Semantic enriched deep learning for document classification [1]

Abderrahmane LARBI, Sarra BEN ABBÈS, Rim HANTACH, Lynda TEMAL, and Philippe CALVEZ
CSAI LAB ENGIE, France

Abstract. Textual data are available in large unstructured volumes. Processing this data is becoming crucial and document classification is a way of structuring and processing this information based on its content. This paper introduces an effective semantic text mining approach for document classification. The proposed approach *Semantic Enriched Deep Learning Architecture* (SE-DLA) allows the model to learn simultaneously from the generated semantic vector representations and the original document vectors. We evaluated the proposed method on topic categorizations and multi-label classification. The experiments demonstrate that the proposed hybrid architecture with the additional semantic knowledge improves the results. This approach was compared to some state-of-the-art text classification approaches not including semantic knowledge. The proposed SE-DLA achieved higher accuracy and maintained great results during the experimental process.

Keywords. Semantic Classication, Textual documents, Deep Learning, Taxonomy

1. Introduction

Nowadays, an exponential growth in data-oriented technologies is rapidly produced an exploding volumes of data. This large data is produced every day. Textual data is unstructured and available as open-source format or in a more confidential manner within companies' ecosystems. Processing textual data is a major challenge for companies. Natural Language Processing (NLP) methods are applied to different tasks such as text classification, sentiment analysis, and more. Document classification is a way of structuring and processing this information based on its content. A text classifier is defined as a model that takes as input a set of labeled documents and learns how to associate the patterns appearing in a document to the appropriate label.
In this context, recent deep learning methods like Convolutional Neural Networks (CNN) [1,2] and Long Short Term Memory (LSTM) [3] have become the standard for learning tasks related to natural language and demonstrating high-performance results in classification tasks. Deep learning techniques for textual data generally used the textual documents as input for its architectures. However other existing approaches are based on external semantic resources like a taxonomy, an ontology, a dictionary, etc, in order to enhance the semantic context of the inputs and to benefit from a higher semantic level

features. Despite the fact that many semantic resources are available, deep learning architectures are still rare to use these semantic knowledge.
In this paper, we present a semantic approach of document classification that used an external semantic resource (WordNet) [4]. This resource is applied to create the semantic vector space representation, that will be employed in various deep learning settings.
This work is structured as follows: section 2 introduces the state-of-the-art related to different approaches of semantic text classification. Section 3 presents our proposed approach. Section 4 conducts a comparison study with the existing approaches. Finally, section 5 concludes our work and introduces eventual further works.

2. Related Work

This section introduces the background and outlines the related work in semantic document classification and deep learning techniques.

Document representation is the most important step in document classification frameworks. Many approaches have been presented in order to solve the document representation issues. Simple bag-of-words are statistical methods for document representation. These methods are often based on the frequency as a single feature (or similar Term frequency Inverse Document frequency methods [5]). More advanced approaches can group together words with similar meanings. These methods are often based on dimensionality reduction techniques.
Latent Semantic Analysis (LSA) [6] and Latent Dirichlet Allocation (LDA)[7] are some of the dimensionality reduction available methods. More methods are recently introduced the word embedding techniques commonly used to transform documents into a lower-dimensional vector space representation. Word2vec [8] is one of the used methods for word embedding. It captures a semantic concept using a two-shallow neural network. The semantic context captured by Word2vec [9] is limited, this is why approaches like GloVe [10] and FastText [11] were developed. The generated feature vectors are positioned closely if they are similar contextually. Recently both the LSA-based approaches and embedding knew a significant improvement in the predictive power and also in terms of scalability.

In [12], the author demonstrated that context-aware approaches outperform the naive approaches. The neural network-based approaches in dealing with the classification task, capture a context while learning word representations, this approach can be referred to as a first level context.

Background semantic knowledge from external semantic resources (ontologies or taxonomies) can be incorporated into the learning phase, this process is referred to as a second-level context. This process can lead to a significant improvement in the performance of semantic-aware frameworks. As presented in [13] and [14] the second-level context improves also the semantic discovery tasks.

