HWE: Hybrid Word Embeddings For Text Classification

Xuebo Song
School of Computing
Clemson, SC, United States
xuebos@clemson.edu

Pradip K Srimani
School of Computing
Clemson, SC, United States
srimani@clemson.edu

James Z. Wang
School of Computing
Clemson, SC, United States
jzwang@clemson.edu

ABSTRACT

Text classification is one of the most important tasks in natural language processing and information retrieval due to the increasing availability of documents in digital form and the ensuing need to access them in flexible ways. By assigning documents to labeled classes, text classification can reduce the search space and expedite the process of retrieving relevant documents. In this paper, we propose a novel text representation method, *Hybrid Word Embeddings (HWE)*, which combines semantic information obtained from WordNet and contextual information extracted from text documents to provide concise and accurate representations of text documents. The proposed HWE method can improve the efficiency of deriving word semantics from text by taking advantage of the semantic relationships extracted from WordNet with less training corpus. Experimental study on classification of documents shows that the proposed HWE outperforms existing methods, including Doc2Vec and Word2Vec, in terms of classification accuracy, recall, precision, etc.

CCS CONCEPTS

• **Information systems** → **Information retrieval**;

KEYWORDS

Word embedding; WordNet; Text classification

1 INTRODUCTION

Text classification is the activity of labeling natural language text with thematic categories from a pre-defined set. Text classification techniques have been applied across natural language processing (NLP) applications such as spam filtering, sentiment analysis, information retrieval, etc.

One of the most important subtasks of performing text classification is to get the representation of documents in order to obtain semantic similarities / distances among documents. Document representation techniques including VSM [1], Latent Semantic Indexing (LSI) [2] and neural network based language models [3–7] have been applied in various aspects of NLP and information retrieval tasks. However, none of them can effectively capture the long-distance semantic relations (semantic relations between words that are far apart from

each other within the text) while maintaining low data dimension with relatively small training set to achieve good performance.

Due to the limited semantic relationships that can be contained in a single text document, especially short text document, it is necessary to incorporate knowledge from other sources to effectively analyze text content as knowledge resources such as ontologies can provide extra information that could lead to better solutions. Ontologies such as WordNet [8] represent real-world entities along with properties and relations as large semantic graphs.

To take advantage of the knowledge in ontologies, we propose a novel text representation model, Hybrid Word Embeddings (HWE), that combines semantic information obtained from WordNet and contextual information extracted from text documents to represent text documents into vectors. The proposed HWE approach can provide concise and accurate representations of words and text documents. Experimental study shows that HWE outperforms existing methods, including Doc2Vec and Word2Vec, in terms of classification accuracy, recall, precision, etc.

The remaining part of the paper is organized as follows. Section 2 introduces the method to learn semantic relationships from WordNet. Section 3 proposes the HWE model and discusses how to generate document vectors based on the proposed HWE model. The experimental study and discussion are given in section 4 and 5 respectively.

2 LEARNING SEMANTIC RELATIONSHIPS FROM WORDNET

Domain knowledge consists of information about the data that is already available either through discovery process or from a domain expert. Knowledge bases such as Wikipedia, WordNet, Gene Ontology, provide fruitful semantic information about data, which might guide the discovery of the true semantics of data. In text mining, it is very important to capture domain knowledge as it can make texture patterns more visible and constrain the search space as well as the rule space. In a classification or predictive modeling task, domain knowledge could help improve learning efficiency as well as the quality of the learned model. It could also be used to initialize a knowledge structure and make the discovered knowledge more interpretable.

Semantic relationships demonstrate their importance in many NLP tasks [9–11]. Although WordNet provides a practical solution to the synonymy problem by grouping synonyms, how to determine the exact meaning of a word is not addressed. Instead, it only lists possible meanings of a word and keep the disambiguation problem open to the applications. A naive strategy was proposed in [11] to address the polysemy problem using WordNet. It chooses the concept (synsetID) with the highest frequency in the text corpus as the word concept. Unfortunately, other meanings of the word may

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

NLPIR 2019, June 28–30, 2019, Tokushima, Japan

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6279-5/19/06...\$15.00

<https://doi.org/10.1145/3342827.3342837>

Figure 1: The architecture of the hybrid word embedding (HWE) system

never be selected due to this high bias on more frequent words in the thesaurus even if the document intends to convey the other meanings. Nowadays, words are endowed with new semantic meanings rapidly in the light of large volumes of news feed, tweets, blogs, etc. This rapid semantic change or enrichment can not be well captured by a single thesaurus or ontology due to the extensive labor involved in the manually curating process. Hence, we propose a hybrid word embedding model that not only can extract knowledge from WordNet but also can capture the semantics within the text. The architecture of the proposed hybrid word embedding (HWE) system is shown in the upper part of Figure 1. We first extract concepts from the document by taking advantage of the knowledge in WordNet. These initial representations of the concepts are called “*Concept trees*” (Definition 1), which are constructed by linking the concepts in a document with relationships found in WordNet. “Concept trees” will then be encoded into “tree vectors”. The structure of the hybrid word embedding model can be found in the lower part of Figure 1. For each word in the document, the corresponding concept trees are fetched from the look-up table. The tree vectors are calculated according to Definition 6.

