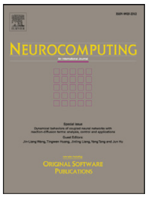
Neurocomputing 326–327 (2019) 141–150

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



WordNet2Vec: Corpora agnostic word vectorization method

Roman Bartusiak*, Łukasz Augustyniak, Tomasz Kajdanowicz, Przemysław Kazienko, Maciej Piasecki

Department of Computational Intelligence, Wrocław University of Science and Technology, Wrocław, Poland

ARTICLE INFO

Article history:
Received 2 June 2016
Revised 5 December 2016
Accepted 29 January 2017
Available online 29 September 2017

Keywords:
Natural language structuring
WordNet
WordNet2Vec
Vectorization
Network transformation
Sentiment analysis
Transfer learning
Big data
Complex networks

ABSTRACT

The complex nature of big data resources requires new structuring methods, especially for textual content. WordNet is a good knowledge source for the comprehensive abstraction of natural language as it offers good implementation for many languages. Since WordNet embeds natural language in the form of a complex network, a transformation mechanism, WordNet2Vec, is proposed in this paper. This creates vectors for each word from WordNet. These vectors encapsulate a general position — the role of a given word related to all other words in the given natural language. Any list or set of such vectors contains knowledge about the context of its components within the whole language. This type of word representation can be easily applied to many analytic tasks such as classification or clustering. The usefulness of the WordNet2Vec method is demonstrated in sentiment analysis including the classification of an Amazon opinion text dataset with transfer learning.

1. Introduction

With fast technological growth, the ability to solve complex problems has increased. More and more data continuously generated by social media and various IT systems require more complex, accurate and efficient methods and algorithms so as to provide valuable insight. The tools of Big Data Analytics and High Performance Computing accessible to all enable us to face these challenges but simultaneously raise a wide variety of new questions and problems which can only be addressed by scientists.

There are many aspects that must be taken into account while processing large amounts of data. One of them concerns the general problem of knowledge representation for complex structures. Textual, multimedia or networked content has an unstructured nature that is unsuited to the application of most known analytic methods. Recently, most of the data sets have come from texts derived from social media, and they are directly impacted by the complexity of natural languages. Transforming big data of such a kind into big knowledge still remains a great challenge. An obvious and commonly performed step in such a transformation is structuring. However, how should we transform the complex nature of natural language into a structured form?

Overall, there are two main knowledge sources for the lexical semantics of a natural language that have practical importance for automated processing: (1) text corpora and (2) wordnets. A wordnet is a lexico-semantic network whose construction originates from the ideas first exemplified in the Princeton WordNet [1], a large language resource describing lexical meanings of English by means of the various lexico-semantic relations. A wordnet provides a partial description of the lexical meanings – we know only constraints expressed by the instances of the lexico-semantic relations – but many wordnets are big enough to provide coverage of practical importance. Many wordnets were built manually with every piece of the structure verified by linguists. That is the case of WordNet and also plWordNet [2] – a large wordnet of Polish, the largest world wordnet. In a wordnet, lexical meanings are enumerated and grouped into sets called *synsets*, i.e. sets of synonyms. A synset can be interpreted as representing a concept (called also *a lexicalised concept*) present in the interpretation of all its members. Synsets are linked by semantic relations derived from lexical semantics such as hypernymy or meronymy. Direct relations between lexical meanings, e.g. antonymy or derivational relations, are also described in wordnets. The former are called conceptual relations and the latter lexical

* Corresponding author.
E-mail addresses: roman.bartusiak@pwr.edu.pl (R. Bartusiak), lukasz.augustyniak@pwr.edu.pl (Ł. Augustyniak), tomasz.kajdanowicz@pwr.edu.pl (T. Kajdanowicz), kazienko@pwr.edu.pl (P. Kazienko), maciej.piasecki@pwr.edu.pl (M. Piasecki).

https://doi.org/10.1016/j.neucom.2017.01.121
0925-2312/© 2017 Elsevier B.V. All rights reserved.

relations[1]. A large wordnet includes a great deal of infrequent words, many expressing specific meanings, that are often difficult to be found even in large corpora. Such words were added by its editors in order to complete selected semantic subfields. A wordnet also describes many specific meanings, which in corpora may either not exist or be dominated by more popular meanings. WordNet itself possesses a complex network structure, which is inappropriate for commonly used analytic and reasoning methods.

Extraction of lexico-semantic knowledge from corpora heavily depends on the statistical information concerning word co-occurrences, n-gramms or lexico-syntactic dependencies. The extracted knowledge mostly takes the form of word-to-word similarity measures or word semantic clusters. Most extraction techniques require large frequencies of words to be well described (>100) that are difficult to be achieved for more specific and infrequent words even in large corpora. In contrast to corpus-based methods, a wordnet provides descriptions for many rarely used words.

However, wordnets are based on limited sets of relations and do not cover many semantic distinctions that can be discovered in corpus-based similarity measures.

The main problem addressed in this paper is how to find a method for the transformation of the complex network structure of the whole WordNet, as a description of the lexico-semantic system, into a simple structured form, that is to say vectors that are suitable for further processing by means of known methods. In particular, the transformation should encapsulate the position of each word in the WordNet network related to all other words in WordNet to represent the meaning of the term as it is constrained by the WordNet relations.

We propose a new method, called *WordNet2Vec*, for textual data representation in a vector space that is able to satisfy the above-mentioned requirements. Based on the network of words from WordNet, we build a word representation in the vector space using its distance from any other word in the network. In order to present the pair-wise word distance, the method calculates the all-pair shortest paths in WordNet. Thanks to that, any list of vectors – list of words - also reflect the complex nature of the whole language encoded within these vectors.

To demonstrate the usefulness of the proposed vectorization method, we suggest a use case in which any textual document is transformed into a list of vectors from *WordNet2Vec*. Next, this representation is applied to the classification problem, namely sentiment analysis and assignment. Finally, we compare the effectiveness of the baseline method and some recently emerging and very popular approaches such as Doc2Vec [3]. Since we derive knowledge from a general language database – WordNet, our method enables us to build more robust knowledge representation models and to achieve a very high level of efficiency, generalization and stability, especially if applied to transfer learning scenarios.

In the experimental part of our research, the proposed *WordNet2Vec* vectorization method was utilized for sentiment analysis in the Amazon product review dataset. In general, sentiment analysis of texts means assigning a measure of how positive, neutral or negative the text is. Using *WordNet2Vec* representation and a supervised learning approach, the sentiment is assigned to the document according to its content. In other words, sentiment analysis is the process of determining the attitudes, opinions and emotions expressed within a text [4].

The rest of this paper is organized as follows: in Section 2 related work is presented. Then, the new *WordNet2Vec* method and other comparable methods are described in Section 3. The experimental design and results are discussed in Section 6. Finally, the ideas presented are summed up and future work directions are sketched out in Section 9.

