
New Orleans, June 5-6, 2018.

word salience using frequency or pre-compiled stop words lists alone do not consider the contextual information. For example, the two sentences (a) and (b) in our previous example are extracted from the same newspaper article and are adjacent. The words in the two sentences are highly related. For example, *Abe* in sentence (a) refers to the *Prime Minister* in sentence (b), and *Trump* in sentence (a) is refers to the *US President* in sentence (b). A human reader who reads sentence (a) before sentence (b) would expect to see some relationship between the topic discussed in (a) and that in the next sentence (b). Unfortunately, methods that compute word salience scores considering each word independently from all other words in near by contexts, ignore such proximity relationships.

To overcome the above-mentioned disfluencies in existing word salience scores, we propose an unsupervised method that first randomly initialises word salience scores, and subsequently updates them such that we can accurately predict the words in local contexts. Specifically, we train a two-layer neural network where in the first layer we take pre-trained word embeddings of the words in a sentence S_i as the input and compute a representation for S_i (here onwards referred to as a *sentence embedding*) as the *weighted average* of the input word embeddings. The weights correspond to the word salience scores of the words in S_i. Likewise, we apply the same approach to compute the sentence embedding for the sentence S_{i-1} preceding S_i and S_{i+1} succeeding S_i in a sentence-ordered corpus. Because S_{i-1}, S_i and S_{i+1} are adjacent sentences, we would expect the sentence pairs (S_i, S_{i-1}) and (S_i, S_{i+1}) to be topically related.[1]

We would expect a high degree of cosine similarity between $\boldsymbol{s}_i$ and $\boldsymbol{s}_{i-1}$, and $\boldsymbol{s}_i$ and $\boldsymbol{s}_{i+1}$, where boldface symbols indicate vectors. Likewise, for a randomly selected sentence $S_j \notin \{S_{i-1}, S_i, S_{i+1}\}$, the expect similarity between S_j and S_i would be low. We model this as a supervised similarity prediction task and use backpropagation to update the word salience scores, keeping word embeddings fixed. We refer to the word salience scores learnt by the proposed method as the *Neural Word Salience* (NWS) scores. We will use the contextual information of a word to learn its salience. However, once learnt, we consider salience as a property of a word that holds independently of its context. This enables us to use the same salience score for a word after training, without having to modify it considering the context in which it occurs.

Several remarks can be made about the proposed method for learning NWS scores. First, we do *not* require labelled data for learning NWS scores. Although we require semantically similar (positive) and semantically dissimilar (negative) pairs of sentences for learning the NWS scores, both positive and negative examples are automatically extracted from the given corpus. Second, we use pre-trained word embeddings as the input, and do *not* learn the word embeddings as part of the learning process. This design choice differentiates our work from previously proposed sentence embedding learning methods that jointly learn word embeddings as well as sentence embeddings (Hill et al., 2016; Kiros et al., 2015; Kenter et al., 2016). Moreover, it decouples the word salience score learning problem from word or sentence embedding learning problem, thereby simplifying the optimisation task and speeding up the learning process.

We use the NWS scores to compute sentence embeddings and measure the similarity between two sentences using 18 benchmark datasets for semantic textual similarity in past SemEval tasks (Agirre et al., 2012). Experimental results show that the sentence similarity scores computed using the NWS scores and pre-trained word embeddings show a high degree of correlation with human similarity ratings in those benchmark datasets. Moreover, we compare the NWS scores against the human ratings for psycholinguistic properties of words such as arousal, valence, dominance, imageability, and concreteness. Our analysis shows that NWS scores demonstrate a moderate level of correlation with concreteness and imageability ratings, despite not being specifically trained to predict such psycholinguistic properties of words.

2 Related Work

Word salience scores have long been studied in the information retrieval community (Salton and

[1] S_{i-1} and S_{i+1} could also be topically related and produce a positive training examples in some cases. However, they are non-adjacent and possibly less related compared to adjacent sentence pairs. Because we have an abundant supply of sentences, and we want to reduce label noise in positive examples, we do not consider (S_{i-1}, S_{i+1}) as a positive example.

Buckley, 1983). Given a user query described in terms of one or more keywords, an information retrieval system must find the most relevant documents to the user query from a potentially large collection of documents. Word salience scores based on term frequency, document frequency, and document length have been proposed such as tfidf and BM25 (Robertson, 1997).

Our proposed method learns word salience scores by creating sentence embeddings. Next, we briefly review such sentence embedding methods and explain the differences between the sentence embedding learning problem and word salience learning problem.

