Evaluating Semantic Localization of Sentence Embeddings through a Semantic Classification Task

withheld
withheld
withheld
withheld

withheld
withheld
withheld
withheld

withheld
withheld
withheld
withheld

withheld
withheld
withheld
withheld

ABSTRACT

Several approaches for embedding a sentence into a vector space have been developed. However, it is unclear to what extent the sentence's position in a vector space reflect its semantic meaning, rather than other factors such as syntactic structure. Depending on the model used for the embeddings this will vary – different models are suited for different down-stream applications. For applications such as machine translation and automated summarization, it is highly desirable to have semantic meaning encoded in the embedding. We consider this to be the quality of *semantic localization* for the model – how well the sentences' meanings coincides with their embedding's position in vector space. Currently the semantic localization is assessed indirectly through practical benchmarks for specific applications.

In this paper, we ground the semantic localization problem by assessing the localization and separability of paraphrase groups in the embedding space. Two corpora, namely, a subset of the Microsoft Research Paraphrase Corpus and and a subset of the Opinosis corpus were grouped into classes of sentences with the same meaning. Several existing models, including URAE, PV–DM and PV–DBOW, were assessed against a bag of words benchmark.

General Terms

Measurement, Performance, Experimentation

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods, Linguistic processing*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language parsing and understanding*

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Keywords

Semantic vector space representations, semantic consistency evaluation, sentence embeddings, word embeddings

1. INTRODUCTION

Sentence embeddings are often referred to as semantic vector space representations [8]. Embedding the meaning of a sentence into a vector space is expected to be very useful for natural language understanding tasks. Vector representation of natural languages enables discourse analysis to take advantage of the array of tools available for computation in vector spaces. However, the embeddings of a sentence may encode a number of factors including semantic meaning, syntactic structure and topic. Since many of these embeddings are learned unsupervised on textual corpora using various models with different training objectives, it is not entirely clear the emphasis placed on each factor in the encoding. For applications where encoding semantic meaning is particularly desirable, such as machine translation and automatic summarization, it is crucial to be able to assess how well the embeddings capture the sentence's semantics. In other words, for successful application to these areas it is required that the embeddings generated by the models correctly encode meaning such that sentences with the same meaning are co-located in the vector space, and sentences with differing meanings are further away. However, few current models are directly trained to optimize for this criteria.

Currently sentence embeddings are often generated as a byproduct of unsupervised, or semi-supervised, tasks. These tasks include: word prediction [11]; recreation of input, as in the auto-encoders of [23, 20] and [8]; alignment of sentence embeddings across a parallel multilingual corpus [7]; and syntactic structural classification [19, 22]. As a result the vector representations of the input sentences learned by these models are tuned towards the chosen optimization task. When employing the embeddings produced as features for other tasks, the information captured by the embeddings often proved to be very useful: e.g. approaching or exceeding previous state-of-the-art results, in sentiment analysis[23, 21, 11] and paraphrase detection[20]. However these practical applications do not directly show how well meaning is captured by the embeddings.

This paper aims to provide a method to assess how well

the models are capturing semantic information. A strict definition for the semantic equivalence of sentences is: that each sentence shall entail the other. Such mutually entailing sentences are called *paraphrases*. In this paper we propose to use paraphrases to assess how well the true semantic space aligns with the vector space the models embed into. It thus assesses whether projecting a sentence via the models in to the vector space preserves meaning.

The evaluation corpora were prepared by grouping paraphrases from the Microsoft Research Paraphrase (MSRP) [3] and Opinosis [5] corpora. A semantic classification task was defined which assesses if the model's embeddings could be used to correctly classify sentences as belonging to the paraphrase group with semantically equivalent sentences. Ensuring that the many sentences of common meaning, but differing form are located in vector space together, is a challenging task and shows a model's semantic encoding strength. This assessment allows for a better understanding of how these models work, and suggest new directions for the development in this area.

The rest of the paper is organized into the following sections. Section §2 discusses the existing models being assessed, the methods traditionally used to assess them, and the more recent work on methods to assess their semantic correctness. Section §3 describes the processes by which the models are evaluated using our new method, and the parameters used in the evaluation. Section §4 continues into more details on the development of the evaluation corpora for the semantic classification evaluation task. Section §5 details the results from evaluating the models and discusses the implications for their semantic consistency. Section §6 closes the paper and suggests new directions for development.