2. Related work

Representation of knowledge is an area of artificial intelligence concerned with how knowledge can be represented symbolically and manipulated in an automated way [5]. Textual documents are intrinsically unstructured and in order to process them robustly one needs to employ various methodologies such as resolving, aggregating, integrating and abstracting. The derived knowledge can be represented in various structures including semantic nets, frames, rules, and ontologies [6]. Due to the fact that the majority of machine learning and supervised learning approaches operate in a vector space, the text representation should also be located within a vector space.

Recent achievements in textual data representation are briefly presented further on in Section 2.1 with a special focus on word embedding methods based on language corpora. Then, a wordnet which is a language resource representing the lexico-semantic system is briefly described from the perspective of its application in the vectorization method proposed in the paper. In addition, some insights into complex network All-pairs Shortest Paths (APSP) computation are discussed. Finally, an introduction to sentiment analysis is provided due to its employment in the use case, Section 5.

2.1. Word embedding techniques

Word embedding is the generous name given to a collection of language modeling and feature learning techniques where words from the vocabulary are mapped to vectors of real numbers. Word embedding can be equally called *a word vectorization method*. The mapping is usually done in a low-dimensional space, but depends relatively on the vocabulary size. Historically, word embedding was related to statistical processing of big corpora and introduced to derive latent semantic features from word co-occurrence in the documents.

In general, word embedding techniques can be dived into two main groups. The first is based on a probabilistic prediction approach. Such methods are used to train a model, based on a context window composed of words from a corpus, generalizing the results into a reduced n dimensional space (n is chosen arbitrarily). Then a word is represented as a vector in the space and preserves a context property, so words that are located close to each other in this space frequently co-occur in the corpus. *Word2Vec* (Word-2-Vector) [7,8] is the best known method in this group. It is based on skip-grams and continuous bags of words (CBOW). Given the neighboring words in the window, the CBOW model is used to predict a particular word w. In contrast, given a window size of n words around a word w, the skip-gram model predicts the neighboring words given the current word.

There have been other deep and recurrent neural network architectures proposed for learning word representation in vector space before, i.e. [9,10], but Word2Vec is one of the best and most commonly studied methods.

Count-based models constitute the second group of word embedding techniques. The GloVe algorithm, presented by Pennington et al. [11], is one of them. In count-based models the vectors are based on the word co-occurrence frequency matrix. In order to shrink the size of word vectors dimensionality reduction algorithms are applied. The main intuition for the GloVe model is the simple observation that ratios of word-to-word co-occurrence probabilities can be utilized to represent some aspects of the meaning of the natural language concerned. Word vectors produced by the GloVe method perform very well as a solution

[1] It is worth noticing that there is no such distinction in plWordNet in which a unified model was introduced with all relations of linguistic nature originating directly from language data. A synset groups lexical meanings with the same network characteristics.

for word similarity tasks, and are similar to the Word2Vec approach. Lebret and Collobert [12] and Dhillon et al. [13] proposed another version of count-based models. Lebret presented a method that simplifies word embedding computation through a Hellinger PCA of the word co-occurrence matrix. Dhillon used a new spectral method based on CCA (canonical correlation analysis), Two Step CCA (TSCCA), to learn an eigenword dictionary. This procedure computes two set of CCAs: the first one between the left and right contexts of the given word and the second one between the projections resulting from this CCA and the word itself. Lebret and Collobert [14] proposed an alternative model based on counts. They used the Hellinger distance to extract semantic representations from the word co-occurrence statistics in a large text corpus.

In conclusion, Word2Vec is a predictive model, whereas GloVe is a count-based model [15]. However, there is no qualitative difference between predictive models and count-based models. They use different computational methods that produce a very similar type of semantic model [16,17].

2.2. WordNet

A wordnet is a large lexico-semantic database of natural language. Following the seminal Princeton WordNet for English [1,18], wordnets for many different languages have been built[2]. Nouns, verbs, adjectives and adverbs are grouped into synsets, vaguely defined as sets of synonyms such that they express some lexicalized concept that is common for synset members called *senses* (i.e. word senses) representing particular lexical meanings. Synsets are linked by *conceptual relations* that were inspired by linguistic lexico-semantic relations such as hyper/hyponymy or holo/meronymy. In addition, senses (i.e. synset members, e.g. *man 1*) are linked by *lexical relations*. Most of them are well known lexico-semantic relations including antonymy or morpho-semantic relations.

Various types of links can be found in WordNet:

- There are 285,348 semantic links between synsets:
  - 178,323 both hypernyms and hyponyms,
  - 21,434 similarity,
  - 18,215 both holonyms and meronyms,
  - 67,376 other connections.
- There are 92,508 lexical links between all words:
  - 74,656 derivations,
  - 7981 antonyms,
  - 9871 other connections.

WordNet is freely and publicly available for download on a licence similar to BSD. Its stable structure means that it provides a comprehensive representation of the whole lexico-semantic system of the natural language concerned, which can be applied in several methods of Computational Linguistics and Natural Language Processing.

In our work, we assume that WordNet can be treated as an approximate representation of the whole lexico-semantic system. In other words, existing connections between synsets and therefore between words (via senses) give a reasonably good representation of the lexical meanings, according to open debate in the language processing community, e.g. [19]. This allows us to create a robust representation of words on the basis of the WordNet structure.

2.3. All-pairs shortest paths

Computation of All-pairs Shortest Paths (APSP) is one of the most fundamental problems in graph theory. The objective is to calculate distances between all pairs of vertices in the graph. Multiple algorithms have been proposed to solve this problem for various graph types including directed, undirected, weighted, unweighted once. Their complexity can vary, depending on the type of graph. One of the most commonly used solutions for the problem has been proposed by Floyd [20]. A simple geometrical optimization allowed a decrease in complexity to $O\left(\frac{n^3}{\log(n)}\right)$ [21]. Work by Yijie Han [22] with a complexity of $O\left(n^3\left(\frac{\log(\log(n))}{\log(n)}\right)^{\frac{5}{4}}\right)$ is currently the best score. Besides this, the method proposed by Floyd can also be distributed in a very easy way which facilitates its application to large datasets.

2.4. Sentiment analysis

Sentiment analysis means the assignment of a measure as to what extent a text is positive, neutral or negative. Nowadays, the most commonly used methods for sentiment analysis are classification approaches with classifiers such as Naive Bayes [23–25], Support Vector Machines (SVM) [26–28], Decision Tree [29,30], Random Forest [31] or Logistic Regression [30]. In addition, feature selection can improve classification accuracy by reducing high-dimensionality to low-dimensionality of the feature space. Yousefpour et al. [32] proposed a hybrid method and two meta-heuristic algorithms are employed to find an optimal feature subset.

Another family of solutions available for sentiment analysis are neural network based approaches. Socher et al. [33] proposed a recursive deep model for sentiment analysis using a treebank structure of sentences (i.e. sentences with meta-data describing their syntactic structures). Zhang and LeCun [34] used deep learning for text understanding from character-level inputs all the way up to abstract text concepts, using temporal convolutional networks (ConvNets). What is more, some systems leverage both hand-crafted features and word level embedding features, such as Do2Vec, with the use of classifiers such as SVM [35].