In text mining, [15] reports an ontology-based web document classifier, while [16] proposes a clustering-based algorithm for document classification, which also benefits from the knowledge stored in the underlying ontologies.

Cagliero *et al* [12] present a custom classification algorithm, which can leverage taxonomies and demonstrates a case study of geospatial data that such information can be used to improve the classification. In this paper [17], the authors have demonstrated that

the Word embedding approaches can take into account semantic-specific information to improve the classification.

Ristoski *et al* [18] show that embeddings-based approaches are useful for taxonomy induction and completion. Liu *et al* [19] addresse the incorporation of taxonomy-derived background knowledge as a constrained optimization problem. Bian *et al* [20] present a leverage morphological, syntactic, and semantic knowledge to achieve high-quality word embeddings and prove that knowledge-powered deep learning can enhance their effectiveness.

Not long ago, deeper neural network architectures have proven their performance for word embedding on classification purposes [1,21].

This section introduces techniques based on machine learning and deep learning architectures.

Word2vec [9] introduced previously is built on a two-layer neural network. Recent studies demonstrated that deeper architectures tend to give better results on the document classification tasks. Deep neural networks (DNN) are designed to learn from the multi-connection of layers. Every single layer receives the connection from the previous layer and provides connections to the following layers in a hidden section. The deep aspect of the DNN comes from the number of hidden layers. Zhang and Kim [1,2] introduce an approach based on deep convolutional neural networks (CNN). A set of vectors containing word indexes or similar input representations are directly used to predict the classes. Convolutions are introduced and used efficiently for text and document-related learning and precisely in document classification tasks [22]. Figure 1 presents a classification task conducted with a CNN, it takes as input an embedded form, performs convolutions, and finally the dense step for the classification.

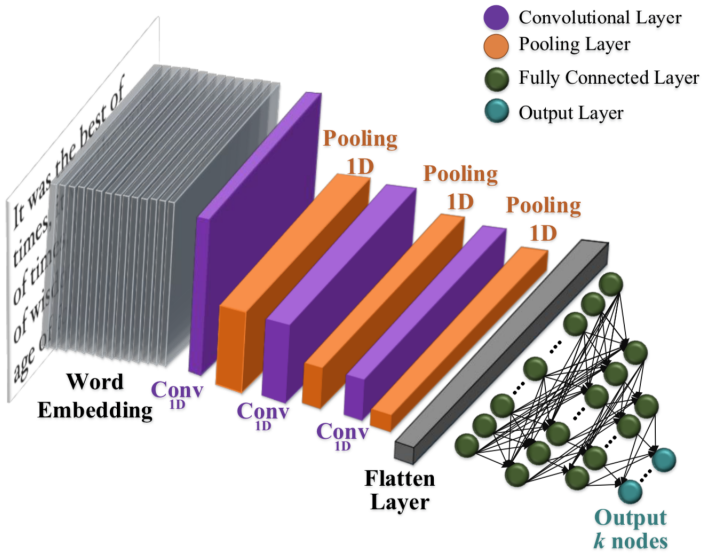

Figure 1. Convolutional neural network (CNN) architecture for text classification [23]

Unlike standard DNNs that handles fixed-sized inputs, Recurrent neural networks (RNN) are designed to take a series of input with no predetermined limit on size. [3]

introduced firstly the Long short term memory (LSTM) that many researchers improved afterward. To properly address the problem of preserving the long-term dependency in an effective way compared to the way that traditional RNNs handles it, the LSTM was typically introduced. This architecture deals properly with the complex problem of the gradient vanishing. Using its numerous gates, an LSTM cell regulates considerably the amount of the information going in and out of it. We can observe in Figure 2 an LSTM cell with its different gates.