2. BACKGROUND

2.1 Models

Three well known sentence embedding methods are evaluated in this work. The compositional distributed model of the Unfoldering Recussive Autoencoder (URAE) by Socher et. al. [20]; and the two word content predictive models, Distributed Memory (PV-DM) and Distributed Bag of Words by Le and Mikolov [11]. In addition to these advanced sentence embedding models, a simple average of word embeddings, from Mikolov et. al. [14], is also assessed. These models and their variant forms have been applied to a number of natural language processing tasks in the past, as detailed in the subsequent sections, but not to a real-sentence semantic classification task as described here.

2.1.1 Unfolding Recursive Auto-Encoder (URAE)

The Unfolding Recursive Autoencoder (URAE) [20] is an autoencoder based method. It functions by recursively using a single layer feedforward neural-network to combine embedded representations, following the parse tree. Its optimization target is to be be able to reverse (unfold) the merges and produce the original sentence. The central folding layer – where the whole sentence is collapsed to a single embedding vector – is the sentence representation.

2.1.2 PV-DM

The Distributed Memory Paragraph Vectors (PV-DM) [11] method is based on an extension of the Continuous Bag-of-Words word-embedding model [13]. It is trained using a sliding window of words to predict the next word. The softmax predictor network is fed a word-embedding for each word in the window, plus an additional sentence embedding vector which is reused for all words in the sentence – called the paragraph vector in [11]. These input embeddings can be concatenated or averaged; in the results below they were concatenated. During training both word and sentence vectors are allowed to vary, in evaluation (i.e. inference), the word vectors are locked and the sentence vector is trained until convergence on the prediction task occurs.

2.1.3 PV-DBOW

Distributed Bag of Words Paragraph Vectors (PV-DBOW) [11], is based on the Skip-gram model for word-embeddings, also from [13]. In PV-DBOW a sentence vector is used as the sole input to a neural net. That network is tasked with predicting the words in the sentence. At each training iteration, the network is tasked to predict a number of words from the sentence, selected with a specified window size, using the sentence vector being trained as the input. As with PV-DM to infer embedding the rest of the network is locked, and only the sentence vector input allowed to vary, it is then trained to convergence.

2.1.4 Sum and Mean of Word Embeddings (SOWE and MOWE)

Taking the element-wise sum or mean of the word embeddings over all words in the sentence also produces a vector with the potential to encode meaning. Like traditional bag of words no order information is encoded, but the model can take into consideration word relations such as synonymity as encoded by the word vectors. The mean was used as baseline in [11]. The sum of word embeddings first considered in [14] for short phrases, it was found to be an effective model for summarization in [10]. The cosine distance, as is commonly used when comparing distances between embeddings, is invariant between sum and mean of word embeddings. Both sum and mean of word embeddings are computationally cheap models, particularly given pretrained word embeddings are available.

2.2 General Evaluation Methods

As discussed in the introduction, current methods of evaluating the quality of embedding are on direct practical applications designed down-stream. The evaluation methods are not directly link to the methods used for training. One of the more semantically focused is paraphrase detection.

Evaluation on a Paraphrase Detection task takes the form of being presented with pairs of sentences and tasked with determining if the sentences are paraphrases or not. The MSRP Corpus [3] which we used in the semantic classification task is intended for such use. This pairwise check is valuable, and does indicate to a certain extent if the embeddings are capturing meaning, or not. However, by considering groups of paraphrases, a deeper intuition can be gained on the arrangement of meaning within the vector space.

Sentiment Analysis is very commonly used task for evaluating embeddings. It was used both for the recursive autoencoder in [23] and for the paragraph vector models in [11]. Sentiment Analysis is normally tasked with classifying as positive or negative, or assigning a score, as in the Sentiment Treebank [24]. Determining the sentiment of a sentence is partially a semantic task, but it is lacking in sev-

eral areas that would be required for meaning. For example, there is only an indirect requirement for the model to process the subject at all. Sentiment Analysis is a key task in natural language processing, but it is very distinct from semantic meaning.

Document Classification is a classic natural language processing task. A particular case of this is topic categorization. Early work in the area goes back to [12] and [1]. Much more recently it has been used to assess the convolution neural networks of [26], where the articles of several news corpora were classified into categories such as "Sports", "Business" and "Entertainment". The topics for classification tend to be broad. A huge spectrum of different sentences are assigned to the same topic. It is thus too board and insufficiently specific to evaluate the consistency of meanings. Information retrieval can be seen as the inverse of the document classification task.

Information Retrieval is the task of identifying the documents which most match a query. Such document selection depends almost entirely on topic matching. Suitable results for information retrieval have no requirement to agree on meaning, though text with the same meaning are more likely to match the same queries.