Sentences have a syntactic structure and the ordering of words affects the meaning expressed in the sentence. Consequently, compositional approaches for computing sentence-level semantic representations from word-level semantic representations have used numerous linear algebraic operators such as vector addition, element-wise multiplication, multiplying by a matrix or a tensor (Blacoe and Lapata, 2012; Mitchell and Lapata, 2008).

Alternatively to applying nonparametric operators on word embeddings to create sentence embeddings, recurrent neural networks can learn the optimal weight matrix that can produce an accurate sentence embedding when repeatedly applied to the constituent word embeddings. For example, skip-thought vectors (Kiros et al., 2015) use bi-directional LSTMs to predict the words in the order they appear in the previous and next sentences given the current sentence. Although skip-thought vectors have shown superior performances in supervised tasks, its performance on unsupervised tasks has been sub-optimal (Arora et al., 2017). Moreover, training bi-directional LSTMs from large datasets is time consuming and we also need to perform LSTM inference in order to create the embedding for unseen sentences at test time, which is time consuming compared to weighted addition of the input word embeddings. FastSent (Hill et al., 2016) was proposed as an alternative lightweight approach for sentence embedding where a softmax objective is optimised to predict the occurrences of words in the next and the previous sentences, ignoring the ordering of the words in the sentence.

Surprisingly, averaging word embeddings to create sentence embeddings has shown comparable performances to sentence embeddings that are learnt using more sophisticated word-order sensitive methods. For example, (Arora et al., 2017) proposed a method to find the optimal weights for combining word embeddings when creating sentence embeddings using unigram probabilities, by maximising the likelihood of the occurrences of words in a corpus. Siamese CBOW (Kenter et al., 2016) learns word embeddings such that we can accurately compute sentence embeddings by averaging the word embeddings. Although averaging is an order insensitive operator, (Adi et al., 2016) empirically showed that it can accurately predict the content and word order in sentences. This can be understood intuitively by recalling that words that appear between two words are often different in contexts where those two words are swapped. For example, in the two sentences "*Ostrich* is a large *bird* that lives in Africa" and "Large *birds* such as *Ostriches* live in Africa", the words that appear in between *ostrich* and *bird* are different, giving rise to different sentence embeddings even when sentence embeddings are computed by averaging the individual word embeddings. Instead of considering all words equally for sentence embedding purposes, attention-based models (Hahn and Keller, 2016; Yin et al., 2016; Wang et al., 2016) learn the amount of weight (attention) we must assign to each word in a given context.

Our proposed method for learning NWS scores is based on the prior observation that averaging is an effective heuristic for creating sentence embeddings from word embeddings. However, unlike sentence embedding learning methods that do not learn word salience scores (He and Lin, 2016; Yin

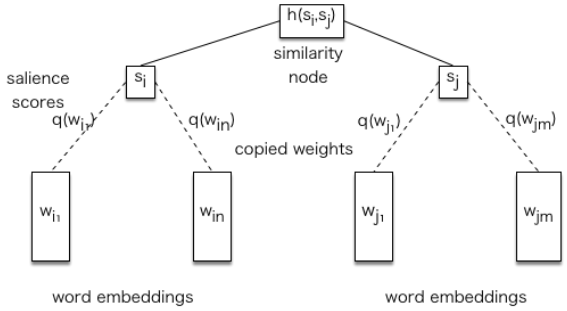


Figure 1: Overview of the proposed neural word salience learning method. Given two sentences (S_i, S_j), we learn the salience scores of words $q(w)$ such that we can predict the similarity between the two sentences using their embeddings $\boldsymbol{s}_i$, $\boldsymbol{s}_j$. Difference between predicted similarity and actual label is considered as the error and its gradient is backpropagated through the network to update $q(w)$.

et al., 2016) , our goal in this paper is to learn word salience scores and not sentence embeddings. We compute sentence embeddings only for the purpose of evaluating the word salience scores we learn. Moreover, our work differs from Siamese CBOW (Kenter et al., 2016) in that we do not learn word embeddings but take pre-trained word embeddings as the input for learning word salience scores. NWS scores we learn in this paper are also different from the salience scores learnt by (Arora et al., 2017) because they do not constrain their word salience scores such that they can be used to predict the words that occur in adjacent sentences.

