Imposing WordNet Relations on Distributional Word Embeddings

Kushal Arora
School of Computer Science
McGill University
{kushal.arora,

Aishik Chakraborty
School of Computer Science
McGill University
aishik.chakraborty}@mail.mcgill.ca

Abstract

In this project, we propose an approach to impose the notion of perceptual relations on the word embeddings. We believe this will help overcome the side-effects of the distributional approach which tends to learn relations from the context and confuses correlation for semantic relatedness. We achieve this objective by projecting the embeddings in a relation subspace where specific relations like synonyms and antonyms are learned. We empirically show that this auxillary objective of learning a perceptual relation subspace helps the original embeddings on a suite of word analogy tasks. The code for conducting all experiments isased [1].

1 Introduction

One of the biggest success on deep learning based methods in natural language processing is learning the word embeddings which exploits the distributional hypothesis. These pre-trained embeddings have proven to improve the accuracy a whole range of neural network based model for language processing tasks ranging from NER and Chunking (Turian et al., 2010; Guo et al., 2014), Machine Translation (Mikolov et al., 2013a) and Sentiment Analysis (Socher et al., 2013). In addition to this practical application, these word embedding s have shown syntactic and semantic regularities (Mikolov et al., 2013b) which have proven to help identifying the semantically related word-pairs (Agirre et al., 2009).

The biggest strength of the embedding methods, their ability to capture the distributionally related word has also been it's biggest criticism. The implicit clustering of words used in similar context is not always desirable. For example, consider the case of adjectives. The words like good and bad, loud and quiet, beautiful and ugly are used in the similar context but express opposite emotions. The representation relying on distributional methods will implicitly cluster embeddings for these words but for tasks like sentiment analysis, it will be desirable to have these embeddings further apart in latent space.

We propose an alternate approach towards training word embeddings that additionally imposes the notion of perceptual relatedness to distributionally learned embeddings. Humans learn this perpetual relations in an embodied way and by common sense reasoning. The distributional methods are by design unable to capture these aspects from language.

Lexical databases like WordNet (Miller, 1995) are manually curated knowledge bases that symbolically encode the perceptual relations like synonymy, antonymy, hyperonymy, hyponymy and troponymy etc. Explicitly endowing these relation to distributional embeddings might help us in the tasks like sentiment analysis and natural language inference.

In this project, we formulate the problem of learning these perceptual relations along with distributional relations as learning the word embeddings space X such that, for every perceptual relation R, there exists a subspace S_R where that relation holds. These subspaces are not a part of the word embeddings but are recoverable by learning a linear projection F_R from the original word embedding space X to S_R. In the simplest of formulation $S_R = X$ but this might hamper with the model's ability to capture distributional similarity. We view learning these relation in the subspace S_R as a sort of regularization on the embedding to discourage it from learning the distributional relations which might not agree with the perceptual relations.

Experimentally, we show that learning these

[1] https://github.com/aishikchakraborty/LM_with_Wordnet

subspaces doesn't impact the original embedding as the perplexity of the LM objective is just marginally worse. We show that the word embedding learned by our model does consistently better at a suite of word analogy tasks despite not learning the relations directly. We also evaluated our model on downstream task and report that it did worse than original LM formulation on sentiment analysis task and the textual entailment task.

2 Related Work

There has been a number of attempts to impose perceptual relations or relational semantics on distributional embeddings. The approach closest to our is by Yu and Dredze (Yu and Dredze, 2014). In their paper, the authors jointly learn the perceptual relations and distributional embeddings but treat perpectional relations as prior. Also, their model can only deal with the symmetric relations whereas our approach can also deal with asymmetric relations like hyponymy.

Another approach is by Fried and Duh (Fried and Duh, 2014), where they add an auxillary objective to language modeling objective to ensure the distance between the word embeddings agrees with the distance among the words in WordNet. Their approach only focuses on hypernymy relation whereas our approach can work with all possible relations in WordNet.

In their work, Kiela et al. (Kiela et al., 2015) also agument the LM objective but their approach is limited to learning synonyms and word associations. Our model on the other hand can also learn adversarial relations like antonyms and part relations like meronymy.

Mrksic et al.'s (Mrkšić et al., 2016) approach of increasing the distance between the antonyms and clustering the synonyms together is similar to our approach but synonym and antonym relation subspace is only one subspace in our model. We also propose different subspaces to learn other WordNet relations. Additional point of difference between our model and theirs is that their approach is post-hoc whereas we learn the relation while training LM objective. Also, we learn the relation in a subspace different from the embedding space whereas in their case the synonym and antonym relation are learned in the embedding space.