The evaluation of semantic consistency requires a task which is fine grained, and preserving meaning. Document Classification and Information Retrieval are insufficiently fine-grained. Sentiment Analysis does not preserve meaning, only semantic orientation. Paraphrase Detection is directly relevant to evaluating semantic constancy, however it is a binary choice based on a pairwise comparison – a more spatial application is desirable for evaluating these vector spaces. Thus the current down-steam application tasks are not sufficient for assessing semantic consistency – more specialized methods are required.

2.3 Evaluations of Semantic Consistency

Semantic consistency for word embeddings is often measured using the analogy task. In an analogy the meta-relation: `A is to B as C is to D`. Mikolov et. al.[15] demonstrated that the word-embedding models are semantically consistent by showing that the semantic relations between words were reflected as a linear offset in the vector space. That is to say, for embeddings $\tilde{x}_a$, $\tilde{x}_b$, $\tilde{x}_c$, $\tilde{x}_d$ corresponding to words A, B, C and D, respectively; it was tested that if for a strong relationship matching between A/B and C/D, then the offset vector would be approximately equal: $\tilde{x}_b - \tilde{x}_a \approxeq \tilde{x}_d - \tilde{x}_c$. Rearranging this in word space gets the often quoted example of $\texttt{King} - \texttt{Man} + \texttt{Woman} \approxeq \texttt{Queen}$, As man is to woman, king is to queen. In the rating task as described by [9], the goal is to rank such analogous word pairs based on the degree the relation matches. Thus to evaluate the word-embedding model using this task, it was a matter of sorting closeness of the corresponding offset vectors. Surprisingly strong results were found on this task[15]. It was thus demonstrated that word embeddings were not simply semantically consistent, but more so that this consistency was displayed as local linearity. This result gives confidence in the semantic quality of the word embeddings. However, this relationship analogy test cannot be performed for sentence embeddings.

Gershman et. al. [6], compares the distances of modified sentences in vector space, to the semantic distances ascribed to them by human raters. Like the analogy task for word vectors, this task requires ranking the targets based on the vector distance, however instead of rating on the strength of relationships it measures simply the similarities of the sentences to an original base sentence for each group. In that evaluation 30 simple base sentences of the form `A [adjective1] [noun1] [prepositional phrase] [adjective2] [noun2]` were modified to produce 4 difference derived sentences. The derived sentences were produced by swapping the nouns, swapping the adjectives, reversing the positional phrase (so `behind` becomes `in front of`), and a paraphrase by doing all of the aforementioned changes. Human raters were tasked with sorting the transformed sentences in similarity to the base sentence. This evaluation found that the embedding models considered did not agree with the semantic similarity rankings placed by humans. While the sentence embedding models performed poorly on the distance ranking measure, it is also worth considering how they perform on a meaning classification task.

A meaning classification task was recently proposed by Ritter et. al. [18], to classify sentences based on which spatial relationship was described. The task was to classify the sentence as describing: *Adhesion to Vertical Surface*, *Support by Horizontal Surface*, *Full Containment*, *Partial Containment*, or *Support from Above*. In this evaluation also, the sentences took a very structured form: `There is a [noun1] [on/in] the [noun2]`. These highly structured sentences take advantage of the disconnection between word content and the positional relationship described to form a task that must be solved by a compositional understanding combining the understanding of the words. *"The apple is on the refrigerator"* and *"The magnet is on the refrigerator"* belong to two separate spatial categories, even though the word content is very similar. Surprisingly, the simple model of adding word vectors outperformed compositional models such as the recursive autoencoder. The result does have some limitation due to the highly artificial nature of the sentences, and the restriction to categorizing into a small number of classes based only on the meaning in terms of positional relationship. To generalize this task, in this paper we consider real world sentences being classed into groups according to their full semantic meaning.

3. METHODOLOGY

To evaluate how well a model's vectors capture the meaning of a sentence, a semantic classification task was defined. The task is to classify sentences into classes where each shares the same meaning. Each class is thus defined as a paraphrase groups. This is a far finer-grained task than topic classification. It is a multiclass classification problem, rather than the binary decision problem of paraphrase detection. Such multiclass classification requires the paraphrase groups to be projected into compact and distinct groups in the vector space. A model which produces such embeddings which are thus easily classifiable according to their meaning can been thus seen to have good semantic localization.

This semantic classification does not have direct practical application – it is rare that the need will be to quantify sentences into groups with the same prior known meaning. Rather it serves as a measure to assess the models general suitability for other tasks requiring a model with consistency between meaning and embedding.

