SemVec: Semantic Features Word Vectors Based Deep Learning for Improved Text Classification: 7th International Conference, TPNC 2018, Dublin, Ireland, December 12–14, 2018, Proceed…

Chapter · January 2018

DOI: 10.1007/978-3-030-04070-3_35

CITATIONS
2

READS
62

2 authors, including:

Adel Taweel
Birzeit University
134 PUBLICATIONS **1,254** CITATIONS

SEE PROFILE

SemVec: Semantic Features Word Vectors Based Deep Learning for Improved Text Classification

Feras Odeh[0000−0001−9330−3411] and Adel Taweel[0000−0003−0240−9857]

Birzeit University, Birzeit, Palestine
ferasodh@gmail.com ataweel@birzeit.edu

Abstract. Semantic word representation is a core building block in many deep learning systems. Most word representation techniques are based on words angle/distance, word analogies and statistical information. However, popular models ignore word morphology by representing each word with a distinct vector. This limits their ability to represent rare words in languages with large vocabulary. This paper proposes a dynamic model, named SemVec, for representing words as a vector of both domain and semantic features. Based on the problem domain, semantic features can be added or removed to generate an enriched word representation with domain knowledge. The proposed method is evaluated on adverse drug events (ADR) tweets/text classification. Results show that SemVec improves the precision of ADR detection by 15.28% over other state-of-the-art deep learning methods with a comparable recall score.

Keywords: Convolution Neural Networks, Deep Learning, Text classification, Word embeddings, features engineering

1 Introduction

Semantic word representation is a key building block in a variety of deep learning systems. In semantic vector space models, each word is represented with a real-valued vector. By representing words with real-valued vectors they can be used as features in a variety of tasks, such as relation extraction [26], named entity recognition [11], question answering [10] and document classification [23].

Most word vector representation techniques represent each word as a vector based on the angle or the distance between word vectors [17]. Recently, Mikolov et al. [14] have developed a new model architecture for learning continuous word representation based on word analogies that preserve words linear regularities named word2vec. Word2Vec is widely used in several deep learning tasks like text classification [10,1,5,12,18] and relation extraction [19].

The three different model families for learning word vectors that are mostly used in deep learning are: 1) pre-trained word vectors, such as word2vec [14] and GloVe [17], 2) region embedding, which generates embedding of small text region instead of each word in isolation, e.g. [9,22] and 3) character embedding which generates embedding from characters, such as [3,25]. These methods aim to generate vector representations that encompass semantic relations between words.

However, most of these ignore the morphological structure of words, which is very important in some languages, such as Finnish [3]. Also, some domain words or word forms rarely occur in training data, or often occur to carry specific semantics in a particular domain, which yields a bad word representation. Such bad word representations of domain words can have clear impact on model performance. Moreover, it is not possible to improve vector representation in such methods for specific languages or domains [3]. However, many domains often use specific words that carry specific semantic representation and common meaningful use, which if taken into consideration may improve text classification. In addition, some domains often employ particular well-defined lexicon and dictionaries, e.g. medical domain, which can be employed to further improve text classification.

Thus, this paper proposes a dynamic model that focuses on the use of semantic features and their word representation. It represents words as a vector of domain-specific and morphological features, employing those with domain semantic relevance and common meaningful use. This model is dynamic, which means its domain features are changeable, i.e. they can be added in or removed from, the model, based on the problem domain being applied to.

The paper is organised as follows: Section 2 covers related work, Section 3 describes the proposed approach, Section 4 describes the designed model aarchitecture, Section 5 and 6 details the conducted experiment design and evaluation of the proposed approach, Section 7 reports the results and their analysis and finaly Section 8 concludes the paper.

2 Related Work

Features engineering and machine learning algorithms for are widely studied on text classification[21,6,16]. Recently, a set of algorithms were introduced that propose a representation of words in a vector space model by grouping similar words. One of the popular algorithms is word2vec, introduced by Mikolov et al.[14]. Word2vec is a shallow, two-layer neural network pre-trained on large amounts of unstructured text data; Word2vec produces high-quality vectors, typically of several hundred dimensions. However, it has a number of issues. Firstly, while word2vec utilizes a word's local context window, Pennington et al. [17] proposed a different model that combines the advantages of local context window and global matrix factorization methods. Both of these models, however, fail to provide a good representation for rare words, such as domain words, which can affect performance. To address, the proposed model utilizes lexicon and dictionaries to handle rare and domain words. Secondly, word2vec represents words in isolation, Johnson et al. [9] proposed a different approach that generates embeddings of small text regions - instead of words - from unlabeled data. Johnson and Zhang [8] proposed a two-view region embedding approach that is trained to predict co-presence and absence of words in a region. However, region embedding methods cannot guarantee the classification performance for very short texts. Wang et al. [22] proposed a weighted region embedding based on region information importance, which focused on modeling short text only.