3 Model

3.1 Neural Language Modelling.

Given a set of tokens in a corpus $C = (x_1, x_2, ..., x_n)$, we use a language model to maximize the following log likelihood function:

$$L_{LM}(C) = \sum_{i=1}^{n} \log P(x_i | x_{i-k}, ..., x_{i-1}, \theta) \quad (1)$$

where k is the size of the context window under consideration, and the conditional probability P is modeled using a neural model having θ parameters. These parameters are trained using the gradient ascent algorithm.

3.2 Introducing WordNet Relationships into LM objective.

WordNet has a hierarchical structure having various relationships between words. One of the interesting relationships present in WordNet includes synonymy/antonymy relationships which encodes how similar or how dissimilar is a pair of words. Another relationship of interest for our study is the hyponymy/hypernymy relationship.

Synonymy Relations. Between any two pair of words which are synonyms, w_x and w_y, we want to ensure that their representation in some subspace is similar. Therefore, we project the word embedding $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$ of w_x and w_y into a h-dimensional subspace X', where we ensure that x and y are very similar. In the current setting, we consider only linear projections, i.e., there is some matrix W of dimension $h \times d$ such that $Wx \in \mathbb{R}^h$ and $Wy \in \mathbb{R}^h$.

Given a piece of text, for each word x, we obtain all synomyms y from WordNet. Let the set of all (x, y) pairs be S. Thus, in a piece of text, the loss for N pairs of words in synonymy relationships is the euclidean distance between Wx and Wy and is given as follows

$$L_{syn} = \frac{1}{N} \sum_{(x,y) \in S} \|Wx - Wy\|_2^2 \quad (2)$$

Antonymy Relations. Between any two pair of words which are antonyms, w_x and w_y, we want to ensure that their representation in some subspace is far apart. We use a similar concept as above, expcept, here instead of simple Euclidean distance, we use a max-margin loss. Therefore, we project the word embedding $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$ of w_x

and w_y into a h-dimensional subspace X', where we ensure that x and y are far apart. In the current setting, we consider only linear projections, i.e., there is some matrix W of dimension $h \times d$ such that $Wx \in \mathbb{R}^h$ and $Wy \in \mathbb{R}^h$.

Let, given a piece of text, for each word x, we obtain all antonyms y from WordNet. Let the set of all (x, y) pairs be A. Thus, in a piece of text, the loss for N pairs of words in antonymy relationships is given as follows

$$L_{ant} = max(0, \gamma - \frac{1}{N} \sum_{(x,y) \in A} \|Wx - Wy\|_2^2) \tag{3}$$

where γ is the margin in the max-margin loss. We set $\gamma = 1$ for our experiments in this work.

Hypernymy Relations. Between any two pair of words which are hypernyms, w_x and w_y, we want to ensure that there is a subspace which encodes all hypernymy relationships through which we can transform w_x to w_y. Therefore, we project the word embedding $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$ of w_x and w_y into a h-dimensional subspace X'. Let there be another matrix V of dimension $h \times h$, such that VWx and Wy are similar. Thus, V encodes the hypernymy relationships between Wx and Wy.

Let, given a piece of text, for each word x, we obtain all hypernyms y from WordNet. Let the set of all (x, y) pairs be H. Thus, in a piece of text, the loss for N pairs of words in hypernymy relationships(y is hypernym of x) is the euclidean distance between Wx and Wy and is given as follows

$$L_{hyp} = \frac{1}{N} \sum_{(x,y) \in H} \|VWx - Wy\|_2^2 \tag{4}$$

Augmented LM Loss. Given the above WordNet relationships that we have considered, we write down the total loss function which we optimize during training.

$$L_f = L_{LM}(C) + L_{syn} + L_{ant} + L_{hyp} \tag{5}$$

Due to constraints of time to run the experiments, we use only the synonymy and antonymy relations and do not consider the hypernymy relations for our experiments. Thus the final augmentented loss we consider for pretraining the language models is as follows:

$$L_{final} = L_{LM}(C) + L_{syn} + L_{ant} \tag{6}$$

Model	Test Perplexity
Vanilla LM	108.62
Augmented LM	109.90

Table 1: LM perplexity on the wikitext-2 test set

We use L_{final} to compute gradients and train our network.

4 Experiments

4.1 Datasets

For the purpose of pretraining a language model, we choose the wikitext-2 word level corpus (Merity et al., 2016). The dataset contains about 2 million tokens in the training set and about $200k$ tokens in the validation set and another $200k$ in the test set. The vocabulary size is $33,278$. Due to limited time, we did not pretrain on larger corpora. We pretrain two language models, one using the normal log likelihood loss ($L_{LM}(C)$) and another using the augmented LM loss L_{final} on the wikitext-2 corpus. To learn the representations learnt, we evaluate on a number of intrinsic and extrinsic tasks.