To evaluate the success at the task three main processes are involved, as shown in Figure 1: Corpus Preparation,

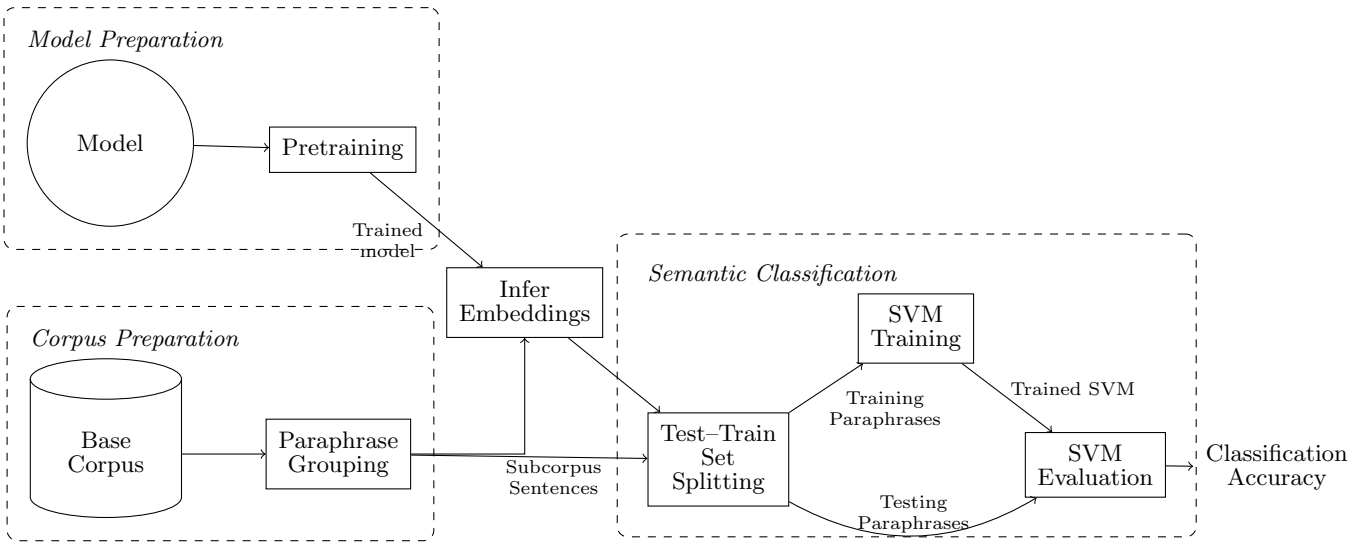

Figure 1: Process Diagram for the Evaluation of Semantic Consistency via our method

Model Preparation, and the Semantic Classification task itself.

3.1 Corpus Preparation

The construction of each of the corpora is detailed more fully in the next section. In brief: Two corpora were constructed by selecting subsets of the Microsoft Research Paraphrase (MSRP) [3] and of the Opinosis [5] corpora. The corpora were partitioned into groups of paraphrases – sentences with the same meaning. Any paraphrase groups with less than three sentences were discarded. The paraphrase grouping was carried out manually for Opinosis, and automatically for the MSRP corpus using the existing paraphrase pairings. The paraphrase groups divide the total semantic space of the corpora into discrete classes, where each class contains sentences sharing the same meaning.

It is by comparing the ability of the models to produce embeddings which can be classified back into these classes, that we can compare the real semantic space partitions to their corresponding vector embedding space regions.

3.2 Model Preparation and Inferring Vectors

Prior to application to semantic classification, as with any task the models had to be pretrained. Here we use the term *pretraining* to differentiate the model training from the classifier training. The pretraining is not done using the evaluation corpora as they are both very small. Instead other data are used, and the inference/evaluation procedure given for each method was then used to produce the vectors for each sentence. The model parameters used are detailed below.

3.2.1 Unfolding Recursive Auto-Encoder (URAE)

In this evaluation we make use of the pretrained network that Socher et. al. have graciously made available[1], full information is available in the paper[20]. It is initialized on the unsupervised Collobert and Weston word embeddings[2], and training on a subset of 150,000 sentences from the gigaword corpus. It produces embeddings with 200 dimensions. This pretrained model when used with dynamic pooling and other word based features performed very well on the MSRP corpus paraphrase detection. However in the evaluation below the dynamic pooling techniques are not used as they are only directly suitable for enhancing pairwise comparisons between sentences.

3.2.2 Paragraph Vector Methods (PV-DM and PV-DBOW)

Both PV-DM and PV-DBOW, were evaluated using the GenSim implementation [17] from the current *develop* branch[2]. Both were trained on approximately 1.2 million sentences from randomly selected Wikipedia articles, and the window size was set to 8 words, and the vectors were of 300 dimensions.

