Take My Word (Representations) For It: An approach for Word, Phrase and Sentence Similarity.

Introduction:

One of the challenges in NLP lies in capturing the nuances and meanings encoded by each individual word. Using the correct representation of these words is paramount for the success of any NLP task; however, we often find that no single representation works for every single task.

Treating each word as an isolated entity, like in methods such as One-Hot Encoding, can lead to models that miss important contextual information. On the other hand, relying solely on relationships between words ignores the unique character of each term.

With respect to our particular task, we believed that while no single representation could directly solve the presented problem; instead a combination of various different text embeddings (representations) could present us with an improved system to solve the same.

Parts of the Task:

- ### Word Similarity (Constrained)

  (Attempted): Used multiple metrics, word embeddings to predict the similarity of two words as per the SimLex-999 dataset. [1]

- ### Word Similarity (Unconstrained)

  (Attempted): Used previously used metrics in constrained version, as well as BERT similarity of word embeddings to predict the similarity as in the same SimLex-999 dataset.

- ### Phrase Similarity (Constrained)

  (Attempted): Predicting whether two phrases are similar (i.e can be used interchangeably) in the context of a particular sentence. [2]

- ### Phrase Similarity (Unconstrained)

  (Attempted): Used embeddings from `all-MiniLM-L6-v2` and similarity to predict whether the two phrases can be used interchangeably in a particular context.

- ### Sentence Similarity (Constrained)

  (Attempted:) Used sentence embeddings and derivable propositions to predict whether two adversial sentences are paraphrases, as per the given dataset. [3]

- ## Sentence Similarity (Unconstrained)

  (Attempted): Used embeddings from `all-MiniLM-L6-v2` for sentence embeddings and derivable propositions.

  Due to unforeseen issues in Google Colab, etc., we use embeddings from `all-MiniLM-L6-v2` [12] for phrase and sentence similarity instead of `bert-base-uncased`.

Word Similarity Task:

Methodology -

Most content words (excluding stopwords) can be represented by a three tuple of values known as **Valence**, **Arousal** and **Dominance**, which aims to represent the word by the emotion evoked by its usage; in particular, its pleasantness (Valence), intensity (Arousal) and degree of control exerted (Dominance.) [4]

We refer to the work of Saif [5], where 20,000 English words were given real valued scores for each of the three dimensions by manual annotation using Best-Worst Scaling. [6]

The aim of the task was to predict the SimLex score provided in the data, using different word representations and embeddings.

Various ML models were tested for this case (Linear Regression, SVM, Random Forest) and it was determined that SVMs were best suited for the task; which was decided by comparing mean cross-validation scores on 5 splits of the data.

We compared the performance of SVM and Neural Networks for the various embedding approaches followed. The following represents the general architecture of the Neural Networks trained;

Layer Type	**Number of Nodes**	**Activation**
Dense	128	ReLU (Input)
Dense	64	ReLU
Dense	1	None (Output)

The Neural models were all trained on 100 epochs, with a batch size of 16. The Adam Optimizer was used during training.

The SVM and Neural Model were compared after incrementally adding new features on top of the previous model. The following features were used:

1. VAD Scores
2. VAD Scores + Word2Vec Similarity [8]
3. VAD Scores + Wordnet (Wu-Palmer) Similarity [9]
4. VAD Scores + Wu-Palmer Similarity + Word2Vec Similarity

Observations -

Features	Model	MSE	<= 1SD	<= 2 SD
VAD	SVM	5.264556766795256	78	134
VAD	NN	5.546095848083496	77	128
VAD + Word2Vec	SVM	5.332867766938792	80	130
VAD + Word2Vec	NN	5.1631999015808105	77	131
VAD + WordNet	SVM	4.078742705415954	89	151
VAD + WordNet	NN	4.390047550201416	86	147
VAD + Word2Vec + WordNet	SVM	4.0478806839649435	91	152
VAD + Word2Vec + WordNet	NN	4.52810525894165	90	140
VAD + BERT Similarity + WordNet	SVM	4.0478806839649435	91	152
VAD + BERT Similarity + WordNet	NN	4.312497138977051	87	149

Analysis -

Overall, we see that Neural Networks are underperforming the SVM counterparts. This may be as a result of the small train/validation size for the dataset provided for this task. A surprising result that is noted is the same performance of the model using Word2Vec similarity as opposed to similarity of BERT embeddings.

It is believed that this is in part because the fact that BERT embeddings are contextual, and so part of the information encoded in its embedding is already latent in our dataset by virtue of the VAD scores and WordNet similarity. In contrast, the WordNet similarity only depends on co-occurences in the corpus trained, and such does not encode meaning similar to the VAD scores/Wu-Palmer similarity.