To address, the proposed approach can be used to model both short and long text documents. Closest to our approach is Sahu et al.[19], that proposed a new approach for relation extraction by combining word2vec and features engineering. In contrast to Sahu et al. approach, our model represents words as a vector of domain features only without word2vec or GloVe.

For Adverse Drug Reactions (ADR) classification, Sarker et al.[21] proposed an approach that utilizes multi-corpus to improve classification performance for classifying Twitter posts. They used three datasets, two of them are annotated posts from social media while the third one contains annotated clinical report sentences. Their proposed method generates a feature vector for each sentence from a large set of semantic features (i.e: sentiment, polarity and topic) and from short text nuggets, whereas our proposed approach generates vector of domain features for each word. Akhtyamova and Alexandrov [1] conducted work on unsupervised word embeddings learning from different datasets including GoogleNews, Wikipedia, and Diego lab. Unlike word embeddings learning, Huynh et al.[7] proposed two new neural network models: Convolutional Recurrent Neural Network (CRNN) by concatenating recurrent neural network to convolution neural network and Convolutional Neural Network (CNN) with Attention (CNNA) by adding attention weights into CNN. While Lee et al. [13] did not propose a new CNN architecture, but they proposed a semi-supervised CNN model which uses several semi-supervised CNN models built from different types of unlabeled data. The CNN model is trained with annotated ADE data. The output layer uses a linear classifier that can classify whether a tweet contains an ADR or not.

All these methods use word embedding, such as word2vec and GloVe, by contrast, our approach presents a new word vector representation model that can be used as the input layer to CNN. It uses semantic features only as a word vector representation. The remainder of this paper describes SemVec, including both the proposed word vector model and convolutional architecture. It then reports experiments setup and evaluation results, of SemVec, on the Twitter ADR corpora.

3 SemVec: The proposed Approach

The proposed approach generates a word representation model using domain-specific features. As shown in figure 1, SemVec represents each word in each sentence as a row vector of features. For example, the word "He" is represented as a vector of feature values:

$$x = f_1^1 \oplus f_2^1 \oplus f_3^1 \oplus f_4^1 \oplus f_5^1 \oplus f_6^1 f_7^1 \oplus f_8^1 \oplus f_9^1 \oplus f_{10}^1 \oplus f_{11}^1 \oplus f_{12}^1 \oplus f_{13}^1$$

Here $\oplus$ is concatenation operation so $x \in \mathbb{R}^{1\times 13}$. It is worth noting that each feature can be represented by one or more dimensions in the resulting word vector. Both number and type of features can be changed based on the domain, task and dataset. A sentence is represented as a matrix of words X features. In order to have a unified number of matrix rows, SemVec sets the max number of words and zero pad to any matrix that has a smaller number of words.

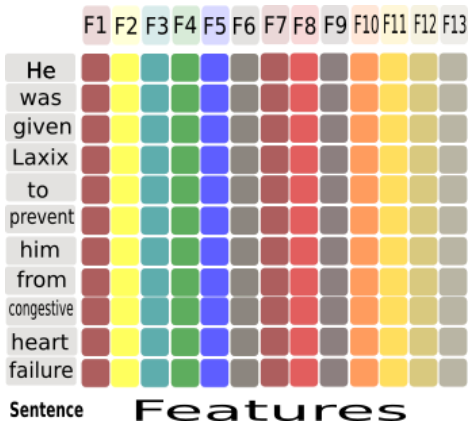

Fig. 1. Each word in this sentence matrix is represented as a row of 13 features.