3.2.3 Sum and Mean of Word Embeddings (SOWE and MOWE)

The word embeddings used for MOWE were taken from the Google News pretrained model[3] based on the method described in [14]. This has been trained on 100 million sentences from Google News. A small portion of the evaluation corpus did not have embeddings in the Google News model. These tokens were largely numerals, punctuation symbols, proper nouns and unusual spellings, as well as the stop-words: “and”, “a” and “of”. These words were simply skipped. The resulting embeddings have 300 dimensions, like the word embeddings they were based on.

3.2.4 Bag of Words (BOW and PCA BOW)

A bag of words (BOW) model is also presented as a baseline. There is a dimension in each vector embedding for the count of each token, including punctuation, in the sentence. In the Opinosis and MSRP subcorpora there were a total of

[1]http://www.socher.org/index.php/Main/DynamicPoolingAndUnfoldingRecursiveAutoencodersForParaphraseDetection

[2]https://github.com/piskvorky/gensim/tree/develop/

[3]https://code.google.com/p/word2vec/

1,085 and 2,976 unique tokens respectively, leading to BOW embeddings of corresponding dimensionality. As it is a distributional rather than distributed representation, the BOW model does not need any pretraining step. For comparison to the lower dimensional models Principle Component Analysis (PCA) was applied to the BOW embeddings to produce an additional baseline set of embeddings of 300 dimensions – in line with PV-DM, PV-DBOW, SOWE, and MOWE models. It does not quite follow the steps shown in Figure 1, as the PCA pretraining step is performed on the training embeddings only during the SVM classification process, and it is used to infer the PCA BOW embeddings during the testing step. This avoids unfair information transfer where the PCA would otherwise be about to choose representations optimized for the whole set, including the test data. It was found that when the PCA model was allowed to cheat in this way it performed a few percentage points better. The bag of words models do not have any outside knowledge.

3.3 Semantic Classification

The core of this evaluation procedure is in the semantic classification step. A support vector machine (SVM), with a linear kernel, and class weighting was applied to the task of predicting which paraphrase group each sentence belongs to. Classification was verified using 3-fold cross-validation across different splits of the testing/training data, the average results are shown in this section. The splits were in proportion to the class size. For the smallest groups this means there were two training cases and one test case to classify.

In this paper, only a linear kernel was used, because a more powerful kernel such as RBF may be able to compensate for irregularities in the vector space, which makes model comparison more difficult. Scikit-learn [16] was used to orchestrate the cross-validation and to interface with the LibLinear SVM implementation [4]. As the linear SVM's classification success depends on how linearly separable the input data is, thus this assessed the quality of the localization of the paraphrase groupings embeddings.

4. CORPUS CONSTRUCTION

4.1 Microsoft Research Paraphrased Grouped Subcorpus

The MSRP corpus is a very well established data set for the paraphrase detection task [3]. Sentences are presented as pairs which are either paraphrases, or not. A significant number of paraphrases appear in multiple different pairings. Using this information, groups of paraphrases can be formed.

The corpus was partitioned according to sentence meaning by taking the symmetric and transitive closures the set of paraphrase pairs. For example if sentences A, B, C and D were present in the original corpus as paraphrase pairs: A, B, D, A and B, C then the paraphrase group $\{A, B, C, D\}$ is found. Any paraphrase groups containing less than 3 phrases were discarded. The resulting sub-corpus has the breakdown as shown in Figure 2.

4.2 Opinosis Paraphrase Grouped Subcorpus

The Opinosis Corpus[5] was used as secondary source of original real-world text. It is sourced from several online review sites: Tripadvisor, Edmunds.com, and Amazon.com, and contains single sentence statements about hotels, cars and electronics. The advantage of this as a source for texts is that comments on the quality of services and products tend to be along similar lines. The review sentences are syntactically simpler than sentences from a news-wire corpus, and also contain less named entities. However, as they are from more casual communications, the adherence to grammar and spelling may be less formal.

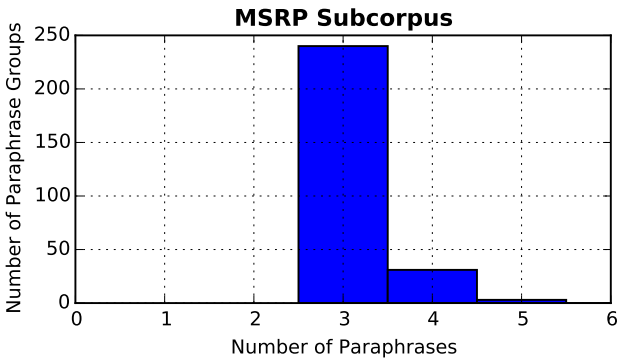

Figure 2: Break down of how many paraphrases groups are present in the MSRP subcorpus of which sizes.It contains a total of 859 unique sentences, broken up into 273 paraphrase groups.