Definition 1. *Concept tree*. A concept tree consists of concepts within a text document that are connected with relationships obtained from ontologies.

2.1 Concept tree construction

The construction of concept trees consists of 3 steps: stop word removal, stemming and linking.

Stop word removal: Stop words, words that do not contribute much to the semantic meaning such as “the”, “a”, “is”, “which”, are removed from the text.

Stemming: Words with the same root appear in various morphological forms. All the words are stemmed using the morphology function provided with WordNet. For instance, “takes”, “took” and “taking” are all mapped to the root word “take”. Since we do not consider the word ordering, the word “taken” can also be mapped to “take” ignoring the subject. After stemming, each word obtained corresponds to a synset ID, which constitutes a set of synonyms.

Linking: Terms obtained from the same text document after pre-processing are connected via a hypernymy relationship between their associated synset IDs, if there is any. A hypernym of a term is a more general term and a hyponym is a more specific term. For example, an apple is a hyponym of edible fruit and an edible fruit is a hypernym of an apple. This hypernym relationship from WordNet is exploited to build the concept trees. For example, an article studying whether eating an apple a day can prevent from cancer might contain words like “food”, “fruit”, “apple”, “tumor”, “cancer”. We thus can construct concept trees for terms “food”, “fruit”, “apple” and that for “tumor” and “cancer” separately as shown in Figure 2. A word in different contexts may represent different concepts, which are represented as multiple synsets. For instance, the word “orange” could be interpreted as a color as in “The sweater is orange.”. It could also be regarded as an edible fruit as in “The orange is so delicious!”. Thus, a word may be placed in different synsets, which form different concept trees. It is hard to determine the correct concept for an ambiguous word that belongs to several synsets and so is deciding the semantics of a document that contains several ambiguous terms. Wang [12] directly uses the first synset of a word which has the highest frequency of occurrence in WordNet as the word concept. Using this approach, all ambiguous words are represented by the same concept. For instance, the first synset of “bank” in WordNet represents the sloping land (especially the slope beside a body of water). When it comes to a finance news, the synset of “bank” denotes a financial institution that accepts deposits and channels the money into lending activities. To avoid mismatching concept, we will keep all the synsets and build individual concept trees for different concepts of ambiguous words (words that have multiple synsets in WordNet). We will then use the text context to assist the disambiguation during the HWE process.

Therefore, all the synsets of a word are used in concept tree construction. For each synset, the associated hypernym trees are obtained through the hypernym relationship. The hypernym trees contain all the ancestors of the synset. Each synset contains multiple words that have the same meaning, thus a hash table is built with words as indices and the list of synsets (word senses) associated with each word in the hypernym trees are treated as values. As there could be overlaps of the synsets among different words, for example, “band” may have the same hypernym “jewelry” as the word “necklace”. If both “necklace” and “band” appear in the document, they are linked together in the concept tree. This can be done in approximately linear time with respect to the total number

Figure 2: Concept trees derived for terms “food”, “fruit”, “apple”, “tumor”, “cancer” from WordNet

of synsets of words that appear in the document in the worse scenario by using the Union-Find algorithm [13] with union-by-rank and path compression. One thing worth noting here is that we are actually linking synsets in the concept tree. The synsets will be then converted to the corresponding words after the concept tree is constructed. To achieve this, inverted indices are built to map synset to corresponding words that appear in the document. An advantage of this implementation is to disambiguate word senses in case of words that appear in the document have different concept trees with different synsets (word senses). For instance, in the above example, if “musical group” also appears in the document, the senses of “band” representing “a group of instrumentalists” can also be used to construct an additional concept tree for this word sense.

2.2 Concept tree encoding

We incorporate domain knowledge from WordNet as well as word statistics into concept trees. If a word / concept appears multiple times in a document. Its importance should be accumulated along with its frequencies. Based on this observation, we first define several terms that will be used in the encoding phase.