2.5. Transfer learning

Transfer learning makes use of system ability to recognize and apply knowledge extraction (learned in previous tasks) to novel tasks (in new domains) [36]. Interestingly, it has been inspired by human learning behavior. We can often transfer knowledge learned in one situation and adapt it to a new one. Yoshida et al. [37] proposed a model where each word is associated with three factors: domain label, domain dependence/independence and word polarity. The major part of their method includes Gibbs sampling for inferring the parameters of the model, from both labeled and unlabeled texts. Moreover, the method proposed by them may also determine whether each word's polarity is domain-dependent or domain-independent. Zhou et al. [38] developed a solution to cross-domain sentiment classification for unlabeled data. To bridge the gap between domains, they proposed an algorithm called topical correspondence transfer (TCT). TCT is achieved by learning domain-specific information from different areas and applying it to unified topics.

3. WordNet2Vec: wordNet-based natural language representation in the vector space – word vectors

The general idea of the WordNet2Vec method is to transfer knowledge about the lexico-semantic system which is encapsulated in the WordNet relation network into word-based vector structures suitable for further processing. Its main steps are presented in Fig. 1.

A wordnet consists of many synsets and words grouped in them, as well as even more numerous instances of lexical and conceptual relations linking them. Several large wordnets have been

[2] For example, see http://globalwordnet.org/wordnets-in-the-world/.

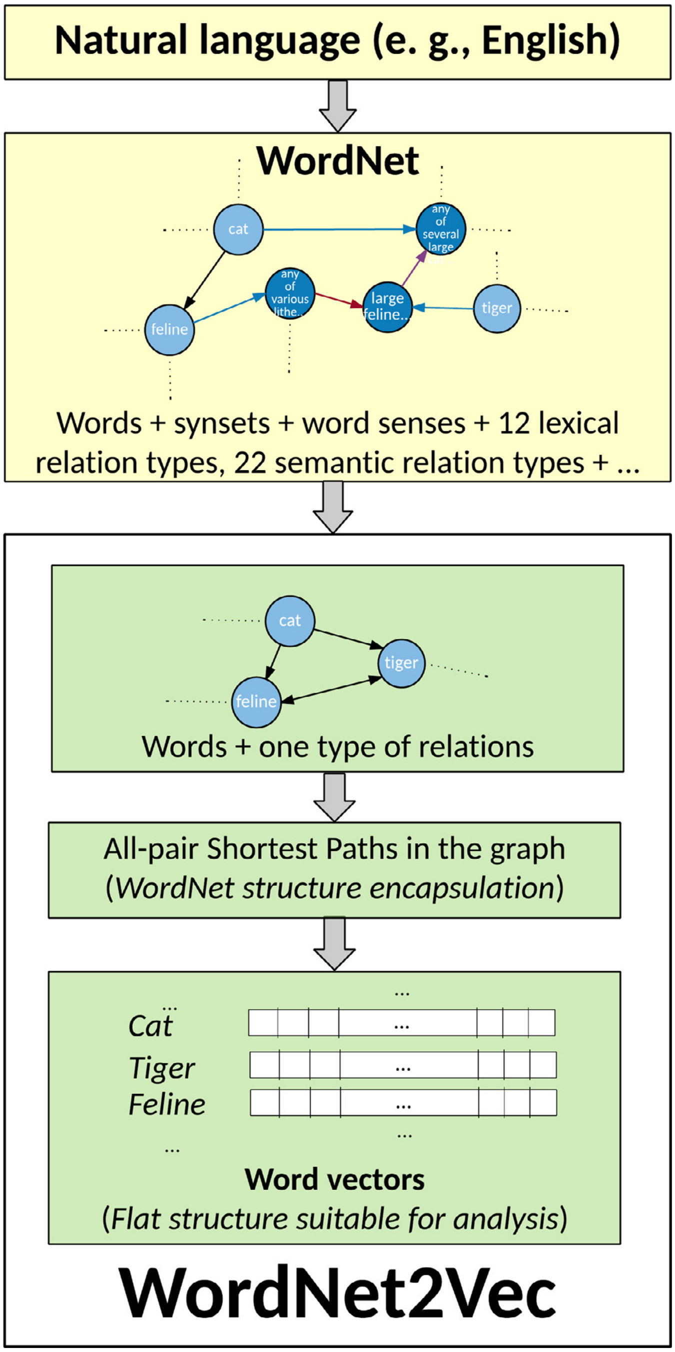

Natural language (e. g., English)
WordNet
cat
any of several large
any of various lithe..
large feline..
tiger
feline
Words + synsets + word senses + 12 lexical relation types, 22 semantic relation types + …
cat
tiger
feline
Words + one type of relations
All-pair Shortest Paths in the graph
(WordNet structure encapsulation)
Cat
Tiger
Feline
…
Word vectors
(Flat structure suitable for analysis)
WordNet2Vec

Fig. 1. Architecture of the WordNet2Vec method.

built manually by linguists for a given language, e.g. WordNet, plWordNet and GermaNet. If a wordnet is large enough, more than a hundred thousand synsets, it may be treated as a reliable and comprehensive representation of the lexical meanings of the given language. Most of the meanings are drawn from the general stylistic register, but there are typically many other meanings from other registers such as popular, colloquial, vulgar and even specialist. Wordnet has a complex network structure and it is hardly suitable for commonly used analytic methods such as machine learning reasoning.

To overcome this limitation, we propose the WordNet2Vec method that provides a set of word vectors embedding a large part of the lexical knowledge stored in the wordnet. The processing starts with a mapping of the wordnet structure onto a simplified graph with only words as nodes and one type of links. A relation between two word lemmas in the simplified graph exists if (1) there exists a direct lexical relation between words in the original wordnet, (2) there exists any semantic relation between two synsets containing the two considered words, (3) both words belong to one synset – are synonyms.

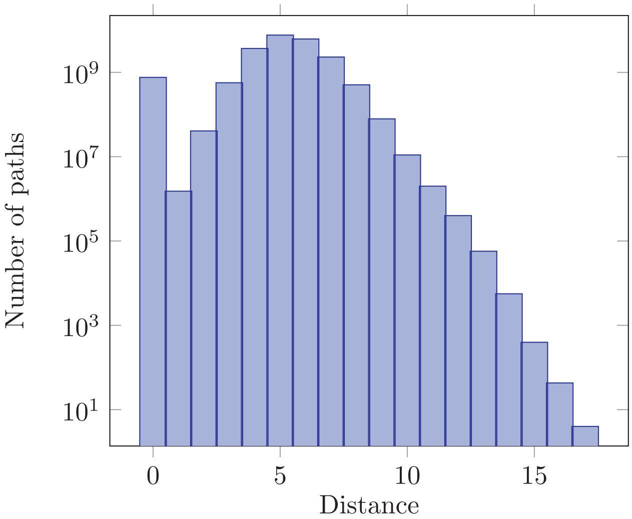


Fig. 2. WordNet shortest paths length distribution.

It is worth remembering that a wordnet is a multiplex with multiple types of connections between multiple types of nodes - a multimodal network. In our approach, all kinds of existing relationships are flattened to achieve a uniplex. This can obviously cause some loss of information. On the other hand, it is not possible to achieve a compact shortest paths representation for each of the layers because in the vast majority of cases they are very sparse and consist of multiple separate components. As a result, the shortest paths matrix would be highly sparse. This could make the representation space improper and with weak potential for generalization. Moreover, the calculation of shortest paths in multiplexes is a separate research problem.