3.1 Features Extraction

SemVec represents each word with a vector of features. Based on problem domain, additional features can be added or removed. The following list shows these features and how they are represented in SemVec:

Table 1. Features samples (* denotes an ADR domain specific words/sentences)

Feature	Samples
Negative Word	*abnormal**, annoying, *ache**, aggressive, sue, *suicidal**, zombie
Positive Word	abundance, adequate, awesome, fascinating
ADR Lexicon	*infection vascular**, *fecal fat increased**, *luteinizing hormone decreased**
More Words	enhance, augment, increase, amplify
Less Words	drop, fewer, slump, fall, down
Good Words	beneficial, improve, advantage, resolve
Bad Words	complication, risk, *adverse**, *chronic**, *bleeding** , morbidity

- Opinion Lexicon Negative Words (F1) / Positive Words (F2): represent a list of negative/positive English opinion words. Hu and Liu [6] have proposed a list of negative opinion words that can be used in sentiment analysis. The list contains 4817 negative/positive words. Table 1 shows a sample of Opinion Lexicon Negative Words.
- SentiWordNet Lexicon Positive Words(F3) / Negative Words(F4): represent a positive/negative English sentiment word. Baccianella et al. [2] have proposed

the SentiWordNet v 3.0, which contains 118,000 English words associated with positive/negative sentiment score between 0 and 1.
- Subjectivity(F5): represents the English language subjectivity value of words. Wilson et al.[24] have proposed a list of words with their subjectivity strength (weak and strong) and their polarity (negative, positive and neutral).
- More-Good (F7), More-Bad (F8), Less-Good (F9), Less-Bad (F10): Niu et al. [16] have proposed these four polarity features. A More-Good feature indicates more positive information in the sentence. This represents how a change happens: for example, reducing headache is considered as a positive outcome; on the other hand, increasing headache is considered as a negative outcome. These features try to find out when there is an increase/decrease in a good/bad thing. A collection of good, bad, more and less words were created and used by Sarker et al. [20]. In order to extract these features, a window of four words is processed on each side of a word. If a Good word was found in this window, then a More-Good feature is activated. A similar process is followed to activate other features.
- Word Length (F11): represents the number of each words characters. for example: amazing: 7, do: 2. The word length represents how complex each word is and how complex the sentence is. The following equation shows how to calculate this feature: $x = W * length(word)$. W is a positive integer that represents this feature weight.
- Word Order (F12): represents the order number of this word in the sentence. The following equation shows how to calculate this feature: $x = W * order(word)$. Where W is a positive integer that represents this feature weight.
- Word Clusters (F13): is proposed by Brown et al. [4]. In this feature, hierarchical word clusters are generated from a huge set of unlabeled tweets via Brown clustering. This produces a base set of 1,000 clusters for happiness, sadness and emotions. The following equation shows how to calculate this feature: $x = wordclusterkey(word)$.

4 Model Architecture

To find the optimal CNN model architecture layers and parameters for SemVec, the grid search method was used to conduct experiements repetitively until optimal values were found. The details of the conducted experiements are outside the scope of this paper.

4.1 Features input layer

In this model, each word is represented with the above identified 13 discrete features as shown in figure 1. The system generates a row vector of features for each word as described in section 3.

4.2 Convolution layer

In this model, a single convolution layer is used with a ReLu [15] activation function. SemVec model uses one filter of size 2*2. This filter size was found, by experimentation (outside the scope of this paper). This can be explained by the nature of text representation (as opposed to images). When words are convoluted, their larger dimension does not carry variations in representation, thus smaller dimension is as optimal as larger sizes,

4.3 Max pooling layer

In this layer, a max pooling operation is applied to find the most useful feature in the generated feature map from the previous layer. Similarly, the max pooling window size was set to 2*2 (found by experimentation, justified as described above).

4.4 Fully connected layers

SemVec model has 2 fully connected layers. The first layer includes 128 neurons with a ReLu activation function and the second one has one neuron. These values have also been found, by experimentation, to provide as optimal values as using a larger number of neurons in layers. Although details results of experimentations are outside the scope of this paper, but they can be similarly explained by the nature of text representation, as opposed to image representation, where granularity of representation does not carry as much variations.

4.5 Sigmoid layer

This layer performs binary classification by applying sigmoid function. The sigmoid equation is: $S(x) = \frac{1}{1+e^{-x}}$

5 Experiments

5.1 Implementation details

Our CNN model is coded in Python and trained using Keras deep learning library[1]. Hyper-parameters for our model were chosen based on the ADR class F-score of the test set. As mentioned above, a grid search was conducted to find the optimal values for each parameter listed below, while others their values were defaulted. The list of all the main model parameters is shown in table 3.

[1] http://keras.io/

5.2 Training and Test Sets

A standard method was used to evaluate the classification performance, in which the dataset was randomly split into a training set and test set. The dataset was divided into two parts: 80% training and 20% testing set. Stratified k-fold, which is a variation of k-fold, was used to return stratified folds. This ensures that each set contains approximately the same proportions of instances for each ADR and non-ADR class as the original complete set. In this experiment, 10 stratified folds were used. The classification model is built based on the training set only. The test set instance classes were hidden from the model in the training step.