3 Neural Word Salience Scores

Let us consider a vocabulary $\mathcal{V}$ of words $w \in \mathcal{V}$. For the simplicity of exposition, we limit the vocabulary to unigrams but note that the proposed method can be used to learn salience scores for arbitrary length n-grams. We assume that we are given d-dimensional pre-trained word embeddings $\boldsymbol{w} \in \mathbb{R}^d$ for the words in $\mathcal{V}$. Let us denote the NWS score of w by $q(w) \in R$. We learn $q(w)$ such that the similarity between two adjacent sentences $\mathcal{S}_i$ and $\mathcal{S}_{i-1}$, or $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ in a sentence-ordered corpus $\mathcal{C}$ is larger than that between two non-adjacent sentences $\mathcal{S}_i$ and $\mathcal{S}_j$, where $j \notin \{i-1, i, i+1\}$. Let us further represent the two sentence $\mathcal{S}_i = \{w_{i1}, \ldots, w_{in}\}$ and $\mathcal{S}_j = \{w_{j1}, \ldots, w_{jm}\}$ by the sets of words in those sentences. Here, we assume the corpus to contain sequences of ordered sentence such as in a newspaper article, a book chapter or a blog post.

The neural network we use for learning $q(w)$ is shown in Figure 1. The first layer computes the embedding of a sentence $\mathcal{S}$, $\boldsymbol{s} \in \mathbb{R}^d$ using Equation 1, which is the weighted-average of the individual word embeddings.

$$\boldsymbol{s} = \sum_{w \in \mathcal{S}} q(w)\boldsymbol{w} \tag{1}$$

We use (1) to compute embeddings for two sentences $\mathcal{S}_i$ and $\mathcal{S}_j$ denoted respectively by $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$. Here, the same set of salience scores $q(w)$ are used for computing both $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, which resembles a Siamese neural network architecture.

The root node computes the similarity $h(\boldsymbol{s}_i, \boldsymbol{s}_j)$ between two sentence embeddings. Different similarity (alternatively dissimilarity or divergence) functions such as cosine similarity, ℓ_1 distance, ℓ_2 distance, Jenson-Shannon divergence etc. can be used as h. As a concrete example, here we use softmax of the inner-products as follows:

$$h(\boldsymbol{s}_i, \boldsymbol{s}_j) = \frac{\exp\left(\boldsymbol{s}_i{}^\top \boldsymbol{s}_j\right)}{\sum_{\mathcal{S}_k \in \mathcal{C}} \exp\left(\boldsymbol{s}_i{}^\top \boldsymbol{s}_k\right)} \tag{2}$$

Ideally, the normalisation term in the denominator in the softmax must be taken over all the sentences $\mathcal{S}_k$ in the corpus (Andreas and Klein, 2015). However, this is computationally expensive in most cases except for extremely small corpora. Therefore, following noise-contrastive estimation (Gutmann and Hyvärinen, 2012), we approximate the normalisation term using a randomly sampled set of K sentences, where K is typically less than 10. Because the similarity between two randomly sampled sentences is likely to be smaller than, for example, two adjacent sentences, we can see this sampling process as randomly sampling *negative* training instances from the corpus.

For two sentences $\mathcal{S}_i$ and $\mathcal{S}_j$ we consider them to be similar (positive training instance) if $j \in \{i-1, i+1\}$, and denote this by the target label $t = 1$. On the other hand, if the two sentences are non-adjacent (i.e. $j \notin \{i-1, i+1\}$), then we consider the pair $(\mathcal{S}_i, \mathcal{S}_j)$ to form a negative training instance, and denote this by $t = 0$.[2] This assumption enables us to use a sentence-ordered corpus for selecting both positive and negative training instances required for learning NWS scores.

Specifically, the model is trained using the two adjacent sentences to $\mathcal{S}_i$ - $\{i-1, i+1\}$ as positive examples, and K=2 negative examples not in $\{i-1, i+1\}$. These are sampled from the whole text corpus using a uniformly. Similar to (Kenter et al., 2016), we found that increasing the number of negative examples increases the training time, but does not have a significant impact on model accuracy.