In the next step, a structural measure is applied to evaluate the distance from a given node-word to any other node-word in the simplified graph. We decided to utilize shortest paths for that purpose. It means that we had to compute all-pair shortest paths. The distribution of all pairs shortest paths is depicted in Fig. 2.

Finally, for each word – node in the simplified network – a separate vector is created. Its coordinates correspond to shortest path lengths to all other nodes so the word vector reflects a position of a given word towards all other words in the language. The set of such word vectors comprises the output of the method and can be used for further processing. Such word vector sets express a kind of projection from a lexico-semantic system into a vector space. In particular, the representation constructed provides insights into the semantic relations between words[3].

4. WordNet2Vec implementation – distributed calculation of all-pairs shortest paths in the simplified WordNet graph

To demonstrate the WordNet2Vec method, it was applied to the English WordNet [1]. We have created a simplified (flattened)

[3] It is worth noticing that words in this representation are semantically ambiguous and the representation describes relations between merged meanings. However, words in text are polysemous on average, too.

network based on semantic and lexical relations present in WordNet. Hence, we received a graph composed of 147,478 words interconnected by 1,695,623 links.

The simplified network was utilized to compute all-pair shortest paths in this network, so we obtained 147, 478^2, in other words over 21 billion path lengths.

Wordnet is a multiplex with multiple types of connections between pairs of nodes. In our approach, all kinds of existing relationships are flattened to achieve a uniplex. Such graph aggregation can apparently cause some loss of information, especially regarding connection semantics. On the other hand, layers in all available wordnets are sparse. It would not be possible to achieve a shortest paths representation based only on each layer separately. It makes the representation improper and with weak potential for generalization. Moreover, the calculation of shortest paths in multiplexes constitutes an entirely new research problem.

Because of the size of the simplified network as well as computational and memory complexity of the path calculation task, we decided to use a heterogeneous computational cluster as an environment for our experiments. Nevertheless, further optimization had to be done. In our approach, we used distributed implementation of the Dijkstra [39] algorithm available in our SparklingGraph library [40]. Most of the known solutions for the all-pair shortest paths (APSP) problem have a memory complexity of $O(n^3)$, where n is the number of nodes in the graph [20]. Because of that, we had to do the computations in an iterative way. We used a divide and conquer approach in order to split APSP into smaller problems that can be computed efficiently on our cluster. In each iteration, we computed the shortest paths for 1000 vertices to all vertices in the simplified graph. Afterwards, the results were gathered into a coherent set of word vectors that represented distances between words in terms of graph topology.

As a result, the computed WordNet2Vec matrix for the current version of WordNet had a size of 147,478 * 147,478. While being stored in dense CSV format, it had a size of 41.8GB. It is worth emphasizing, that the computation of the matrix needs to be performed only once for a particular version of Wordnet. Although a matrix of such size might be uncomfortable for storing in memory on low performance computers, it can be easily organized in any lasting structure, for example relational database or < key,value > pairs.

5. Use case: WordNet2Vec application to sentiment analysis

The sentiment analysis use case is presented as one of the possible application areas. The WordNet2Vec method was applied to build a representation of words from a given document and associated opinion. All such vectors were aggregated into a single one, which in turn was utilized as a single case for learning the classifier in a supervised learning scenario. The same was done for a new document to achieve its vector representation. Then, the sentiment orientation of the new document is predicted by the trained model, see Fig. 3.

In detail, sentiment analysis can be performed with an appropriate sequence of processing steps that include text segmentation and lemmatization, vector look-up in the WordNet2Vec matrix for each word in the document, aggregation of vectors for all words within documents, training and testing the dataset split, and finally classifier learning and testing (for example within the same domain or across domains - transfer learning). In order to perform the learning and testing phase properly we should have some sentiment classes assigned to all documents. The overall flow of the sentiment assignment is presented in Fig 3 and all of the steps are discussed below.

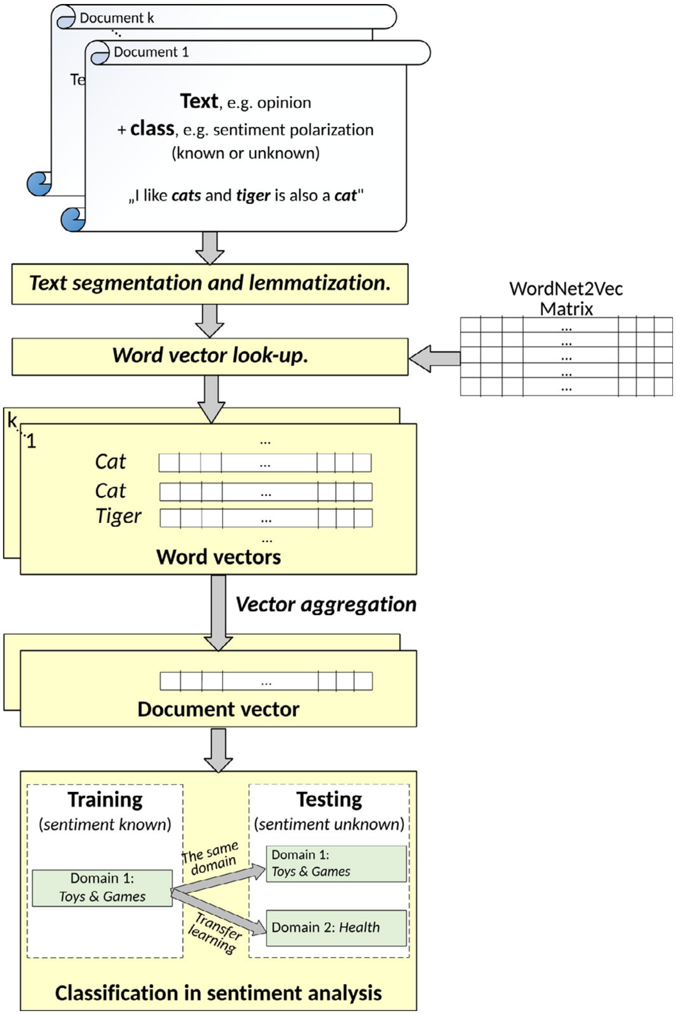


Fig. 3. Exemplary application of WordNet2Vec method to Sentiment Analysis problem.

5.1. Text segmentation and lemmatization

In the first step, some basic natural language processing methods, namely segmentation and lemmatization, were applied. In order to process each document, it has to be segmented into words. Due to the fact that the WordNet2Vec matrix is configured for words in their lemma form, each word from each document must be lemmatized. It needs to be emphasized here that the method is limited only to words that are included in a wordnet. However, due to the fact that a large wordnet, such as WordNet or plWordNet, is a reliable representation of the lexico-semantic system, our method may be suitable for a very large number of application domains.