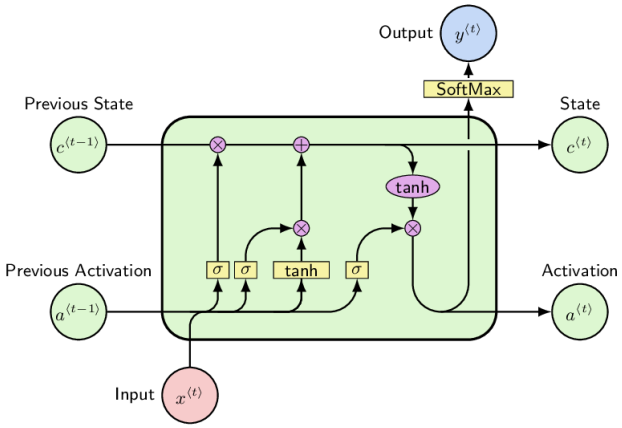

Figure 2. LSTM Cell [24]

Recent techniques like char-level CNN [1] proposed by Zhang *et al* are based on using a character level embedding before applying convolutional networks to perform the classification. An empirical study was conducted demonstrating that character level ConvNets can achieve state-of-the-art results.

Kim and Yang presented in [25] an approach based on a model named Seq2CNN. This approach is divided into two blocks. A sequential block, this block summarizes the input texts and a convolutional block that classifies the summarized text.

Gupta *et al* [26] presented a recent approach named Sparse Composite Document Vector Multi-Sense(SCDV-MS) in their paper named Improving Document Classification with Multi-Sense Embeddings. They proposed an approach that addresses the problem of higher dimensionality representations. This approach uses multi-sens embedding learning lower-dimensional manifolds. This is an interesting approach in terms of time and space complexity.

Another work that uses semantic information is led by [27] under the name of *Towards Robust Text Classification with Semantic-Aware Recurrent Neural Architecture*. This article is presents a semantic text mining approach, which converts semantic information related to a given set of documents into a set of novel features used for learning. The proposed Model is Semantics-aware Recurrent deep neural Architecture SRNA enables the system to learn from semantic vectors and the raw text simultaneously. The effectiveness of this approach is tested on three text classification tasks: new topic categorization, sentiment analysis, gender profiling.

3. Proposed approach

This section introduces the proposed semantic enriched deep learning architecture (SE-DLA) approach. An efficient architecture for the semantics enrichment of a deep learning model addressing the document classification task. It combines the use of semantic resources such as WordNet [4] and the standard word representation of the documents corpus. This solution uses knowledge present in the semantic resources in order to generate efficiently semantic vectors. Those vectors are used along with the vector space representation of the corpus in a custom hybrid architecture.

*3.1. **Architecture***

The proposed SE-DLA is based on 2 steps of classification. The main idea of this approach is to pre-build two vector representations of the corpus.

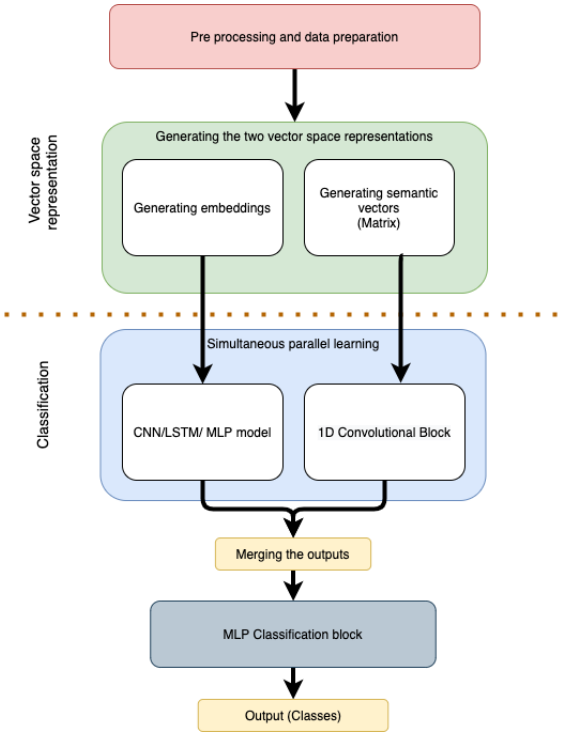

Figure 3. Semantic enriched deep neural network architecture

The first step is focused on standard word embedding. It consists of using the corpus and the hypernyms extracted from the semantic resource in building. In first place, the vector representation of the corpus **D**, and on a second-place constructing a semantic separate matrix **S**.
The second step is based on a semantic resource. It is using those representations (**S**, **D**) as inputs to a hybrid deep neural network architecture in order to perform the document classification.