5.3 Experiment Design

All experiments were conducted on Google cloud virtual machines running Debian GNU/Linux 9 (Stretch) operation system. Each virtual machine has 4 virtual CPUs and 3.6 GB of RAM.

Table 2. Experiment settings

Parameter Name	Parameter Value
BATCH_SIZE	128
DROPOUT_KEEP_PROB	0.7
EMBEDDING_DIM	32
FILTER_SIZES	2
NUM_FILTERS	32
KERNEL_SIZE	2
PADDING	same
LOSS	binary_crossentropy
OPTIMIZER	RMSprop

5.4 Dataset

In order to evaluate the performance of SemVec, a group of experiments was conducted on the publicly available Twitter ADR corpus[21]. This dataset represents a specific domain, i.e. medical domain in this case, and provides a reasonable gold standard dataset with manual annotation of ADRs, by domain experts. A total of 74 drugs from IMS Health's Top 100 drugs by volume for 2013 was used to collect this dataset tweets. Tweets were collected using the generic and brand names of drugs, including phonetic misspellings. The dataset is composed of a total of 7,574 instances, 6,672 of which do not contain ADRs, and only 902 include ADR mentions. This dataset is highly imbalanced as only 11.9% of the tweets has new ADRs but it also reflects a typical real-world tweet-generated data.

6 Evaluation

To evaluate the performance of our approach, 13 semantic features for each word from the Twitter ADR dataset as described in section 3 was used. Precision, Recall, and F1-score were used as the standard evaluation metrics to report the results for the ADE class. For comparison, classification performance of a number of supervised and semi-supervised classification models were implemented, these are described briefly below. **ADR Classifier** is a state-of-the-art supervised binary classifier [21] that uses a wide range of features derived from n-grams to UMLS semantic types that represent medical concepts. The authors reported achieving 59.7% F1-score when the model was trained and tested on multi-corpus. Since it was not possible to re-create the exact training/test data due to unavailability of daily strength corpus, the results are reported on the Twitter ADR dataset only.

CNN is a supervised convolutional neural network classifier trained only on labeled tweets [13]. We compare the performance of CNN and SemVec to determine if SemVec is improving ADR classification results.

CNN + Google News is a supervised convolutional neural network classifier with word2vec embeddings pre-trained on Google News [1].

CRNN, RCNN and CNNA are new neural network models. Both RCNN and CRNN combine recurrent neural networks and CNN; CNNA combines CNN with attention neural networks [7].

Majority Vote is a semi-supervised convolutional neural network model leveraging different types of unlabeled data for ADR classification [13]. This approach represents the new state-of-the-art f-score in ADR classification.

7 Results

In this section, our proposed approach is evaluated with several configurations of domain features. Our main contribution is a semantic-features-based word representation approach, which shows that a significant classification precision improvement on the state-of-the-art CNN architectures can be achieved on Twitter ADR classification task by using a smaller vector size of only 13 domain-specific features.

Table 3 shows the experimental results of SemVec on the Twitter ADR dataset compared to other states of the art methods. The CNNA approach [7] has the highest reported recall (66%) among all classification results, but also the lowest precision (40%). This shows that CNNA architecture is more suitable for ADR extraction, whereas CNN is more suitable for ADR classification as convolutions are enough to capture necessary information for ADR classification [7].

Better results were achieved using CNN and pre-trained Google News by [1]. This method achieved 54.2% F-score. This indicates that word2vec word representation is clearly affected by the pre-trained corpus. ADR classifier achieved a closer results to CNN and pre-trained Google News by [1], CNN and pre-trained Google News still achieved a lower F-score(53.62%). The ADR classifier uses

Table 3. Twitter ADR classification results

Method	ADR Precision	ADR Recall	ADR F-score
SemVec	77.93	49.68	60.48
CNN [13]	55.3	50	52.53
CRNN [7]	49	55	51
RCNN [7]	43	59	49
CNNA [7]	40	66	49
Majority Vote [13]	70.21	59.64	64.50
ADR-classifier [21]	N/A	N/A	53.62
CNN+GoogleNews [1]	N/A	N/A	54.20

features engineering and support vector machine. Majority Vote [13] method achieves the highest F-score compared to other methods. This model generates the region embeddings from both unlabeled tweets and health-related sentences. The final score is reported using Majority Vote for an ensemble prediction method.

