Learning embedding matrices using linguistic constraints and context similarity for sub-word units

Barun Patra
bpatra@andrew.cmu.edu

Daniel Martin
dlmartin@andrew.cmu.edu

Haitian Sun
haitians@andrew.cmu.edu

Yang Zhang
yz6@andrew.cmu.edu

Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213

1 Introduction

Computing continuous vector representations for words using large corpuses of data has repeatedly shown to be useful in numerous NLP tasks. However, this technique faces numerous problems:

- Poor embeddings for words rarely found in the training corpus.
- Much linguistic "common sense" knowledge is not leveraged, both at the word level (synonyms, antonyms) and morpheme level (prefix modifiers etc.)
- Most words are restricted to a single embedding (i.e., issues of polysemy are not handled).

In this project, we intend to improve on current embedding techniques with respect to the first two issues. Specifically, we aim to learn embedding matrices for word sub-units, and then compose these matrices to form more effective embeddings for words. We wish to extend the commonly-used Skip-gram model to make use of word sub-units, as well as take into account the "common sense" linguistic knowledge during training such as synonyms and antonyms. Moreover, we are looking to move away from whole-word embedding matrices; instead, we aim to learn embedding matrices for sub-units, and treat word formation as a matrix product. This would allow us to model prefixes like "un" and "dis", which can entirely change the meaning of root words when combined. Finally, we will evaluate our proposed method against a set of established benchmarks and use it in a downstream task to measure efficacy.

2 Related Work

Word vectors have been immensely useful in numerous NLP tasks since their introduction in [1]. Recent work in improving these vectors has been in two directions: to use sub-word units and to incorporate linguistic information. In their recent work, [2] try incorporating sub-unit knowledge. Specifically, they learn embeddings of N-grams (N$\in \{3, 4, 5, 6\}$), and then treat each word as the sum of the embeddings of all its N-grams. As mentioned earlier, a summation approach cannot model morphological derivations like "pro" which strengthen the subsequent word, or prefixes like "un" and "dis" which lead to the reversal of the subsequent word. Consequently, we move to embedding matrices in our approach; by representing words as products of their constituent matrices, we would be able to capture said derivations.
[3] and [4] try and use linguistic knowledge of synonyms and antonyms to improve vector representation. In [3], the authors use a dictionary of synonym and antonym pairs extracted from PPDB

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2.0 [5] and WordNet [6]. In [4], the authors improve on their previous work by extracting the synonym-antonym pairs using simple lexical rules. Nevertheless, in both cases, the incorporation of synonym-antonym knowledge is done as a post-processing step. The embeddings are first learned using contextual information, and then fine-tuned using the aforementioned approach. We hypothesize that incorporating the synonym-antonym information while learning from the context itself would allow us to learn better embeddings. Concretely, we hope that if there exist words that are not present explicitly in the synonym-antonym dictionary, but share context (and vice-versa), they would be mapped to the same region in the embedding space.
[7] is the closest in spirit to our work, wherein the authors use a deep recursive network to learn sub-unit embeddings. However, the training strategy used by the authors is very different from the one we propose to use; moreover, the authors do not make use of any linguistic knowledge.

3 Approach

We propose to combine the two aforementioned methods, i.e use synonyms as supplement to the context and antonyms as opponents alongside negative sampling, which is equivalent to minimizing the following loss function :

$$\begin{aligned}\mathcal{L}(w, C, S, A, N_c) = &\sum_{w_c \in C} l(-s(E(w), E(w_c))) + \sum_{w_s \in S} l(-s(E(w), E(w_s))) \\ &+ \sum_{w_a \in A} l(s(E(w), E(w_a))) + \sum_{w_n \in N_c} l(s(E(w), E(w_n))) \\ E(w) = &\prod_{m \in \text{tok}(w)} E(m)\end{aligned}$$

Where, C is the context set, S is the synonym set, A is the antonym set, N_c is the randomly sampled negatives set, s is the scoring function (higher means more similar), l is the logistic loss $l(x) = \log(1 + e^{-x})$, and E is the embedding generating function, i.e E : word $\mapsto \mathbb{R}^{n \times n}$. The function tok$(w)$ yields the sub-units of the words. Instead of using all the N-grams (N$\in \{3, 4, 5, 6\}$), as done by [2], we would experiment with using Byte Pair encodings [8] and morphemes, as detected by Morfessor 2.0 [9], an unsupervised morphological segmentation toolkit. Note that we use matrix multiplication instead of vector sum, which allows us to model morphological derivations.

We would be using a subset of the English Wikipedia dump for training the embeddings. We would also be using the synonym-antonym set present in PPDB 2.0 [5] and the antonym set from Wordnet [6].

4 Evaluation

We wish to evaluate the model on the following tasks:

- Correlation with similarity scores from the SimLex-999 dataset [10] and the SimVerb-3500 dataset [11]. Both datasets contain pairs of words scored by humans and are standard datasets used to measure efficacy of embedding methods.
- Performance on the Word Analogies dataset, as introduced by [1], and by inspecting T-SNE[12] representations for sanity checks
- Correlation on the rare word dataset, as introduced by [7]. This dataset consists of rare words that can be decomposed into their more-common constituent sub-units, thereby allowing us to check the effectiveness of our sub-unit approach.

For baseline comparison, we intend to use the traditional Word2Vec (CBOW and Skip-gram) models ([1]), models using synonym-antonym sets ([3] and [4]) and models using the sub-unit embeddings ([2] and [7]).

References

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou,

M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

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

[4] Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. *arXiv preprint arXiv:1706.00377*, 2017.

[5] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. 2015.

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

[7] Thang Luong, Richard Socher, and Christopher D Manning. Better word representations with recursive neural networks for morphology. In *CoNLL*, pages 104–113, 2013.

[8] Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. Technical report, Technical Report DOI-TR-161, Department of Informatics, Kyushu University, 1999.

[9] Mathias Creutz, Krista Lagus, Krister Lindén, and Sami Virpioja. Morfessor and hutmegs: Unsupervised morpheme segmentation for highly-inflecting and compounding languages. 2005.

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

[11] Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. In *EMNLP*, 2016.

[12] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.