5.2. Vector look-up in the WordNet2Vec matrix for a word

The method for word vectorization proposed in the paper provides a pre-computed WordNet2Vec Matrix. It contains a vector representation for each word from WordNet. In the next step of the flow, vectors for all words in documents are retrieved from the matrix in $O(1)$ time.

5.3. Aggregation of vectors for all words within documents

As sentiment assignment is performed for documents and not for single words, we have to aggregate all the document words.

Table 1
Statistics of the whole Amazon reviews dataset.

Number of reviews	34,686,770
Number of users	6,643,669
Number of products	2,441,053
Users with > 50 reviews	56,772
Median no. of words per review	82
Time span	Jun 1995 - Mar 2013

Any aggregation method can be applied including averaging or weighting. However, for the purpose of this use case we applied a simple vector sum, see Eq. 1),

$$\overrightarrow{v(d)} = \sum_{i=1}^{|L_d|} \overrightarrow{v(l_i)} \qquad (1)$$

where

$\overrightarrow{v(d)}$ is the WordNet2Vec aggregated representation of document d,
$|L_d|$ denotes the list of lemmas extracted from document d,
l_i is the ith lemma from document d, $l_i \in L_d$
$v(l_i)$ is the ith Wordnet2Vec vector corresponding to lemma l_i.

5.4. Dataset split, classifier learning and testing

Once all the documents possess a single vector representation, they form a dataset appropriate for training and testing classifiers. In order to examine the generalization abilities of the classifiers, two distinct scenarios were designed: learning and testing within the same domain or corpus of texts (the same type of documents) and transfer learning, learning on one domain and testing on another one.

6. Experimental setup

Following the scenario presented in the Section 5, we performed validity tests by examining a sentiment assignment task. In the following sections, we describe the data that were utilized, the details of the experiments and the received results.

6.1. Datasets

We chose data from one data source - the Amazon e-commerce platform [41]. The data spans a period of 18 years, including 35 million reviews up to March 2013. Reviews include product and user information, ratings, and a plain text review. Some basic statistical information about the dataset is presented in Table 1.

The whole experiment was conducted on a selected part of the Amazon data that consisted of 7 domains, namely: Automotive, Sports & Outdoors, Books, Health, Video Games, Toys & Games, Movies & TV. Due to the fact that the distribution of classes is important while interpreting the results of classification validation, the proper histogram is presented in Fig. 4. The domains of the review dataset that were chosen for the experiment are listed in Table 2).

In order to check the accuracy of the proposed methods, we extracted the sentiment orientation from ratings expressed with stars. Ratings were mapped to the following classes: positive, neutral and negative, using 1 and 2 stars, 3 stars, 4 and 5 stars respectively, see Table 3.

6.2. Distributed environment

In our experiments, we utilized a computational cluster that consists of 12 nodes with 24 cores and 60GB of RAM each. Apache Spark 1.5 and our SparklingGraph 0.0.5 were used for distributed computations.

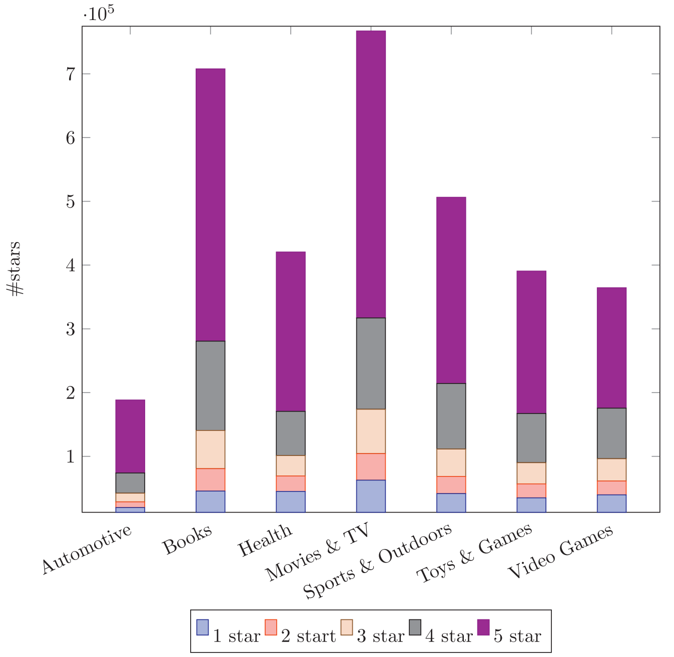


Fig. 4. Distribution of original scores expressed in stars over domains in the Amazon Dataset.

Table 2
Dataset domains used in the experiment.

Domain	Number of reviews
Automotive	174,414
Book	697,225
Health	428,781
Movie TV	765,961
Sports Outdoor	504,773
Toy Game	389,221
Video Game	364,206

Table 3
Star rating mapping to sentiment classes.

Star Score	Sentiment Class
★	Negative
★★	Negative
★★★	Neutral
★★★★	Positive
★★★★★	Positive

6.3. Methods

Our experiments were divided into two distinct groups. The first of them consisted of a classic machine learning evaluation using a 10 times randomly repeated train/test 50:50 split performed on each of the seven mentioned datasets. In the second group of experiments, we used a one versus all transfer learning evaluation. Whenever one domain was used as a training set, it was evaluated on the rest of the domains. Thanks to both scenarios, we examined the quality of a classic sentiment analysis within one dataset domain, and between different domains. In both experimental groups, we used two methods of text vectorization: WordNet2Vec proposed by us and Doc2Vec as a reference.

Doc2Vec expresses a generalization of the Word2Vec algorithm on the document level. The Word2Vec model shows how a word is usually used in a window context according to other words (how words co-occur with each other). The procedure of counting Doc2Vec is very similar to Word2Vec, except it generalizes the model by adding a document vector. There are two methods used in Doc2Vec: Distributed Memory (DM) and Distributed Bag of Words (DBOW). The first one attempts to predict a word given its previous words and a paragraph vector. Even though the context window moves across the text, the paragraph vector does not (hence distributed memory) and allows for some word order to be captured. On the other hand, DBOW predicts a random group of words in a paragraph given only its paragraph vector. In our experiments, we used the Distributed Memory method trained on the Amazon SNAP (see Section 6.1) review dataset. A separate model was trained for each domain and length of a vector equal to 400.