SemVec achieves the best ADR precision score compared to other methods. We argue that this is due to the use of semantic features. Compared to Majority Vote SemVec requires less training time and resources. Majority Vote is a semi-supervised approach that generates embeddings in the unsupervised Pre-training step from different types of unlabeled data -in this experiment the authors created seven models from different datasets of unlabelled medical and non-medical domains-. In the second phase, a CNN is trained with embeddings generated from different models, and the ADE dataset. In order to apply Majority Vote to other domains, huge datasets of unlabeled data have to be collected and used to generate embeddings which limits approach scalability.

8 Conclusions and Future Work

The paper proposed an approach to create word vector representation by using semantic features. The proposed approach builds on the work that was introduced by Sahu et al. [19]. Using domain features, our model improved ADR detection precision, which is very critical in the field of ADR detection, over other state-of-the-art methods. Results show that SemVec outperforms other methods that do not take into account domain characteristics as well as methods that employ non-domain semantic or word representation such as word2vec/Glove.

SemVec was evaluated on short sentences or Twitter posts and thus results may not scale to other longer sentences. The evaluation of the proposed approach would benefit from conducting more experiments on different datasets to test the scalability of SemVec; however, the used dataset is considered a gold standard for ADRs, which is carefully and manually annotated by domain experts and thus provides a unique reference dataset. Creating similar different datasets would require enormous efforts.

In addition, this work would benefit from studying additional domain features, which may contribute to the classification performance. Further, creating a set of pre-trained word vectors, which can be used without additional feature engineering work, would improve the scalability of the approach.

References

1. Akhtyamova, L., Alexandrov, M., Cardiff, J.: Adverse drug extraction in twitter data using convolutional neural network. In: Database and Expert Systems Applications (DEXA), 2017 28th International Workshop on. pp. 88–92. IEEE (2017)
2. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC. vol. 10, pp. 2200–2204 (2010)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational linguistics **18**(4), 467–479 (1992)
5. Dos Santos, C.N., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: COLING. pp. 69–78 (2014)
6. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 168–177. ACM (2004)
7. Huynh, T., He, Y., Willis, A., Rüger, S.: Adverse drug reaction classification with deep neural networks. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 877–887 (2016)
8. Johnson, R., Zhang, T.: Semi-supervised convolutional neural networks for text categorization via region embedding. In: Advances in neural information processing systems. pp. 919–927 (2015)
9. Johnson, R., Zhang, T.: Supervised and semi-supervised text categorization using lstm for region embeddings. In: International Conference on Machine Learning. pp. 526–534 (2016)
10. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
11. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
12. Lee, J.Y., Dernoncourt, F.: Sequential short-text classification with recurrent and convolutional neural networks. arXiv preprint arXiv:1603.03827 (2016)
13. Lee, K., Qadir, A., Hasan, S.A., Datla, V., Prakash, A., Liu, J., Farri, O.: Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In: Proceedings of the 26th International Conference on World Wide Web. pp. 705–714. International World Wide Web Conferences Steering Committee (2017)
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
15. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
16. Niu, Y., Zhu, X., Li, J., Hirst, G.: Analysis of polarity information in medical text. In: AMIA annual symposium proceedings. vol. 2005, p. 570. American Medical Informatics Association (2005)

17. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. vol. 14, pp. 1532–1543 (2014)
18. Poria, S., Cambria, E., Gelbukh, A.: Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In: Proceedings of the 2015 conference on empirical methods in natural language processing. pp. 2539–2544 (2015)
19. Sahu, S.K., Anand, A., Oruganty, K., Gattu, M.: Relation extraction from clinical texts using domain invariant convolutional neural network. arXiv preprint arXiv:1606.09370 (2016)
20. Sarker, A., Aliod, D.M., Paris, C.: Automatic prediction of evidence-based recommendations via sentence-level polarity classification. In: IJCNLP. pp. 712–718 (2013)
21. Sarker, A., Gonzalez, G.: Portable automatic text classification for adverse drug reaction detection via multi-corpus training. Journal of biomedical informatics **53**, 196–207 (2015)
22. Wang, J., Yu, L.C., Lai, K.R., Zhang, X.: Dimensional sentiment analysis using a regional cnn-lstm model. In: ACL 2016Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany. vol. 2, pp. 225–230 (2016)
23. Wang, P., Xu, B., Xu, J., Tian, G., Liu, C.L., Hao, H.: Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing **174**, 806–814 (2016)
24. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. pp. 347–354. Association for Computational Linguistics (2005)
25. Xiao, Y., Cho, K.: Efficient character-level document classification by combining convolution and recurrent layers. arXiv preprint arXiv:1602.00367 (2016)
26. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 2335–2344 (2014)