Using t and $h(\boldsymbol{s}_i, \boldsymbol{s}_j)$ above, we compute the cross-entropy error $E(t, (\mathcal{S}_i, \mathcal{S}_j))$ for an instance $(t, (\mathcal{S}_i, \mathcal{S}_j))$ as follows:

$$E(t, (\mathcal{S}_i, \mathcal{S}_j)) = t \log\left(h(\boldsymbol{s}_i, \boldsymbol{s}_j)\right) + (1-t) \log\left(1 - h(\boldsymbol{s}_i, \boldsymbol{s}_j)\right) \tag{3}$$

Next, we backpropagate the error gradients via the network to compute the updates as follows:

$$\frac{\partial E}{\partial q(w)} = \frac{(t - h(\boldsymbol{s}_i, \boldsymbol{s}_j))}{h(\boldsymbol{s}_i, \boldsymbol{s}_j)(1 - h(\boldsymbol{s}_i, \boldsymbol{s}_j))} \frac{\partial h(\boldsymbol{s}_i, \boldsymbol{s}_j)}{\partial q(w)} \tag{4}$$

[2] It is possible in theory that two non-adjacent sentences could be similar, but the likelihood of this event is small and can be safely ignored in practice.

Here, we drop the arguments of the error and simply write it as E to simplify the notation. To compute $\frac{\partial h(\boldsymbol{s}_i, \boldsymbol{s}_j)}{\partial q(w)}$ let us define

$$g(\boldsymbol{s}_i, \boldsymbol{s}_j) = \log\left(h(\boldsymbol{s}_i, \boldsymbol{s}_j)\right) \tag{5}$$

From which we have,

$$\frac{\partial h(\boldsymbol{s}_i, \boldsymbol{s}_j)}{\partial q(w)} = h(\boldsymbol{s}_i, \boldsymbol{s}_j)\frac{\partial g(\boldsymbol{s}_i, \boldsymbol{s}_j)}{\partial q(w)}. \tag{6}$$

We can then compute $\frac{\partial g}{\partial q(w)}$ as follows:

$$\mathcal{I}[w \in \mathcal{S}_i]\boldsymbol{w}^\top \boldsymbol{s}_j + \mathcal{I}[w \in \mathcal{S}_j]\boldsymbol{w}^\top \boldsymbol{s}_i \tag{7}$$

$$-\log(\sum_k \exp\left({\boldsymbol{s}_i}^\top \boldsymbol{s}_j\right)\mathcal{I}[w \in \mathcal{S}_i]\boldsymbol{w}^\top \boldsymbol{s}_k + \tag{8}$$

$$\mathcal{I}[w \in \mathcal{S}_k]\boldsymbol{w}^\top \boldsymbol{s}_i)) \tag{9}$$

Here, the indicator function $\mathcal{I}$ is given by (10).

$$\mathcal{I}[\theta] = \begin{cases} 1 & \theta \text{ is True} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Substituting (10), (7), in (4) we compute $\frac{\partial E}{\partial q(w)}$ and use stochastic gradient descent with initial learning rate set to 0.01 and subsequently scheduled by AdaGrad (Duchi et al., 2011). The NWS scores can be either randomly initialised or set to some other values such as ISF scores. We found experimentally that the best performing models are the ones with the weights initialised with ISF. Source code of our implementation is available[3].

4 Experiments

We use the Toronto books corpus[4] as our training dataset. This corpus contains 81 million sentences from 11,038 books, and has been used as a training dataset in several prior work on sentence embedding learning. Note that only 7,807 books in this corpus are unique. Specifically, for 2,098 books there exist one duplicate, for 733 there are two and for 95 books there are more than two duplicates. However, following the training protocol used in prior work (Kiros et al., 2015), we do not remove those duplicates from the corpus, and use the entire collection of books for training. We convert all sentences to lowercase and tokenise using the Python NLTK[5] punctuation tokeniser. No further pre-processing is conduced beyond tokenisation. The proposed method is implemented using TensorFlow[6] and executed on a NVIDIA Tesla K40c 2880 GPU.

[3] https://bitbucket.org/u3ks/year3
[4] http://yknzhu.wixsite.com/mbweb
[5] http://www.nltk.org/
[6] https://www.tensorflow.org/

4.1 Measuring Semantic Textual Similarity

It is difficult to evaluate the accuracy of word salience scores by direct manual inspection. Moreover, no such dataset exists where human annotators have manually rated words for their salience. Therefore, we resort to extrinsic evaluation, where, we first use (1) to create the sentence embedding for a given sentence using pretrained word embeddings and the NWS scores computed using the proposed method. Next, we measure the semantic textual similarity (STS) between two sentences by the cosine similarity between the corresponding sentence embeddings. Finally, we compute the correlation between human similarity ratings for sentence pairs in benchmark datasets for STS and the similarity scores computed following the above-mentioned procedure. If there exists a high degree of correlation between the sentence similarity scores computed using the NWS scores and human ratings, then it can be considered as empirical support for the accuracy of the NWS scores. Note that we have not trained the word salience model on the SemEval datasets, but are only using them to test the effectiveness of the computed NWS scores. As shown in Table 1, we use 18 benchmark datasets from SemEval STS tasks from years 2012 (Agirre et al., 2012), 2013 (Agirre et al., 2013), 2014 (Agirre et al., 2014), and 2015 (Agirre et al., 2015). Note that the tasks with the same name in different years actually represent different tasks.