Definition 2. Semantic content value. The sum of frequencies of words mapped to a synset ID i is the semantic content value of synset ID i , denoted as ω_i . The sum of the semantic content values of all the synset ID associated with a concept k is the semantic content value of the concept k , denoted as SCV_k .

$$SCV_k = \sum_{i \in T_k} \omega_i \quad (1)$$

where T_k is the set of words constituting concept k .

Definition 3. Semantic content rate (SCR). The semantic content value of a concept k normalized by the total sum of the semantic content value of all the concept in a concept tree is the SCR of the concept k , which is defined as,

$$SCR_k = SCV_k / \sum_{i=1}^n SCV_i \quad (2)$$

However, words at upper levels of WordNet have more general concepts than words at lower levels and the semantic similarities

between words at upper levels should be less than those between words at lower levels. This implies that more general terms convey less information than more specific terms. For instance, an article discussing “the impact of eating apple on avoiding cancer” may probably mention the word “fruit”, which would indicate the ‘fruit-apple’ hypernymy relation constructed in a concept tree. High frequencies of “fruit” would mislead the topic to represent the concept “fruit” instead of “apple”. Key points of the text might not be captured as the word “fruit” also applies to other fruits like “pineapple”. This could affect the classification precision or the recall. For example, in information retrieval scenario where the search query contains “apple”, the above article that has both “fruit” and “apple” might have a rather low rank even though the article is talking about the “apple” instead of the more general concept “fruit”. Hence, the impact of words at upper levels should be dampened while lower level words should have higher impact since they convey more specific information than the upper level ones.

In [14], statistical approaches were used to address the semantic impacts of different ontological levels. However, this is impractical for short text documents, as words are not likely to appear many times as they would in longer documents, which would result in a sparse term frequency matrix that can hardly capture the underlying structure of the document.

We use the scaling depth effect to model the impact of different levels on the WordNet. This effect $\theta(h)$ of level h is captured by,

$$\theta(h) = \frac{1}{1 + e^{-\gamma h}} \quad (3)$$

We note that $\theta(h)$ is a monotonically increasing function with respect to depth h , while γ acts as a scaling factor that can be used to control the extent of the depth effect. As a result, it would scale up the impact of word frequencies at lower layers while scaling down those at upper layers. By adjusting the contribution of depth of each synset ID, we have the following definitions,

Definition 4. Weighted semantic content value. The sum of the semantic content values of all the synset ID associated with a concept k weighted by a scaling depth effect $\theta(h)$ is the weighted semantic content value of the concept, denoted as $WSCV_k$,

$$WSCV_k = \sum_{h=1} \theta(h) SCV_k \quad (4)$$

Definition 5. Weighted Semantic content rate (WSCR). The weighted semantic content value of a concept k normalized by the total weighted sum of the semantic content value of all the concepts within a concept tree is the WSCR of the concept k defined as,

$$WSCR_k = WSCV_k / \sum_{i=1}^n WSCV_i \quad (5)$$

We use V dimension vectors to encode concept trees, where V is the vocabulary size. The vocabulary is typically defined using the V most common words in the text corpus. To encode concept tree, we define tree vectors as follows,

Definition 6. Tree vector. Vector with weighted semantic content rate at each dimension representing concepts in a concept tree.

Taking the “tumor-cancer” shown in Figure 2 as an example, suppose $ct_{tumor} = 10, ct_{cancer} = 5$, γ in Equation 3 is 0.5. The tree vector of “tumor-cancer” will have $\frac{10/(1+e^{-0.5*0})}{10/(1+e^{-0.5*0})+5/(1+e^{-0.5*1})} = 0.62, \frac{5/(1+e^{-0.5*1})}{10/(1+e^{-0.5*0})+5/(1+e^{-0.5*1})} = 0.38$ at the dimension of “tumor” and “cancer”, respectively, and 0 elsewhere. If a word stands alone in the concept tree (the concept tree only contains one word), the corresponding tree vector would have 1 at the position of the word, and 0 elsewhere. The tree vectors generated are the target values representing the context. The procedure is shown in the next section.

3 HYBRID WORD EMBEDDINGS (HWE)

Distributed word representations (word embeddings) were explored in the NLP community [4, 6, 7] where words are represented as dense, low-dimensional vectors. Each dimension is a measure of the association between the word and a particular context. To incorporate WordNet knowledge to enrich semantics of text content, we propose a neural network model to generate word embeddings that convey both contextual information and the knowledge extracted from WordNet.

Our proposed hybrid word embeddings model is adapted from the continuous bag-of-words model (CBOW) [6] by re-engineering the outputs and the objective function.