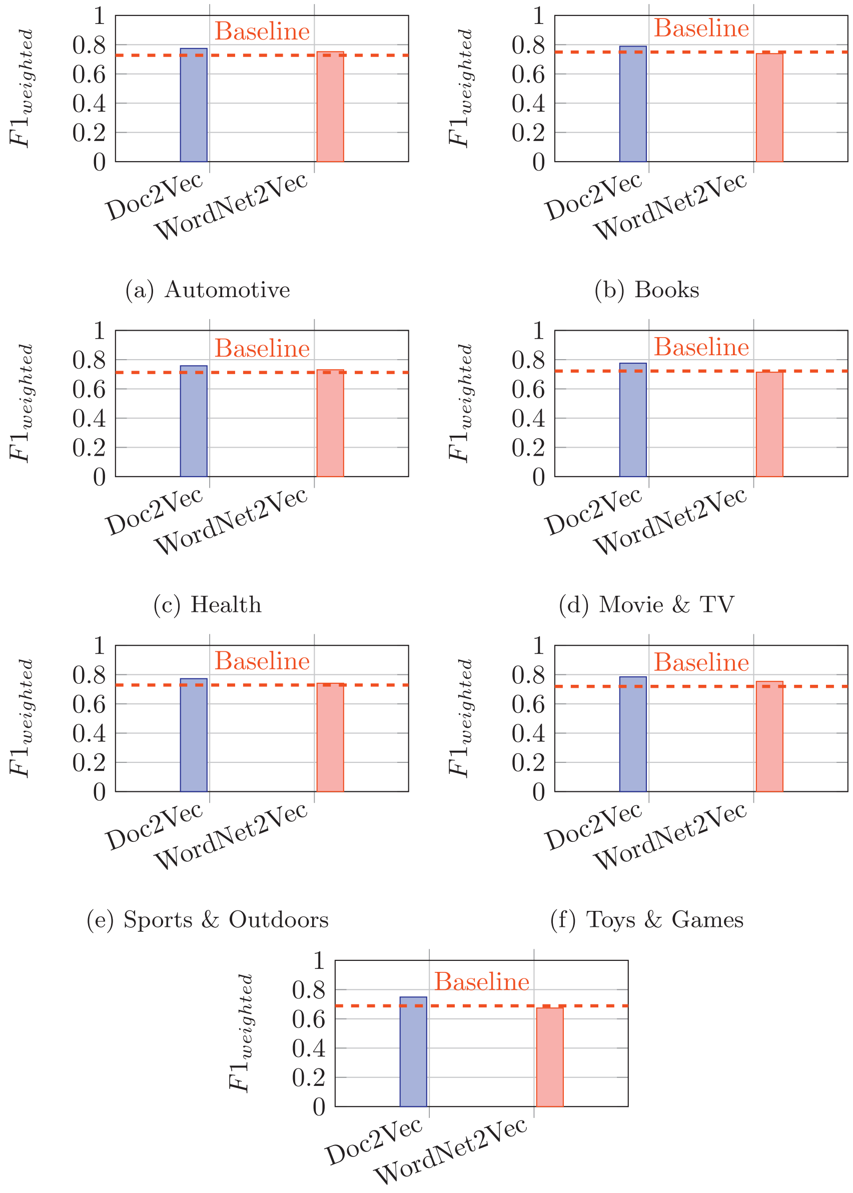
In order to evaluate the standard classification approach we also provided a baseline $F1_{weighted}$ value that would be achieved if we were to use a classifier that always returned a major class.

Logistic regression was selected as a supervised learning model and in order to train it we used a limited memory BFGS algorithm [42]. There was a computational trade-off for vectorized documents in the space size dependent on the Wordnet2Vec Matrix size. Due to the fact that datasets used in the experiments are imbalanced, see Fig. 4), we used a weighted F1 score for the evaluation of the results that properly handle imbalanced class distributions. The weighting was based on the size of each class, see Eq. (2).

$$F1_{weighted} = \frac{\sum_i^k |C_i| \cdot F1_{C_i}}{\sum_i^k |C_i|} \tag{2}$$

where:

- k - number of classes,
- $|C_i|$ - cardinality of set C_i with classification results for class i, i.e. number of cases in class i,
- $F1_{C_i}$ - $F1$ score measure for C_i classification results.



(a) Automotive

(b) Books

(c) Health

(d) Movie & TV

(e) Sports & Outdoors

(f) Toys & Games

(g) Video Games

Fig. 5. In domain classification results.

Table 4
Wilcoxon rank-sum test results for classification.

Method 1	Method 2	H_a	p-value
Doc2Vec	WordNet2Vec	$F1_{weighted}^{Doc2Vec} > F1_{weighted}^{WordNet2Vec}$	0.007813
WordNet2Vec	Baseline	$F1_{weighted}^{WordNet2Vec} \neq F1_{weighted}^{Baseline}$	0.2969

Table 5
Wilcoxon rank-sum test results for transfer learning.

Method 1	Method 2	H_a	p-value
Doc2Vec	WordNet2Vec	$F1_{weighted}^{WordNet2Vec} > F1_{weighted}^{Doc2Vec}$	$6.821 * 10^{-13}$

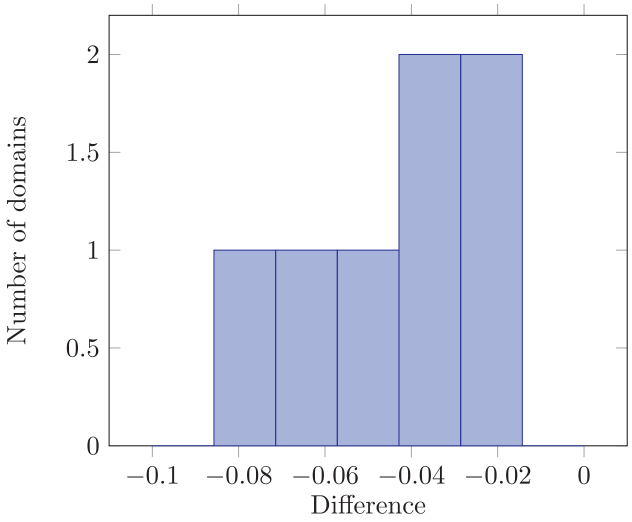


Fig. 6. Histogram of differences between $F1_{weighted}$ measure between WordNet2Vec and Doc2Vec within whole domains. Negative values denote worse results.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{3}$$

In order to compare the two approaches (WordNet2Vec and Doc2Vec), we applied a statistical test on paired measures for each of the experiments. We used the Wilcoxon signed-rank test [43] for pairs of weighted F1 score (Eq. 2) with confidence level $\alpha = 0.05$ (Tables 4 and 5). To provide deeper insight into the differences between the results achieved by the methods we also present histograms of differences between $F1_{weighted}$ of both methods (see Figs. 6, 10).

7. Results

7.1. Classification in one domain

For every domain out of the seven, Doc2Vec is slightly better than the proposed approach, if learning is performed in the same domain (Fig. 5). The difference between the results achieved by Doc2Vec and WordNet2Vec is not as large as that which can be observed in Fig. 6. Nevertheless, it is statistically significant, as was shown by the Wilcoxon rank-sum test (Table 4). It is important to note that a statistical analysis of results also showed that WordNet2Vec is not worse than baseline.

7.2. Transfer learning: generalization ability

A transfer learning approach consists of learning on data from one domain, e.g. *Automotive* and the application of the trained model (testing) on data from another domain, e.g. *Books*. The results achieved in a transfer learning setting show the true power and abilities of the WordNet2Vec method. We observe that our method achieves better results than Doc2Vec (Fig. 7 and 8). The variance of the weighted F1 results in a transfer learning setting for both methods showed that WordNet2Vec is more stable than Doc2Vec (Fig. 9). It is important to notice that differences in results are much greater for the WordNet2Vec method than for the Doc2Vec classification (Fig. 10). In addition, we used statistical tests in order to check the statistical significance of the differences in results (Table 5). The superiority of WordNet2Vec over Doc2Vec is statistically significant.