Figure 3 illustrates the global architecture of the SE-DLA where we can observe the sequential steps including the data cleaning and pre-processing, generating the vector space representations, and finally the classification part using deep neural networks.

*3.2. **Vector space representation***

Figure 4 illustrates the first step of the SE-DLA. It encompasses the pre-processing and creating the vector space representation. It includes a pre-processing step of the text present in the documents, this incorporates handling the following cases:

- *Stop words*: The words that frequently appear on a corpus. Usually, those words give no additional information to the document. Pronouns conjunctions and other terms are considered as stop words.
- *Capitalization*: This step consists of converting the text in a uni-case format. Noise removal: Most of the text and document data sets contain many unnecessary characters such as punctuation and special characters that can be removed.
- *Spelling mistakes*: This is an optional task, indeed, if we dispose of reliable data we don't need to proceed to corrections.

The pre-processing task allows to considerably reduce the vocabulary size. This is extremely important for increasing the quality of the classification.

Once the text is ready, we proceed to the generation of the features vectors. First, documents are encoded (converted to vectorial format). This is highly dependent on the deep learning technique used in the classification step. While using LSTMs [3] and CNNs [28], Word2vec [8] is applied as an embedding of the document. However, while using deep MLPs, TF-IDF [29] is chosen over Word2Vec.

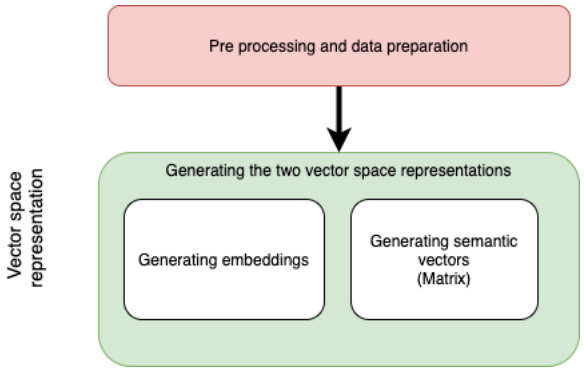

Figure 4. Vector space representation

Secondly, we use Wordnet [4] for the semantic feature vector's generation in order to provide semantics knowledge incorporation.

WordNet [4] is considered as a large and commonly used semantic resource. In this resource, words are annotated with meanings and connected with semantic relations that can be hypernymy. (e.g, $bike \rightarrow vehicle$), hyponymy (e.g, $vehicle \rightarrow bike$) and synonymy (e.g, $bicycle \leftrightarrow bike$)

In our work, we extracted a taxonomy from the wordnet [4] resource considering only the hypernymy relationship. In this case, we obtained a hierarchical structure. In order to explore and represent the knowledge present in this taxonomy, and transcribe it to a suitable representation for the deep learning models. We proposed a vectorization algorithm that exploits word embeddings and cluster summarization techniques.

The proposed approach is based on three main steps:

- Extracting a set of pertinent terms from each document that are named **W**
- Retrieving a set of synonyms and related words **C** for each term in W. This set of terms is considered as a cluster of words sharing the same context information.
- Each term in **C** is converted to a vector form and the centroid of **C** is calculated generating a summarized vector of the entire context.

These steps are introduced in detail in the following sections.

3.2.1. ***Pertinent terms selection***

The first step is to create a set **W** of the pertinent words in a document. In order to create that set, TF-IDF [29] transformation is used for its characteristics of representing a document with the terms that aren't highly present in the entire corpus. Each row in the matrix resulting from the TF-IDF transformation represents a document. In order to select the most pertinent terms in that document, a threshold th was selected empirically. (*Terms with a TF-IDF value higher than th are selected*). The set **W** is created and provided to the following steps.

3.2.2. ***Synsets retrieval***

For a given document we extract the most pertinent terms as shown in the previous step then for each term of that list the following process is applied. A set **H** of the paths to each hypernym of that word is created from the extracted taxonomy (from WordNet). The intersection of **H** is processed creating a final set **h** of the context-related words including the initial word.