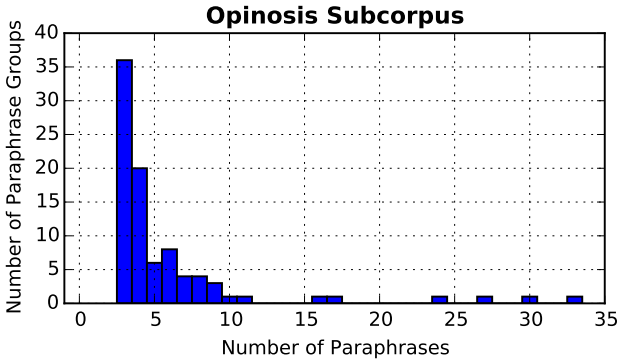

Figure 3: Break down of how many paraphrases groups are present in the Opinosis subcorpus of which sizes. It contains a total of 521 unique sentences, broken up into 89 paraphrase groups.

Paraphrases were identified using the standard criterion: bidirectional entailment. For a paraphrase group $\mathcal{S}$ of sentences: $\forall s_1, s_2 \in \mathcal{S}, \quad s_1 \vDash s_2 \quad \wedge \ s_2 \vDash s_1$, every sentence in the group entails the every other sentence in the group. A stricter interpretation of bidirectional entailment was used, as compared to the "mostly bidirectional entailment" used in the MSRP corpus. The grouping was carried out manually. The general guidelines were as follows.

- Tense, Transitional Phrases, and Discourse and Pragmatic Markers were ignored.
- Statement intensity was coarsely quantized.
- Approximately equal quantitative and qualitative values were treated as synonymous.
- Sentences with entities mentioned explicitly were grouped

	MSRP Subcorpus	Opinosis Subcorpus
PV-DM	78.00%	38.26%
PV-DBOW	89.93%	32.19%
URAE	51.14%	20.86%
MOWE	97.91%	**69.30%**
SOWE	98.02%	68.75%
BOW	**98.37%**	65.23%
PCA BOW	97.96%	54.43%

Table 1: The semantic classification accuracy of the various models across the two evaluation corpora.

separately from similar statements where they were implied.

- Sentences with additional information were grouped separately from those without that information.

The final point is the most significant change from the practices apparent in the construction of the MSRP corpus. Sentences with differing or additional information were classified as non-paraphrases. This requirement comes from the definition of bidirectional entailment. For example, *"The staff were friendly and polite."*, *"The staff were polite."* and *"The staff were friendly."* are in three separate paraphrase groups. The creators of the MSRP corpus, however, note "...the majority of the equivalent pairs in this dataset exhibit 'mostly bidirectional entailments', with one sentence containing information 'that differs' from or is not contained in the other." [3]. While this does lead to more varied paraphrases; it strays from the strict linguistic definition of a paraphrase, which complicates the evaluation of the semantic space attempted here. This stricter adherence to bidirectional entailment resulted in finer separation of groups, which makes this a more challenging corpus.

After the corpus had been broken into paraphrase groups some simple post-processing was done. Several artifacts present in the original corpus were removed, such as substituting the ampersand symbol for `&amp`. Any paraphrase groups containing identical sentences were merged, and duplicates removed. Finally, any group with less than three phrases was discarded. With this complete the breakdown is as in Figure 3.

Further information on the construction of the corpora in this section, and download links are available online.[4]

5. RESULTS AND DISCUSSION

5.1 Classification Results and Discussion

The results of performing the evaluation method described in Section §3 are shown in Table 1.

While the relative performance of the models is similar between the corpora, the absolute performance differs. On the absolute scale, all the models perform much better on the MSRP subcorpus than on the Opinosis subcorpus. This can be attributed to the significantly more distinct classes in the MSRP subcorpus. The Opinosis subcorpus draws a finer line between sentences with similar meanings. As discussed earlier, for example there is a paraphrase group for *"The staff were polite."*, another for *"The staff were friendly."*, and a third for *"The staff were friendly and polite."*. Under the guidelines used for paraphrases in MSRP, these would all have been considered the same group. Secondly, there is a much wider range of topics in the MSRP. Thus the paraphrase groups with different meanings in MSRP corpus are also more likely to have different topic entirely than those from Opinosis. Thus the the ground truth of the semantics separability of phrases from the MSRP corpus is higher than for Opinosis. Making the semantic classification of the Opinosis subcorpus is a more challenging task.