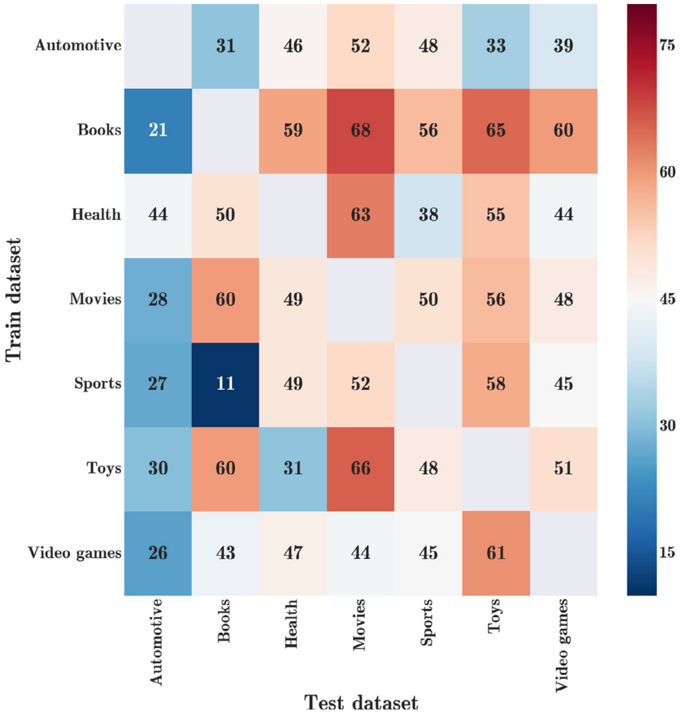


Fig. 7. Transfer learning using Doc2Vec: $F1_{weighted}$ expressed in %.

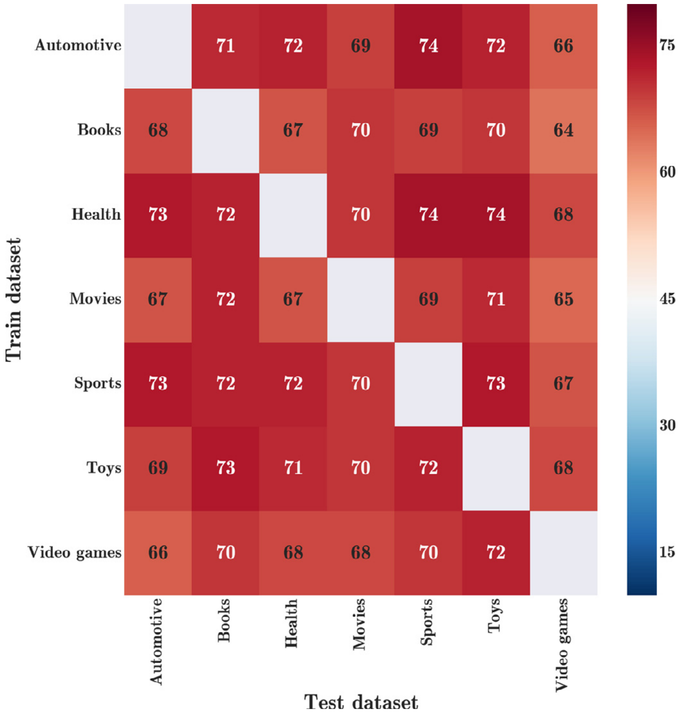


Fig. 8. Transfer learning using WordNet2Vec: $F1_{weighted}$ expressed in %.

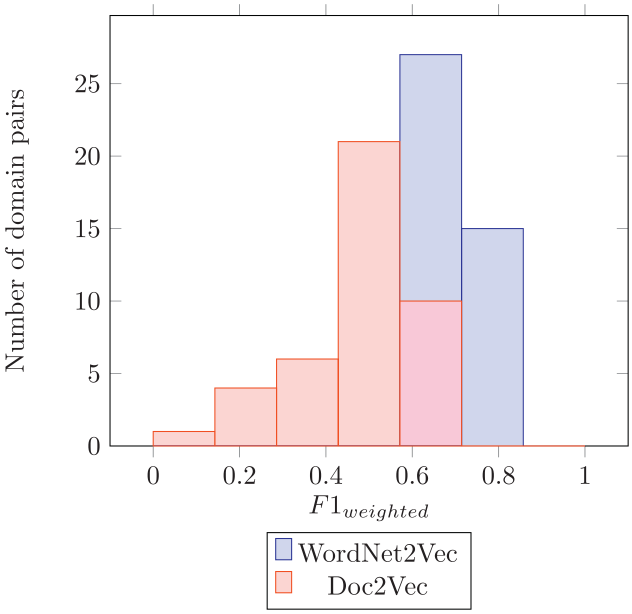


Fig. 9. Histogram of $F1_{weighted}$ in transfer learning – between domains.

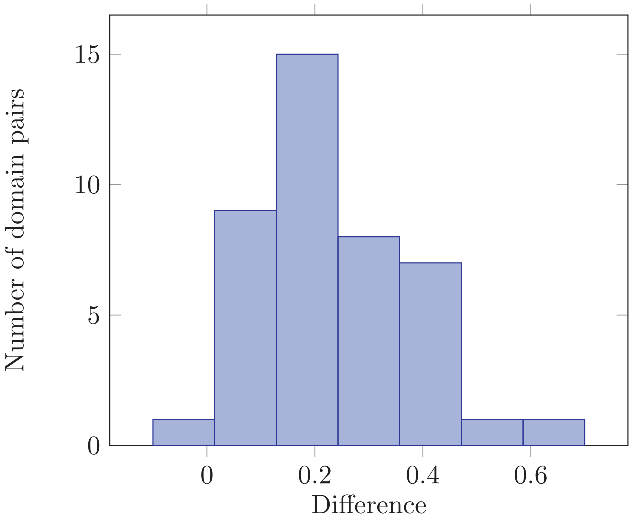


Fig. 10. Histogram of differences in $F1_{weighted}$ classification measure between WordNet2Vec and Doc2Vec in transfer learning scenario. Positive values denote better results.

7.3. Time complexity

The time of calculation should be analyzed separately for both computation phases: data vectorization and classification. In both phases, time varies because of different dataset sizes. The longest vectorization lasted 1 h 45 min, while the shortest was about 9 min. The classification times were longer: from 1 h 43 min to 6 h 16 min. It is important to note that classification times last longer because of the large dimension of word vectors. In our further experiments we will study various methods for dimensionality reduction such as deep autoencoders.

8. Discussion

Since a large wordnet reflects the structure of the lexico-semantic system of a given language it is rarely updated. The vectors once created using the WordNet2Vec method can be treated as a stable and convenient representation of this system and lexical meanings. However, if we would like to operate in a more specific domain such as texts related to biomaterials or those from evolving social media, a wordnet-based knowledge resource should be replaced by another dynamic semantic network. In such cases, it may be necessary to work out some other incremental and efficient WordNet2Vec methods.