3.2.3. ***Word clusters and word vectorization***

As shown in the previous step, for each pertinent term a list of context-related words is created. These word lists are considered as having the same base context. The key idea in this step is to summarize those words obtaining a vector representing the entire cluster. To do so, we use GloVE [10] to create the vectorial representation for each word in the cluster. Then we consider the initial term we created the cluster from as a reference. For each term, we compute its similarity with the reference term using the cosine similarity. Once the vectors are generated and similarity is computed, the centroids of the cluster are generated based on the vectors and similarities using the following equation (1).

$$C_{(i)} = \frac{1}{N_i} \times \sum_{j=1}^{N_i} Sim(V(T_i), V(h_j)) \times V(h_j) \tag{1}$$

The similarity, in this case, is used as a ponderation weight as we give more importance to the terms highly similar to the reference term. The algorithm for this approach is presented here below.

Algorithm 1 Semantic vector representation generation

Require: corpus $\mathscr{D}$, $\mathscr{W}ordNet$ taxonomy

Initilize $V(s) = 0$, for all $s \in \mathscr{S}^{+}$
for each $Doc \in \mathscr{D}$ **do**
  for each $W_i \in \mathscr{P}ertinant(Doc)$ **do**
    $H_i \leftarrow hypernyms(W_i)$
    $S_i \leftarrow similarities(H_i)$
    $C_i \leftarrow Centroids(H_i \times S_i)$
  end for
  $C \leftarrow Matrix(C_i)$
end for

The output of the algorithm, and for each document is a dense matrix, where each line corresponds to the enriched representation of the selected word .

*3.3. **Classification***

Figure 5 illustrates the classification process using deep learning techniques. This process is divided into two parallel models learning simultaneously without interacting with each other. These two models use the two vector space representations generated in the previous step as inputs.

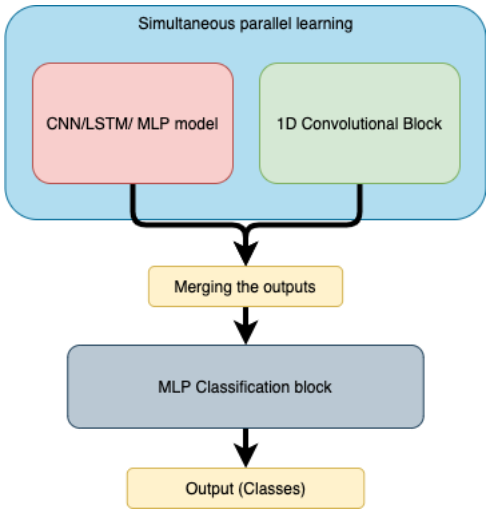

Figure 5. Classification using deep learning

3.3.1. ***Learning from the corpus***

The first model uses the vectors resulting from encoding the documents. In this model, we used various techniques of deep learning separately. CNNs and LSTMs were applied with Word2vec [8] embedding as inputs. Deep MLP networks were also explored while using TF-IDF [29] transformation as an embedding instead of Word2vec [8]. All the configurations of this model learned directly from the corpus independently from the semantic resources. This allows the model to learn relationships and patterns present in the text.

3.3.2. ***Learning from the semantic space***

The second model, unlike the first one, learns from the semantic enriched representation generated in the previous step. The deep learning technique used in this step is convolutional neural networks. One dimensional convolution neural network block is used in this part, this technique allows the extraction of patterns present in the semantic matrix. The output of that model is a result of an average pooling reducing the output size to a one-dimensional vector. This model provides another level of generalizations for the SE-DLA.

3.3.3. ***Merging techniques***

The result of the previous models is provided to a final classification model. In order to perform this classification, we apply a merging of the two outputs of the previous models using a concatenation this will provide a unique vector that will be used as an input for the final step. Another technique for the merging is summing the two vectors and applying an activation function on the result. Relu [30] was used as an activation function for this task. This technique requires that the two inputs must have the same size. Once the merge performed, the result is passed to the final classification model. For this step, we used an MLP with the final layer as an output.