We use Pearson correlation coefficient as the evaluation measure. For a list of n ordered pairs of ratings $\{(x_i, y_i)\}_{i=1}^n$, the Pearson correlation coefficient between the two ratings, $r(\boldsymbol{x}, \boldsymbol{y})$, is computed as follows:

$$r(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \tag{11}$$

Here, $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$. Pearson correlation coefficient is invariant against linear transformations of the similarity scores, which is suitable for comparing similarity scores assigned to the same set of items by two different methods (human ratings vs. system ratings).

We use the Fisher transformation (Fisher, 1915) to test for the statistical significance of Pearson correlation coefficients. Fisher transformation, $F(r)$, of the Pearson correlation coefficient r is given by (12).

$$F(r) = \frac{1}{2}\log\left(\frac{1+r}{1-r}\right) \tag{12}$$

Then, 95% confidence intervals are given by (13).

$$\tanh\left(F(r) \pm \frac{1.96}{\sqrt{n-3}}\right) \tag{13}$$

We consider two baseline methods in our evaluations as described next.

Averaged Word Embeddings (AVG) As a baseline that does not use any salience scores for words when computing sentence embeddings, we use *Averaged Word Embeddings* (AVG) where we simply add all the word embeddings of the words in a sentence and divide from the total number of words to create a sentence embedding. This baseline demonstrates the level of performance we would obtain if we did not perform any word salience-based weighting in (1).

Inverse Sentence Frequency (ISF) As described earlier in section 2, term frequency is not a useful measure for discriminating salient vs. non-salient words in short-texts because it is rare for a particular word to occur multiple times in a short text such as a sentence. However, (inverse of) the number of different sentences in which a particular word occurs is a useful method for identifying salient features because non-content stop words are likely to occur in any sentence, irrespective of the semantic contribution to the topic of the sentence. Following the success of Inverse Document Frequency (IDF) in filtering out high frequent words in text classification tasks (Joachims, 1998), we define *Inverse Sentence Frequency* (ISF) of a word as the reciprocal of the number of sentences in which that word appears in a corpus. Specifically, ISF is computed as follows:

$$\text{ISF}(w) = \log\left(1 + \frac{\text{no. of sentences in the corpus}}{\text{no. of sentences containing } w}\right) \tag{14}$$

In Table 1, we compare NWS against AVG, ISF baselines. **SMOOTH** is the unigram probability-based smoothing method proposed by (Arora et al., 2017).[7] We compute sentence embeddings for NWS, AVG and ISF using pre-trained 300 dimensional GloVe embeddings trained from the Toronto books corpus using contextual windows of 10 tokens.[8] For reference purposes we show the level of performance we would obtain if we had used sentence embedding methods such as, skip-thought (Kiros et al., 2015), and Siamese-CBOW (Kenter et al., 2016). Note that however, sentence embedding methods do not necessarily compute word salience scores. For skip-thought, Siamese CBOW and SMOOTH methods we report the published results in the original papers. Because (Kiros et al., 2015) did not report results for skip-thought on all 18 benchmark datasets used here, we report the re-evaluation of skip-thought on all 18 benchmark datasets by (Wieting et al., 2016).

Statistically significant improvements over the ISF baseline are indicated by an asterisk $*$, whereas the best results on each benchmark dataset are shown in bold. From Table 1, we see that between the two baselines AVG and ISF, ISF consistently outperforms AVG in all benchmark datasets. In 9 out of the 18 benchmarks, the proposed NWS scores report the best performance. We suspect that the word salience model has the best performance in the OWNs datasets because they are closest to the training data. However, it outperforms the other models in other datasets such as images, and student-answers which talks about the generalisability of the model. Moreover, in 9 datasets NWS statistically significantly outperforms the ISF baseline. Siamese-CBOW reports the best results in 5 datasets, whereas SMOOTH reports the best results in 2 datasets. Overall, NWS stands out as the best performing method among the methods compared in Table 1.