The model trained on VAD Scores, Word2Vec Similarity and Wu-Palmer Similarity performs well in distinguishing between words that are commonly antonyms (such as `accept` and `forgive`; or `make` and `destroy`), often predicting a score within 2 standard deviations from the value provided in the dataset. (With exceptions; `floor` and `ceiling` received high similarity scores from the model.)

The model's performance for synonym pairs varies with the specific pair used. For example, `boundary` and `border` received a significantly low similarity. On the other hand, `know` and `comprehend`, `frustration` and `anger` received better similarity scores within 2 standard deviations of the value provided in the dataset.

Phrase Similarity Task:

Methodology -

In an isolated scenario, devoid of external context from a sentence, it was assumed that phrases can be compared by averaging the individual word similarity scores using a greedy matching approach, similar to that used in the paper by Zhang et al. [7]

However, as mentioned above, this metric is devoid of context, and is only so useful in a task checking the similarity of a phrase in a particular target sentence.

We have used a metric based on the average similarity of the content words in the sentence and the root of the phrase using the WordNet (Wu-Palmer) similarity metric. This metric is based on the assumptions that lead to the Lesk Algorithm [10] for Word Sense Ambiguation, that the words in the surrounding context would be similar to a particular sense of the word.

Observations -

We observe the following performance metrics among various models:

Due to computation constraints, we used a sample of 7000 data points from the training set; with a test size of 1000 examples and a validation set of size 2000.

Features	**Model**	**Accuracy**
Greedy Matching + Overlap	Logistic Regression	50.35%
Greedy Matching + Overlap	Random Forest (100 Estimators)	43.79%
Greedy Matching + Overlap	NN	50.05%
Greedy Matching + Overlap	SVM	53.64%
Greedy Matching + Overlap (Sentence Transformers)	Logistic Regression	52.81%
Greedy Matching + Overlap (Sentence Transformers)	Random Forest (100 Estimators)	50.90%
Greedy Matching + Overlap (Sentence Transformers)	NN	54.01%
Greedy Matching + Overlap (Sentence Transformers)	SVM	50%

The Neural Model used for this task is described below:

Layer	**Number of Nodes**	**Activation**
Dense	128	ReLU (Input)
Dense	64	ReLU
Dense	1	Sigmoid (Output)

- Optimizer: Adam
- Epochs: 100
- Batch Size: 64

Analysis -

The overall performance in identifying the correct context of the sentence for the given phrase is **varied**; and may be improved by weighting of content words, which has not been implemented. In some examples, the system is correctly able to determine the context, such as `on air`, as opposed to `on posture while jumping`, as in the sentence: `In 1990, Petit accepted a full-time overnight on air position at gospel radio station WYLD-AM.`

However, we also see that the prediction of the phrase pair `prior case` and `preceding game` is a False Positive, given the context of the sentence: `However, James Alfred was not convicted in this or in a prior case of a similar nature.` The model used for computing word similarites in this case was FastText, trained on 1 million tokens from the Brown Corpus; we believe that the vocabulary of this corpus was insufficient for examples such as this, resulting in the incorrect prediction.

We believe that the overall performance may be improved with other Neural model architectures. Using contextual text embedding approaches (for the Unconstrained version of this task) such as embeddings from the `all-MiniLM-L6-v2` transformer increases the performance as opposed to simple static embeddings such as Word2Vec or FastText.

Sentence Similarity Task:

Methodology -

Two sentences can be compared using Sentence Embeddings, using the Cosine Similarity measure which has been employed in several of the previous portions of this task. These sentence embeddings are obtained by aggregating individual word embeddings of the tokens in the sentence, and in the case of some models, also takes in consideration the parsed grammar/dependency relations in the sentence.

However, we believe that a simple similarity will not be sufficient in evaluating the sentences in the context of our dataset; given that the two sentences are generated by rearranging its words, a simple aggregation of word embeddings is believed to be insufficient.

As a preliminary check, we check whether the Named Entities in both sentences are identical/ measure their similarities. It is expected that data points having low similarity with respect to this NER-based system will not be paraphrases.

We also use the system developed by Del Corro et al, ClausIE, to extract the various propositions that can be derived from a given sentence using its clauses; and as such, from its parse trees and dependency grammar. We refine the sentence similarity by additionally obtaining a similarity score over all propositions, by means of a greedy matching approach.