The URAE model performs the worst of all models evaluated. In [10] is was suggested that the URAE's poor performance at summarizing the Opinosis corpus could potentially be attributed to the less formally structured product reviews – the URAE being a highly structured compositional model. However, here it also performed poorly on the MSRP – which it was created for [20]. The exact same model from [20] was used here – though this did put it at a dimensional disadvantage over the other models having 200 dimensions to the other's 300. The key difference from [20], beyond the changing to a multiclass classification problem, was the lack of the complementary word-level features as used in the dynamic pooling layer. This suggests the model could benefit from such world level features – as the very strong performance of the word-based model indicates.

The word based models, MOWE, SOWE, BOW and PCA BOW, performed very well. This suggests that word choice is a very significant factor in determining meaning; so much so that the models which can make use of word order information, URAE and PV-DM, were significantly outperformed by methods which made more direct use of the word content.

The very high performance of the BOW maybe attributed to its very high dimensionality, though the MOWE and SOWE performed similarly. The PCA step can be considered as being similar to choosing an optimal set of words to keep so as to maximum variability in the bag of words – like manually to choosing the best keywords/stopwords. It loses little performance, even though decreasing vector size by an order of magnitude – particularly on the easier MSRP dataset.

5.2 Model Agreement

The misclassifications of the models can be compared. By selecting one of the test/train folds from the classification task above, and comparing the predicted classifications for each test-set sentence, the similarities of the models were assessed. The heatmaps in Figure 4 and Figure 5 show the agreement in errors. Here misclassification agreement is given as an approximation to $P(m_1(x) = m_2(x) \,|\, m_1(x) \neq y \wedge m_2(x) \neq y)$, for a randomly selected sentence x, with ground truth classification y, where the models m_1 and m_2 are used to produce classifications. Only considering the the cases where both models were incorrect, rather than simple agreement, avoids the analysis being entirely dominated by the agreement of the models with the ground truth.

The word based models showed significant agreement. Unsurprisingly MOWE and SOWE have almost complete agreement in both evaluations. The other models showed less agreement – while they got many of the same cases wrong the models produced different misclassifications. This overall suggests that the various full sentence models are producing substantially dissimilar maps from meaning to vector space. Thus it seems reasonable that using a ensemble

[4][withheld]

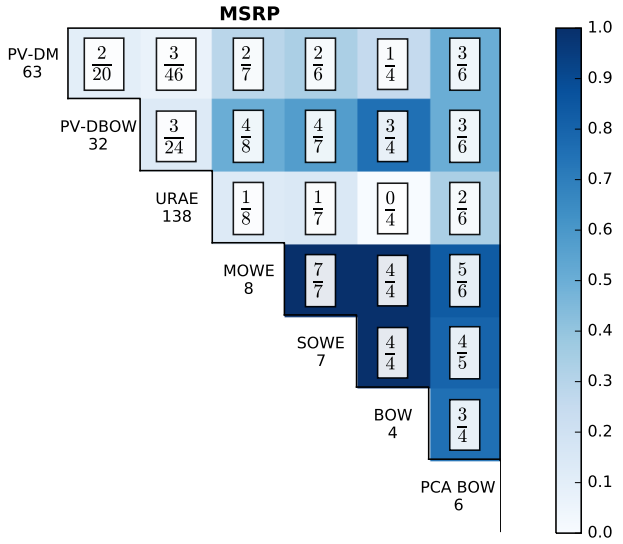

Figure 4: The misclassification agreement between each of the models for the MSRP subcorpus. Below each model name is the total mistakes made. The denominator of each fraction is the number of tests cases incorrectly classified by both models. The numerator is the portion of those misclassifications which were classified in the same (incorrect) way by both models. The shading is in-proportion to that fraction.

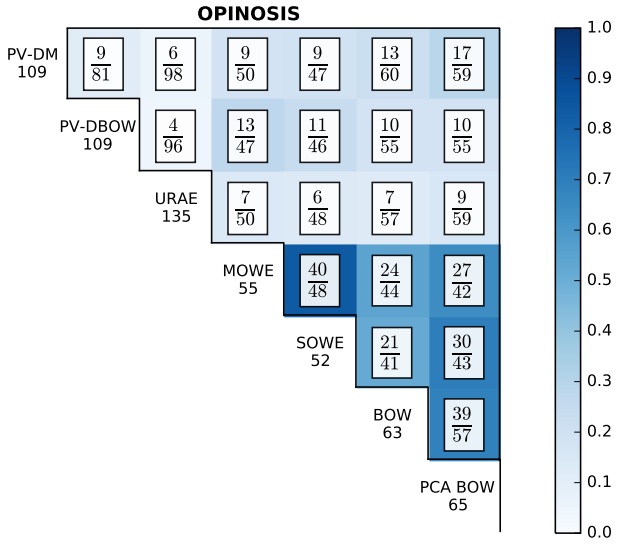

Figure 5: The misclassification agreement between each of the models for the Opinosis subcorpus. Below each model name is the total mistakes made. The denominator of each fraction is the number of tests cases incorrectly classified by both models. The numerator is the portion of those misclassifications which were classified in the same (incorrect) way by both models. The shading is in-proportion to that fraction.

approach between multiple sentence models and one word-based model would produce strong results. Yin and Schütze [25] found this successful when combining different word embedding models.