Our proposed method for learning NWS scores does not assume any specific properties of a particular word embedding learning algorithm. Therefore, in principle, we can learn NWS scores using *any* pre-trained set of word embeddings. To evaluate the accuracy of the word salience scores computed using different word embeddings, we conduct the following experiment. We use SGNS, CBOW and GloVe word embedding learning algorithms to learn 300 dimensional word embeddings from the Toronto books corpus.[9] The vocabulary size, cut-off frequency for selecting words, context window size are are kept fixed across differ-

[7] Corresponds to the GloVe-W method in the original publication.

[8] We use the GloVe implementation by the original authors available at https://nlp.stanford.edu/projects/glove/

[9] We use the implementation of word2vec from https://github.com/dav/word2vec

Table 1: Performance on STS benchmarks.

Dataset	SMOOTH	skip-thought	Siamese-CBOW	AVG	ISF	NWS
2012						
MSRpar	43.6	5.6	**43.8**	28.4	39.1	28.5
OnWN	54.3	60.5	64.4	47.1	60.5	**65.5***
SMTeuroparl	**51.1***	42.0	45.0	37.1	44.5	50.1
SMTnews	42.2	39.1	39.0	32.2	34.9	**44.7***
2013						
FNWN	23.0	**31.2**	23.2	26.9	29.4	25.2
OnWN	68.0*	24.2	49.9	25.0	63.2	**78.1***
headlines	63.8	38.6	**65.3***	40.2	59.4	57.0
2014						
OnWN	68.0	46.8	60.7	41.1	68.5	**80.8***
deft-forum	29.1	37.4	**40.8**	27.1	37.1	29.9
deft-news	**68.5**	46.2	59.1	48.8	63.6	65.4
headlines	59.3	40.3	**63.6***	41.9	58.8	56.2
images	74.1*	42.6	65.0	35.3	66.3	**75.9***
tweet-news	57.3	51.4	**73.2***	41.7	57.1	64.5*
2015						
answers-forums	41.4	27.8	21.8	25.7	37.6	**49.6***
answers-students	61.5	26.6	36.7	56.5	67.1	**68.0**
belief	47.7	45.8	47.7	29.3	43.2	**54.3***
headlines	64.0	12.5	21.5	49.3	**65.4**	65.3
images	75.4*	21	25.6	49.8	66.1	**76.6***
Overall Average	55.1	35.5	47.0	38.0	53.4	**57.6**

ent word embedding learning methods for the consistency of the evaluation. We then trained NWS with each set of word embeddings. Performance on STS benchmarks is shown in Table 2, where the best performance is bolded.

From Table 2, we see that GloVe is the best among the three word embedding learning methods compared in Table 2 for producing pre-trained word embeddings for the purpose of learning NWS scores. In particular, NWS scores reports best results with GloVe embeddings in 10 out of the 18 benchmark datasets, whereas with CBOW embeddings it obtains the best results in the remaining 8 benchmark datasets.

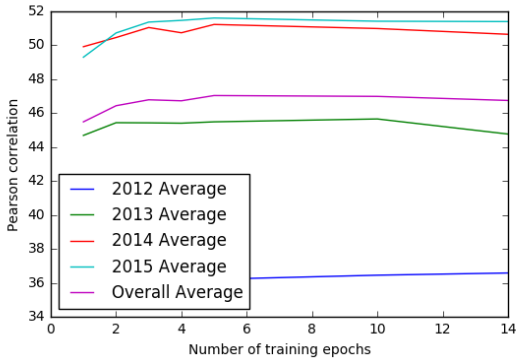
Figures 2a and 2b show the Pearson correlation coefficients on STS benchmarks obtained by NWS scores computed respectively for GloVe and SGNS embeddings. We plot training curves for the average correlation over each year's benchmarks as well as the overall average over the 18 benchmarks. We see that for both embeddings the training saturates after about five or six epochs. This ability to learn quickly with a small number of epochs is attractive because it reduces the training time.