Observations -

Due to computation constraints, we used a sample of 7000 data points from the training set; with a test size of 1000 examples and a validation set of size 2000.

The performance is tabulated below:

Features	Model	Accuracy
Sentence Similarity + Entity Match + Clause Similarity	Logistic Regression	54.88%
Sentence Similarity + Entity Match + Clause Similarity	Random Forest (n_estimators=100)	52.88%
Sentence Similarity + Entity Match + Clause Similarity	SVM	56.01%
Sentence Similarity + Entity Match + Clause Similarity	NN	55.88%

Due to computation constraints, while testing performance of contextual embeddings from `all-MiniLM-L6-v2` we instead train models on 1000 samples, with test size and validation size of 200 examples.

The performance is noted below:

Features	Model	Accuracy
Sentence Similarity + Entity Match + Clause Similarity (Contextual Embeddings)	Logistic Regression	56.20%
Sentence Similarity + Entity Match + Clause Similarity (Contextual Embeddings)	Random Forest (n_estimators=100)	62.09%
Sentence Similarity + Entity Match + Clause Similarity (Contextual Embeddings)	SVM	58.16%
Sentence Similarity + Entity Match + Clause Similarity (Contextual Embeddings)	NN	56.86%

The Neural Model used for this task is described below:

Layer	Number of Nodes	Activation
Dense	128	ReLU (Input)
Dense	64	ReLU
Dense	1	Sigmoid (Output)

Analysis -

True Positives: The model is capable in correctly identifying sentences where adjunct clauses (separated by commas, modifying the verb/noun without completing meaning in the sentence) are rearranged within in the same sentence; and when items provided in a list are given in a different order, etc.

False Positives: The model fails to capture some relations where entity positions are swapped; these are commonly noted to be complements in the tree grammar of the sentence.

True Negatives: The model is able to correctly identify sentences where Named Entities in the same positions are modified to create a new entity; consider `In June of 1997 , Kristin met Richard Armstrong .` and `In June of 1997 , Armstrong met Kristin Richard .` where new `PERSON`s

have been introduced by means of rearranging the words of the sentence. The Entity Match/Similarity appears to be a useful feature in determining the correctness of examples such as above.

False Negatives: The model may be failing due to the probabilistic nature of the proposition extraction from Clauses. The sentence embeddings are resistant to individual synonyms to an extent; however, in certain cases, such as `sophomore` and `second` where the tokens are synonyms only given certain contexts (of high-school or university) - a low similarity results in the incorrect prediction of paraphrase pairs.

Conclusion:

In conclusion, our exploration of various methods for computing word, phrase, and sentence similarity has provided us several insights into this specific task. Due to time constraints, it was unfortunately not possible for us to explore the possibilities of zero-shot/ few-shot training with Large Language Models and to check the performance of fine-tuned transformers for this specific task.

Bibliography:

1. Hill, Felix, Roi Reichart, and Anna Korhonen. "Simlex-999: Evaluating semantic models with (genuine) similarity estimation." *Computational Linguistics* 41.4 (2015): 665-695

2. Pham, Thang M., et al. "PiC: A Phrase-in-Context Dataset for Phrase Understanding and Semantic Search." *arXiv preprint arXiv:2207.09068* (2022).

3. Zhang, Yuan, Jason Baldridge, and Luheng He. "PAWS: Paraphrase adversaries from word scrambling." *arXiv preprint arXiv:1904.01130* (2019).

4. C.E. Osgood, Suci G., and P. Tannenbaum. 1957. The measurement of meaning. University of Illinois Press.

5. Mohammad, Saif. "Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words." *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)*. 2018.

6. Flynn, Terry N., and Anthony AJ Marley. *Best-worst scaling: theory and methods*. Diss. Edward Elgar, 2014.

7. Zhang, Tianyi, et al. "Bertscore: Evaluating text generation with bert." *arXiv preprint arXiv:1904.09675* (2019).

8. Church, Kenneth Ward. "Word2Vec." *Natural Language Engineering* 23.1 (2017): 155-162.

9. Wu, Zhibiao, and Martha Palmer. "Verb semantics and lexical selection." *arXiv preprint cmp-lg/9406033* (1994).

10. Lesk, Michael. "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone." *Proceedings of the 5th annual international conference on Systems documentation*. 1986.

11. Del Corro, Luciano, and Rainer Gemulla. "Clausie: clause-based open information extraction." *Proceedings of the 22nd international conference on World Wide Web*. 2013.

12. Wang, Wenhui, et al. "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers." *Advances in Neural Information Processing Systems* 33 (2020): 5776-5788.