Unlike the CBOW model which uses the target word as output and softmax to obtain the posterior distribution of words, we use tree vectors derived in Section 2 as the output. For each word, its corresponding concept tree vector is directly regarded as the output. Instead of predicting a single word, using tree vectors as the output can automatically connect related concepts. This is because tree vectors capture more semantics and requires smaller training corpus with the help of information residing in existing knowledge bases. For example, the two sentences “I love eating apple”, “I love eating fruits” are both valid and related and the word “apple” and the word “fruits” are related. However, a traditional CBOW model treats “apple” and “fruits” as totally unrelated, even though “apple” is a kind of “fruit” in real world, described as a hypernymy relationship in WordNet. As they share a similar context, with the help of our encoded tree vectors, training the word “apple” in the sentence “I love eating apple” will also train the word “fruits” in “I love eating fruits” as “fruits” and “apple” are connected in the encoded tree vector.

The objective function is the **Multilabel Soft Margin Loss**,

$$loss(x, y) = - \sum_{i=1}^V \frac{y[i] \log \frac{e^{x[i]}}{1+e^{x[i]}} + (1-y[i]) \log \frac{1}{1+e^{x[i]}}}{V} \quad (6)$$

where $y[i]$ is the i -th element in the tree vector and $x[i]$ is the i -th element of the output. The training objective (for one training sample) is to minimize the loss in Equation 6. The weights are updated by stochastic gradient descent and backpropagation. According to stochastic gradient descent, the corresponding weights are updated with a proportion to the *loss* in Equation 6. If the predicted output have a large difference (*loss*) from the target tree vector, the corresponding weights will be changed significantly. Conversely, if the output vector is close to the target tree vector (small loss),

meaning word (concept) may be fairly accurately predicted, it will have little effect on the corresponding weights.

The hybrid word embeddings obtained from the proposed HWE model can be regarded as new representations of words. They are vectors of a fixed dimension. Here, we further extend our word embeddings to document representations. A document is a collection of words, where similar words convey semantics at similar dimension in the word embeddings while the dimensions expressed by dissimilar words may have less overlaps, if at all. Based on this observation, we adopt a simple **Normalized sum** strategy to generate document vectors.

4 PERFORMANCE STUDY

To evaluate the effectiveness of the learned word embeddings of the proposed HWE, we focus on conducting text classification using the 20 newsgroups data set [15], in comparison with existing approaches, including VSM [1], Word2Vec [6], Google-News pre-trained Word2Vec, and Doc2Vec (PV-DM, PV-DBOW) [4].

The data set is a collection of 19,974 newsgroup documents, partitioned evenly across 20 different newsgroups. Some of the news groups are very closely related to each other (e.g., comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g., rec.autos / talk.politics.guns). To train the proposed HWE model as well as Word2Vec and Doc2Vec, we randomly select 70% of the documents for training and use the remaining 30% for testing. k -Nearest Neighbors algorithm (k -NN) is used to classify an unseen document to k documents that are most similar to it. A query document is assigned the documents class which has the most representatives within the k most similar documents. We use the proposed **Normalized sum** to construct document vectors for word embeddings from the proposed HWE model, Word2Vec and the pre-trained Word2Vec. Document vectors for VSM and Doc2Vec are directly obtained from the corresponding model. The similarity between two documents is obtained by taking the cosine similarity between the corresponding two document vectors.

We evaluate the performance using the following metrics, macro-average classification **Recall**, **Precision**, **Accuracy**, **F1-Score**. The macro-average precision, recall, accuracy are taken as the average of the precision, recall, accuracy on different classes. The macro-average F1-Score is the harmonic mean of the macro-average precision and recall.

Table 1 shows experimental results of all models we examined on the 20 newsgroups data set. The proposed HWE model outperforms all the other models. The major problem of Doc2Vec (PV-DM, PV-DBOW) is that the document vectors need to be trained by the entire corpus at the cost of time efficiency and computation resource. The extensive labor and procedure to train documents to get document embeddings inevitably limit the application of Doc2Vec. Due to the large amount of documents, it is hard to train Doc2Vec models in real time to obtain document embeddings. The rapid generation of new online documents only aggravate the situation. New blogs or tweets cannot be identified, classified or even analyzed in real time using the Doc2Vec model. The proposed HWE model can construct document representations in linear time in proportion to the lengths of documents. As the semantic relationships between concepts are already extracted in the training process of generating

Table 1: Performance of text classification on 20 newsgroups

Model	Accuracy	Recall	Precision	F1-Score
HWE	0.761	0.833	0.857	0.845
PV – DM	0.744	0.818	0.844	0.831
PV – DBOW	0.722	0.793	0.852	0.821
Word2Vec	0.621	0.667	0.833	0.740
Word2Vec _{pre}	0.667	0.714	0.862	0.781
LSI	0.648	0.75	0.778	0.764
BOW	0.535	0.588	0.769	0.667

hybrid word embeddings, when constructing document vectors, we only need to scan the document once to apply the *Normalized sum* to the words composing the documents.