4.2 Correlation with Psycholinguistic Scores

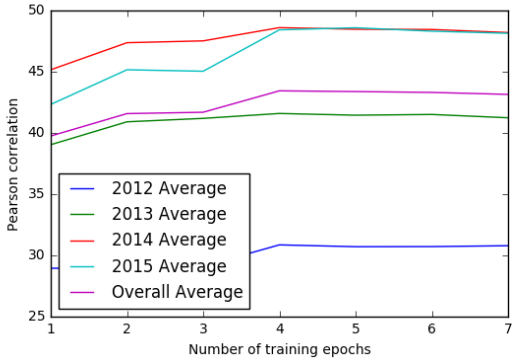
Prior work in psycholinguistics show that there is a close connection between the emotions felt by humans and the words they read in a text. *Valence* (the pleasantness of the stimulus), *arousal* (the intensity of emotion provoked by the stimulus), and *dominance* (the degree of control exerted by the stimulus) contribute to how the meanings of words affect human psychology, and often referred to as the *affective* meanings of words. (Mandera et al., 2015) show that by using SGNS embeddings as features in a k-Nearest Neighbour classifier, it is possible to accurately extrapolate the affective meanings of words. Moreover, perceived psycholinguistic properties of words such as *concreteness* (how "palpable" the object the word refers to) and *imageability* (the intensity with which a word arouses images) have been successfully predicted using word embeddings (Turney et al., 2011; Paetzold and Specia, 2016). For example, (Turney et al., 2011) used the cosine similarity between word embeddings obtained via La-

Table 2: Effect of word embeddings.

Dataset	NWS with pre-trained SGNS	CBOW	GloVe
2012			
MSRpar	14.27	24.15	**28.47**
OnWN	59.76	61.25	**65.50**
SMTeuroparl	41.04	45.51	**50.12**
SMTnews	43.42	**46.94**	44.73
2013			
FNWN	21.47	**29.31**	25.21
OnWN	67.37	70.04	**78.06**
headlines	57.05	**57.46**	57.02
2014			
OnWN	73.06	73.71	**80.83**
deft-forum	28.62	**32.49**	29.90
deft-news	59.63	61.95	**65.35**
headlines	56.05	55.64	**56.20**
images	76.94	**78.08**	75.88
tweet-news	61.49	**66.41**	64.46
2015			
answers-forums	36.35	46.78	**49.65**
answers-students	59.53	59.92	**68.01**
belief	51.97	**55.65**	54.27
headlines	61.24	63.04	**65.32**
images	77.67	**78.39**	76.55
Overall Average	52.60	55.92	**57.52**



(a) GloVe



(b) SGNS

Figure 2: Pearson correlations on STS benchmarks against the number of training epochs

tent Semantic Analysis (LSA) (Deerwester et al., 1990) to predict the concreteness and imageability ratings of words.

Table 3: Pearson correlation coefficients against Psycholinguistic ratings of words in the ANEW and MRC databases.

Embed.	Arousal	Conc.	Dom.	Img.	Valance
GloVe	0.03	0.26	0.09	0.25	0.03
CBOW	0.04	-0.35	-0.04	-0.37	0.04
SGNS	-0.01	0.27	0.06	0.27	-0.01

On the other hand, prior work studying the relationship between human reading patterns using eye-tracking devices show that there exist a high positive correlation between word salience and reading times (Dziemianko et al., 2013; Hahn and Keller, 2016). For example, humans pay more attention to words that carry meaning as indicated by the longer fixation times. Therefore, an interesting open question is that *what psycholinguistic properties of words, if any, are related to the NWS scores we learn in a purely unsupervised manner from a large corpus?* To answer this question empirically, we conduct the following experiment. We used the Affected Norms for English Words (ANEW) dataset created by Warriner et al. (2013), which contains valence, arousal, and dominance ratings collected via crowd sourcing for 13,915 words. Moreover, we obtained concreteness and imageability ratings for 3364 words from the MRC psycholinguistic database. We then measure the Pearson correlation coefficient between NWS scores and each of the psycholinguistic ratings as shown in Table 3.

We see a certain degree of correlation between NWS scores computed for all three word embeddings and the concreteness scores. Both GloVe and SGNS show moderate positive correlations for concreteness, whereas CBOW shows a moderate negative correlation for the same. A similar trend can be observed for imageability ratings in Table 3, where GloVe and SGNS correlates positively with imageability, while CBOW correlates negatively. Moreover, no correlation could be observed for arousal, valance and dominance ratings. This result shows that NWS scores are not correlated with affective meanings of words (arousal, dominance, and valance), but show a moderate level of correlation with perceived meaning scores (concreteness and imageability).