5.3 Limitations

This evaluation has some limitations. As with all such empirical evaluations of machine learning models, a more optimal choice of hyper-parameters and training data will have an impact on the performance. In particular, if the model training was on the evaluation data the models would be expected to be better able to position their embedding. This was however unfeasible due to the small sizes of the datasets used for evaluation, and would not reflect real word application of the models to data not prior seen. Beyond the limitation of the use of the datasets, is their contents.

The paraphrase groups were not selected to be independent of the word content overlap – they were simply collected on commonality of meaning from real world sourced corpora. This is a distinct contrast to the the work of Ritter et. al.[18] discussed in section 2.3 where the classes were chosen to not have meaningful word overlap. However our work is complementary to theirs, and our findings are well aligned. The key difference in performance is the magnitude of the performance of the sum of word embeddings (comparable to the mean of word embeddings evaluated here). In [18] the word embedding model performed similarly to the best of the more complex models. In the results presented above we find that the word embedding based model performs significantly beyond the more complex models. This can be attributed to the word overlap in the paraphrase groups – in real-world speech people trying to say the same thing do in fact use the same words very often.

6. CONCLUSION

A method was presented, to evaluate the semantic localization of sentence embedding models. Semantically equivalent sentences are those which exhibit bidirectional entailment – they each imply the truth of the other. Paraphrases are semantically equivalent. The evaluation method is a semantic classification task – to classify sentences as belonging to a paraphrase group of semantically equivalent sentences. This classification was performed across two subcorpora derived from existing sources, from the closure of the MRSP corpus, and manually group from the Opinosis corpus. The relative performance of various models was consistent across the two tasks, though differed on an absolute scale.

The word embedding and bag of word models performed best, followed by the paragraph vector models, with the URAE trailing in both tests. The strong performance of the sum and mean of word embeddings (SOWE and MOWE) compared to the more advanced models aligned with the results of Ritter et. al.[18]. The difference in performance presented here for real-word sentences, were more marked than for the synthetic sentence used by Ritter et. al. This may be attributed to real-world sentences often having meaning overlap correspondent to word overlap. Combining the results, it can be concluded that summing word vector representations is a practical and surprisingly effective method for encoding the meaning of a sentence.

7. REFERENCES

[1] H. Borko and M. Bernick. Automatic document classification. *Journal of the ACM (JACM)*, 10(2):151–162, 1963.

[2] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[3] W. B. Dolan and C. Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, 2005.

[4] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[5] K. Ganesan, C. Zhai, and J. Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 340–348. Association for Computational Linguistics, 2010.

[6] S. J. Gershman and J. B. Tenenbaum. Phrase similarity in humans and machines. *Proceedings of the 37th Annual Conference of the Cognitive Science Society*, 2015.

[7] K. M. Hermann and P. Blunsom. A simple model for learning multilingual compositional semantics. *CoRR*, abs/1312.6173, 2013.

[8] M. Iyyer, J. Boyd-Graber, and H. D. III. Generating sentences from semantic vector space representations. In *NIPS Workshop on Learning Semantics*, 2014.

[9] D. A. Jurgens, P. D. Turney, S. M. Mohammad, and K. J. Holyoak. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364. Association for Computational Linguistics, 2012.

[10] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi. Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL*, pages 31–39, 2014.

[11] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

[12] M. E. Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[15] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[18] S. Ritter, C. Long, D. Paperno, M. Baroni, M. Botvinick, and A. Goldberg. Leveraging preposition ambiguity to assess compositional distributional models of semantics. *The Fourth Joint Conference on Lexical and Computational Semantics*, 2015.

[19] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *ACL*. 2013.

[20] R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems 24*. 2011.

[21] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

[22] R. Socher, C. D. Manning, and A. Y. Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9, 2010.

[23] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[24] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[25] W. Yin and H. Schütze. Learning word meta-embeddings by using ensembles of embedding sets. Aug. 2015.

[26] X. Zhang and Y. LeCun. Text understanding from scratch. *CoRR*, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2015.