5 DISCUSSION

In this paper, we proposed the Hybrid Word Embeddings (HWE) model which combines semantic information obtained from WordNet and contextual information extracted from text documents to provide concise and accurate representations of text documents. Experimental study on classification of documents shows that HWE outperforms existing methods, Vector Space Model (VSM) model [1], Doc2Vec [4], Latent Semantic Indexing (LSI) [2], Word2Vec [6], in terms of classification accuracy, recall, precision, etc. Unlike existing document representations which need very large corpus as input to create sparse representations and project them into a lower dimensional dense vector space, including Doc2Vec [4] and Word2Vec [3]. The proposed HWE model can use much less data with the help of existing knowledge resources like WordNet. Nowadays, words are endowed with new semantic meanings rapidly in the light of large volumes of news feed, tweets, blogs, etc. This rapid semantic change or enrichment can not be well captured by a single thesaurus or ontology. The proposed HWE can alleviate this problem by taking advantage of both contextual information as well as the pre-established structured knowledge in ontologies. The benefits from enriching text content using ontologies such as WordNet is two-fold. First, it resolves synonyms. Words or phrases referring to the same concept are grouped together. Similar words like “fabulous”, “fantastic” will be grouped as one automatically. Second, it alleviates disambiguation. Concepts along with their relations captured in existing knowledge bases and context semantics found in surrounding text can complete each other to surface the true meanings of concepts. With the combination of knowledge bases and word embeddings model, the proposed system can provide insights to optimize NLP systems. The semantics of these concepts can be derived directly without knowledge of the contexts. These commonly already known concepts (context-free concepts) could be excluded in regular NLP systems. In lieu of training context-free concepts, we encoded these concepts with the aid of domain knowledge which could be considered the preprocessing of our proposed text representations with significant reduction in computational costs involved in the training process. Although the focus of this work is to represent documents, with proper knowledge resources, such as abbreviation dictionaries, this work can also be applied to a wider range of scenarios, short text analysis, information retrieval, etc.

6 ACKNOWLEDGEMENT

This work was partially supported by NSF grant 1759856, NIH grant 2R01HD069374-0681, and Guangzhou major industrial research grant #201802010010.

7 REFERENCES

- [1] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.
- [2] TK LANDAUER and ST DUMAIS. A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211–240, 1997.
- [3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
- [4] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196. JMLR Workshop and Conference Proceedings, 2014.
- [5] Bo Li, James Z. Wang, Frank Alex Feltus, Jizhong Zhou, and Feng Luo. Effectively integrating information content and structural relationship to improve the go-based similarity measure between proteins. *CoRR*, abs/1001.0958, 2010.
- [6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
- [7] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, 2012.
- [8] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.
- [9] Kemafor Anyanwu, Angela Maduko, and Amit Sheth. Semrank: Ranking complex relationship search results on the semantic web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW ’05, pages 117–127, New York, NY, USA, 2005. ACM.
- [10] Xuebo Song, Lin Li, Pradip K. Srimani, Philip S. Yu, and James Z. Wang. Measure the semantic similarity of go terms using aggregate information content. In Zhipeng Cai, Oliver Eulenstein, Daniel Janies, and Daniel Schwartz, editors, *Bioinformatics Research and Applications*, volume 7875 of *Lecture Notes in Computer Science*, pages 224–236. Springer Berlin Heidelberg, 2013.
- [11] Andreas Hotho, Steffen Staab, and Gerd Stumme. Wordnet improves text document clustering. In *In Proc. of the SIGIR 2003 Semantic Web Workshop*, pages 541–544, 2003.
- [12] James Z. Wang and William Taylor. Concept forest: A new ontology-assisted text document similarity measurement method. In *Web Intelligence, IEEE/WIC/ACM International Conference on*, pages 395–401, Nov 2007.
- [13] Robert E. Tarjan and Jan van Leeuwen. Worst-case analysis of set union algorithms. *J. ACM*, 31(2):245–281, March 1984.
- [14] Xuebo Song, Lin Li, Pradip K. Srimani, Philip S. Yu, and James Z. Wang. Measure the semantic similarity of go terms using aggregate information content. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 11(3):468–476, May 2014.
- [15] 20-newsgroup collection. 1999.