3.4. ***Optimisation***

The loss function used in our approach highly depends on the final task, for sentiment analysis and performing binary classification we used Binary Cross-Entropy (BCE) as shown in the following equation (2) For multi-label classification, we proceeded as the following: each neuron on the output layer represents a label and the activation function applied on it is sigmoid. It means that for each label we perform a binary classification returning if the label is suitable for the given document.

$$BCE(t,p) = -(t * \log(p) + (1-t) * \log(1-p)) \tag{2}$$

For the multi-class classification, Multi-Class Cross-Entropy (MCE) Loss was used as shown in the equation (3)

$$L_{\text{cross-entropy}}\left(\hat{\mathbf{y}}, \mathbf{y}\right) = -\sum y_i \log\left(\hat{y}_i\right) \tag{3}$$

4. Experiments

For the evaluation of the effectiveness of our approach, we considered two datasets as benchmarks:

1. **AG-newsgroups**[2] dataset contains news documents. It is an open-source dataset used widely in the literature to evaluate models on the task of text classification.
2. **Reuters**[3] dataset contains also news texts. A document in this dataset, can belong to multiple classes. In this case, we will evaluate the approach for the task of multi-label classification.

The purpose of our evaluation is to understand how our model performs when compared to existing solutions on different datasets. All details of datasets are described in the Table 1.

Table 1. Datasets description

Dataset	Size	Classes
AG-newsgroups	18000	20
Reuters	10788	90

We chose also, for the evaluation of our approach, to using wordnet as an external semantic resource without picking a specific domain-related resource in order to ideally fit with the chosen datasets.

The performance yield by our SE-DLA approach was compared to various baseline classifiers on the benchmarks introduced previously. We considered two non-neural-based approaches:

1. The support vector machine was trained using the following parameters: (i) the kernel used was rbf, and (ii) the C-value was determined with a grid search. The random forest was trained with the number of trees parameter which is determined as the average length of documents
2. We considered also the following deep learning techniques: a CNN model with a 1D convolutional neural network and a classification LSTM model. For these models, no semantic background was introduced and Word2vec [8] was used as an embedding. We used for the evaluation as well other deep-learning-based techniques that were evaluated with the same benchmarks.

For the AG-newsgroups dataset, we compared our results to Sequence-to-convolution Neural Networks (seq2CNN) [25] and Character-level Convolutional Networks for Text Classification (Char-level CNN) [1]. For the Reuters dataset, we compared our results to (SCDV-MS). [26].

[2] http://qwone.com/ jason/20Newsgroups/

[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/

4.1. ***Results***

The approach was evaluated with an F-1 score measure and the error rate. The F-1 score is measured as the following.

$$F_1 = \frac{2TP}{2TP+FP+FN} \tag{4}$$

$$Err = 1 - \frac{2TP}{2TP+FP+FN} \tag{5}$$

Table 2 shows that for the AG newsgroup dataset, the model yields very efficient results up to 91.2 % of F1 score outperforming the results yield by the baseline approaches and additionally providing better results than Char-level CNN and Seq2CNN solutions.

Table 2. Results on the AG newsgroups dataset.

Model	**F1 Score %**	**Error %**
RF	56	44
SVM	73	27
LSTM	89	11
CNN	90	10
Char-level CNN	90.49	9.51
Seq2CNN	90.36	9.64
SE-DLA (ours)	**91.2**	**8.7**

Table 3 compares results of the considered models on the Reuters dataset. Our approach returns a high score of 85% outperforming the baseline machine learning approaches including the deep learning models (CNN and LSTM) and also the SCDV-MS.

Table 3. Results on the Reuters dataset.

Model	**F1 Score %**	**Error %**
RF	63	37
SVM	77	27
LSTM	81	19
CNN	79	21
SCDV-MS	82.71	17.29
SE-DLA (ours)	**85**	**15**

The results of the experiments show that the SE-DLA, and with the additional external semantic knowledge, provides an additional semantic abstraction for the document's representation. This enhances the results of the various classification tasks. The additional semantic knowledge allows the SE-DLA, unlike other approaches, to avoid quick overfitting on the learning phase by providing a better generalization ability.