Hyperparameters We learn 300 dimensional embeddings, use 700 hidden units in the LSTM and use a dropout value of 0.5 for all models. We use Adam as our optimizer with learning rate 0.001. We also use a learning rate scheduler to reduce the learning rate by a factor of 0.5 when the learning plateaus. We do not perform any hyperparameter tuning in this work.

4.2 Intrinsic Evaluation

Language Modelling Perplexity.

As a very simple evaluation, we evaluate both the language models trained in the wikitext-2 test set. From Table 1, we can see that the we achieve an almost similar perplexity on the test set. The augmented LM performs a bit worse than the traditional LM using the same set of hyperparameters.

Word Similarity Task.

We used nine popular test sets to evaluate word similarity. The datasets include WordSim Similarity and WordSim Relatedness((Zesch and Gurevych), (Agirre et al., 2009)), MEN dataset by (Bruni et al., 2014), Mechanical Turk dataset by (Radinsky et al., 2011), SimLex-999 dataset by (Hill et al., 2015), semeval 17 task 2 by

Metric	Vanilla LM	Augmented LM
men_test	0.15	**0.19**
radinskymturk	**0.22**	0.21
semeval17task2_test	**0.20**	**0.20**
simlex999	0.09	**0.11**
simverb3500	0.07	**0.08**
wordsim353_relatedness	0.02	**0.08**
wordsim353_similarity	0.28	**0.37**
yangpowersverb130	0.26	**0.29**

Table 2: Word Similarity Results on various word similarity test sets

Model	5-class Accuracy	3-class Accuracy
BCN(Vanilla LM)	48.46%	90.54%
BCN (Augmented LM)	45.92%	89.72%

Table 3: SST Classification Results

Model	Accuracy
ESIM(Vanilla LM)	83.46%
ESIM(Augmented LM)	83.10%

Table 4: SNLI Results

(Camacho-Collados et al., 2017), verb dataset by (Yang and Powers, 2006) and simverb 3500 by (Gerz et al., 2016). These datasets contain word pairs together with human-assigned similarity scores. We measure the Pearson's correlation between the cosine similarity(using the embeddings of the word pairs) and the human evaluations. Table 2 contains all results for the two types of word embeddings. As we can see, we get better Pearson's correlation across test sets(except the Randinsky Mechanical Turk test set) for the Augmented LM embeddings.

4.3 Extrinsic Evaluation

Sentiment Analysis. We use the Biattentive Classification Network(BTN) by (Peters et al., 2018) to train a sentiment classifier. We create two versions of these models, one where we use the Vanilla LM embedding layer as the pretrained embeddings and one where we use the Augmented LM embedding layer as the pretrained embeddings. We train and evaluate the models on the Stanford Sentiment Treebank (Socher et al., 2013). The results for both models are listed in Table 3. As we can see, our model does worse than the original formulation in terms of both 3-class and 5-class accuracies.

Textual Entailment. For textual entailment, we use the ESIM model by (Chen et al., 2016) for our experiments. We create two versions of these models, one where we use the Vanilla LM embedding layer as the pretrained embeddings and one where we use the Augmented LM embedding layer as the pretrained embeddings. We train and evaluate the models on the SNLI dataset (Bowman et al., 2015). The results for both models are listed in Table 4. As we can see, our model does worse on the textual entailment task as compared to the original formulation.

5 Conclusion

We created a novel way to augment the language modelling objective with WordNet relationships. We did evaluated our models on both intrinsic and extrinsic tasks and found that the the augmented objective does better than the language modelling objective in the word similarity tasks. However, our model fails to do well in the extrinsic tasks.

Shortcomings and Future Work. One of the shortcomings of our model is that we train on a small dataset with only 2 million tokens and $33k$ vocabulary size. We wish to train a much larger model in the future using the much larger wikitext-103 corpus, which has 103 million tokens in the training set and a much larger vocabulary. We also plan to conduct experiments with the hypernymy relations introduced. Also we plan to run our models on more extrinsic tasks to further test the quality of representations learnt. Also in our current work, we used only linear projections. It would be great to explore non-linear projections as well and see how the space in which we project the embeddings affect the downstream performance.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009*

Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 19–27. Association for Computational Linguistics.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Daniel Fried and Kevin Duh. 2014. Incorporating both distributional and relational semantics in word representations. *arXiv preprint arXiv:1412.4369*.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. *arXiv preprint arXiv:1608.00869*.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 110–120.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2044–2048.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Dongqiang Yang and David Martin Powers. 2006. *Verb similarity on the taxonomy of WordNet*. Masaryk University.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 545–550.

Torsten Zesch and Iryna Gurevych. The more the better? assessing the influence of wikipedia's growth on semantic relatedness measures.