4.3 Sample Salience Scores

Tables 4 and 5 show respectively low and high salient words for ISF, NWS (ISF initialised) and NWS (randomly initialised) methods. words) se-

Table 4: Sample words with the low salience

ISF	NWS (ISF init.)	NWS (rand init.)
the	your	alexis
to	our	tobias
i	we	copyright
and	my	rupert
a	you	spotted
of	us	vehicle
was	me	sword
he	i	isaac
his	voice	fletcher
you	has	cook

Table 5: Sample words with the high salience

ISF	NWS (ISF init.)	NWS (rand init.)
pathways	guess	hurdling
conspiratorial	boulder	happen
henna	autopsy	weird
alejandro	hippy	alejo
bedpost	alejandro	bolivians
swiveling	philosophy	his
confederate	arrow	answer
mid-morning	germany	her
alejo	spotted	yesterday
phd	bookstore	replied

lected from a sample of 1000 words. The probability of each word appearing in the sample was based on its frequency in the text corpus. The fact that the top ranked words with NWS differ from that of ISF suggests that the proposed method learns salience scores based on attributes other than frequency and provides a finer differentiation between words. The effectiveness of the NWS scores when initialised with ISF might be due to incorporating frequency information in addition to salience.

5 Conclusion

We proposed a method for learning Neural Word Salience scores from a sentence-ordered corpus, without requiring any manual data annotations. To evaluate the learnt salience scores, we computed sentence embeddings as the linearly weighted sum over pre-trained word embeddings. Our experimental results show that the proposed NWS scores outperform baseline methods, previously proposed word salience scores and sentence embedding methods on a range of benchmark datasets selected from past SemEval STS tasks. Moreover, the NWS scores shows interesting correlations with perceived meaning of words indicated by concreteness and imageability psycholinguistic ratings.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv* .

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montese Martxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce M. Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proc. of SemEval*. pages 252 – 263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Weibe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proc. of the 8th International Workshop on Semantic Evaluation (SemEval)*. pages 81–91.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proc. of the first Joint Conference on Lexical and Computational Semantics (*SEM)*. pages 385 – 393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. shared task: Semantic textual similarity. In *Proc. of the Second Joint Conference on Lexical and Computational Semantics (*SEM):*. pages 32–43.

Jacob Andreas and Dan Klein. 2015. When and why are log-linear models self-normalizing? In *Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 244–249.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proc. of International Conference on Learning Representations (ICLR)*.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proc. of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pages 546–556.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE* 41(6):391–407.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12:2121 – 2159.

Michal Dziemianko, Alasdair Clarke, and Frank Keller. 2013. Object-based saliency as a predictor of attention in visual tasks. In *Proc. of the 35thth Annual Conference of the Cognitive Science Society*. pages 2237–2242.

R. A. Fisher. 1915. Frequency distribution of the values of the correlation coefficient in samples of an indefinitely large population. *Biometrika* 10(4):507–521.

Michael U. Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research* 13:307 – 361.

Michael Hahn and Frank Keller. 2016. Modeling human reading with neural attention. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*. pages 85–95.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 937–948.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning disributed representations of sentences from unlabelled data. In *Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1367–1377.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of the European Conference on Machine Learning (ECML)*. pages 137–142.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28:11–21.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese cbow: Optimizing word embeddings for sentence representations. In *Proc, of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 941–951.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*. pages 3276–3284.

Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaret. 2015. How useful are corpus-based methods for extrapolating psycholinguistic variables? *The Quarterly Journal of Experimental Pscychology* 68(8):1623–1642.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proc. of Annual Meeting of the Association for Computational Linguistics*. pages 236 – 244.

Gustavi Henrique Paetzold and Lucia Specia. 2016. Inferring psycholinguistic properties of words. In *Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 435–440.

S. E. Robertson. 1997. Overview of the okapi projects. *Journal of Documentation* 53(1):3 – 7.

G. Salton and C. Buckley. 1983. *Introduction to Modern Information Retreival*. McGraw-Hill Book Company.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK., pages 151–161. http://www.aclweb.org/anthology/D11-1014.

Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*. pages 27 – 31.

Yashen Wang, Heyan Huang, Chong Feng, Qiang Zhou, Jiahui Gu, and Xion Gao. 2016. Cse: Conceptutal sentence embeddings based on attention model. In *Proc. of Annual Meeting of the Association for Computational Linguistics*. pages 505–515.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaret. 2013. Norms of valence, arousal, and dominance for 13,915 enlish lemmas. *Behavior Research Methods* 45(4):1191–1207.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proc. of International Conference on Learning Representations (ICLR)*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of Association for Computational Linguistics* pages 259–272.