5. Conclusion

In this paper, we present a novel and efficient semantic enriched approach for document classification. The proposed approach is a hybrid two-input deep architecture. This approach uses a new algorithm for semantic representation providing an additional richer representation from external semantic resources enhancing performance in text classification on Reuters and AG newsgroup benchmarks. Our approach, and with additional external background knowledge, allows the model to better generalize the learning phase avoiding quick overfitting by adding another level of abstraction and two different representations of the same document. For future work, our approach can be adapted using other deep learning techniques to a different NLP task. The use of domain ontologies, for instance, including the use of an energy ontology instead of terminologies such as wordnet, can provide a more accurate context, and it is a pertinent path of improvement of the proposed approach.

References

[1] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[2] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Christiane Fellbaum. Wordnet. *The encyclopedia of applied linguistics*, 2012.

[5] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[6] Thomas K Landauer. *Latent Semantic Analysis*. American Cancer Society, 2006.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[8] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[10] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[11] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[12] Luca Cagliero and Paolo Garza. Improving classification models with taxonomy information. *Data & Knowledge Engineering*, 86:85–101, 2013.

[13] Anže Vavpetič and Nada Lavrač. Semantic subgroup discovery systems and workflows in the sdm-toolkit. *The Computer Journal*, 56(3):304–320, 2013.

[14] Blaž Škrlj, Jan Kralj, and Nada Lavrač. Cbssd: community-based semantic subgroup discovery. *Journal of Intelligent Information Systems*, 53(2):265–304, 2019.

[15] Mohamed K Elhadad, Khaled M Badran, and Gouda I Salama. A novel approach for ontology-based feature vector generation for web text document classification. *International Journal of Software Innovation (IJSI)*, 6(1):1–10, 2018.

[16] Rajinder Kaur and Mukesh Kumar. Domain ontology graph approach using markov clustering algorithm for text classification. In *International Conference on Intelligent Computing and Applications*, pages 515–531. Springer, 2018.

[17] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, 2014.

[18] Petar Ristoski, Stefano Faralli, Simone Paolo Ponzetto, and Heiko Paulheim. Large-scale taxonomy induction using entity and word embeddings. In *Proceedings of the International Conference on Web Intelligence*, pages 81–87, 2017.

[19] Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1501–1511, 2015.

[20] Jiang Bian, Bin Gao, and Tie-Yan Liu. Knowledge-powered deep learning for word embedding. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 132–148. Springer, 2014.

[21] Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng. Some effective techniques for naive bayes text classification. *IEEE transactions on knowledge and data engineering*, 18(11):1457–1466, 2006.

[22] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016.

[23] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019.

[24] Jennifer J Gago, Valentina Vasco, Bartek Łukawski, Ugo Pattacini, Vadim Tikhanoff, Juan G Victores, and Carlos Balaguer. Sequence-to-sequence natural language to humanoid robot sign language. *arXiv preprint arXiv:1907.04198*, 2019.

[25] Taehoon Kim and Jihoon Yang. Abstractive text classification using sequence-to-convolution neural networks. *arXiv preprint arXiv:1805.07745*, 2018.

[26] Vivek Gupta, Ankit Saw, Pegah Nokhiz, Harshit Gupta, and Partha Talukdar. Improving document classification with multi-sense embeddings. *arXiv preprint arXiv:1911.07918*, 2019.

[27] Blaž Škrlj, Jan Kralj, Nada Lavrač, and Senja Pollak. Towards robust text classification with semantics-aware recurrent neural architecture. *Machine Learning and Knowledge Extraction*, 1(2):575–589, 2019.

[28] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[29] Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*, pages 986–987. Springer US, Boston, MA, 2010.

[30] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[31] Ho Tin Kam. Random decision forest. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, volume 1416, page 278282. Montreal, Canada, August, 1995.

[32] BE Boser, IM Guyon, and VN Vapnik. A training algorithm for optimal margin classifiers. proceedings of the fifth annual workshop on computational learning theory; pittsburgh, pennsylvania, usa. 130401: Acm. 1992.

[33] V Vapnik and A Ya Chervonenkis. A class of algorithms for pattern recognition learning. *Avtomat. i Telemekh*, 25(6):937–945, 1964.