In general, the WordNet2Vec validation could be performed with respect to different dimensions: (1) quality for different wordnets (different languages), (2) word vector calculation using various structural measure (we analyzed the lengths of shortest paths), (3) efficiency – algorithms for vector calculations, (4) a method for vector application, in particular vector aggregation for sentences, paragraphs, documents or collections, (5) aggregation utilization dependent on application area (for example sentiment analysis).

9. Conclusions and future work

A novel method, namely WordNet2Vec, for word vectorization that enables us to build a more general knowledge representation of texts on the basis of a large wordnet was presented in the paper. It provides a word representation in the vector space by exploiting distance to any other word in the wordnet network. In order to present the pair-wise word distance, the method calculates all-pairs shortest paths in the wordnet.

The usefulness of the WordNet2Vec method was demonstrated in the sentiment analysis problem, that is to say classification in a transfer learning setting using an Amazon reviews dataset. We compared the WordNet2Vec-based classification of sentiment to the Doc2Vec approach. Doc2Vec proved to be more accurate in a homogeneous setting (learning and testing within the same domain). However, in cases of cross domain application (transfer learning), our method outperformed the Doc2Vec results. Hence, we presented its generalization ability in text classification problems.

There are several potential applications for the WordNet2Vec method such as opinion mining, emotional and controversy analysis, recommendation systems based on textual content, social media analysis, text summarization, document comparison, antiplagiarism systems, and market basket analysis based on product names and descriptions.

In future work, we would like to investigate different methods of combining word vectors into documents (including various forms of vector normalization), treat WordNet as a multiplex network while calculating shortest paths, reduce the feature space of WordNet2Vec Matrix and validate our method on different sources of data such as Twitter or Facebook. In addition, we are working on methods for shortening the time of All-pairs Shortest Paths computation using various heuristics which will enable us to evaluate WordNet2Vec on larger networks such as BabelNet [44].[4]

Acknowledgments

The work was partly supported by The National Science Centre, decision no. DEC-2013/09/B/ST6/02317, the European Commission under the 7th Framework Programme, Coordination and Support Action, Grant Agreement Number 316097, Engine project, the Faculty of Computer Science and Management, Wrocław University of Science and Technology statutory funds, the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 691152, RENOIR project and the Polish Ministry of Science and Higher Education fund for supporting internationally co-financed projects in 2016–2019 (agreement no. 3628/H2020/2016/2). The calculations were carried out

[4] http://www.wcss.wroc.pl.

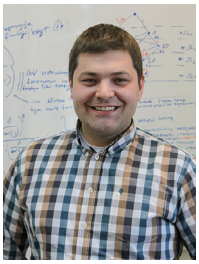
in the Wrocław Centre for Networking and Supercomputing, grant No 177.

References

[1] G.A. Miller, Wordnet: a lexical database for english, Commun. ACM 38 (11) (1995) 39–41.
[2] M. Piasecki, S. Szpakowicz, B. Broda, A Wordnet from the Ground Up, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2009.
[3] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents., in: ICML, vol. 14, 2014, pp. 1188–1196.
[4] C.O. Alm, D. Roth, R. Sproat, Emotions from text: machine learning for text-based emotion prediction, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, in: HLT '05, Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 579–586, doi:10.3115/1220575.1220648.
[5] R. Brachman, H. Levesque, Knowledge Representation and Reasoning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
[6] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 2, Pearson Education, 2003.
[7] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[8] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781. (2013).
[9] T. Luong, R. Socher, C.D. Manning, Better word representations with recursive neural networks for morphology., in: CoNLL, 2013, pp. 104–113.
[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (Almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537, doi:10.1145/2347736.2347755.
[11] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[12] R. Lebret, R. Collobert, Word Emdeddings through Hellinger PCA(2013) 9.
[13] P. Dhillon, J. Rodu, D. Foster, L. Ungar, Two step CCA: a new spectral method for estimating vector models of words (2012).
[14] R. Lebret, R. Collobert, Rehabilitation of count-based models for word vector representations, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9041, 2015, pp. 417–429.
[15] M. Baroni, G. Dinu, G. Kruszewski, Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 238–247.
[16] A. Österlund, D. Ödling, M. Sahlgren, Factorization of latent variables in distributional semantic models, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015, 2015, pp. 227–231.
[17] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, Adv. Neural Inf. Process. Syst. (NIPS) (2014) 2177–2185.
[18] WordNet 3.1, 2016, [Online; accessed 10-May-2016].
[19] M. Piasecki, S. Szpakowicz, M. Maziarz, E. Rudnicka, PlWordNet 3.0 - Almost there, in: Proceedings of the 8th Global WordNet Conference, GWC 2016, 2016.
[20] R. Floyd, Algorithm 97: shortest path, Commun. ACM 5 (6) (1962) 345.
[21] T.M. Chan, All-pairs shortest paths with real weights in $o(n^3/log(n))$ time, in: Algorithms and Data Structures, Springer, 2005, pp. 318–324.
[22] Y. Han, An $o(n^3(loglog(n)/log(n))5/4)$ time algorithm for all pairs shortest paths, in: Algorithms–ESA 2006, Springer, 2006, pp. 411–417.
[23] P. Bo, L. Lillian, V. Shivakumar, Thumbs up?: sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10, Association for Computational Linguistics, 2002, pp. 79–86.
[24] Q. Ye, Z. Zhang, R. Law, Sentiment classification of online reviews to travel destinations by supervised machine learning approaches, Expert Syst. Appl. 36 (3, Part 2) (2009) 6527–6535.
[25] V. Narayanan, I. Arora, A. Bhatia, Fast and accurate sentiment classification using an enhanced naive bayes model, CoRR abs/1305.6143 (2013).
[26] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 142–150.
[27] X. Bai, Predicting consumer sentiments from online text, Decis. Support Syst. 50 (4) (2011) 732–742.
[28] M. Gamon, Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis, in: Proceedings of the 20th International Conference on Computational Linguistics, Association for Computational Linguistics, 2004, p. 841.
[29] M. Koppel, J. Schler, The importance of neutral examples for learning sentiment, Comput. Intell. 22 (2) (2006) 100–109.
[30] L. Augustyniak, P. Szymanski, T. Kajdanowicz, W. Tuliglowicz, Comprehensive study on lexicon-based ensemble classification sentiment analysis, Entropy 18 (1) (2016) 4, doi:10.3390/e18010004.
[31] H. Parmar, S. Bhanderi, G. Shah, Sentiment mining of movie reviews using random forest with tuned hyperparameters, 2014.
[32] A. Yousefpour, R. Ibrahim, H.N.A. Hamed, T. Yokoi, Integrated Feature Selection Methods Using Metaheuristic Algorithms for Sentiment Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 129–140.
[33] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, et al., Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), vol. 1631, 2013, p. 1642.
[34] X. Zhang, Y. LeCun, Text understanding from scratch, arXiv preprint arXiv:1502.01710. (2015).
[35] H. Liang, R. Fothergill, T. Baldwin, Rosemerry: A baseline message-level sentiment classification system, SemEval-2015 (2015) 551.
[36] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[37] Y. Yoshida, T. Hirao, T. Iwata, M. Nagata, Y. Matsumoto, Transfer learning for multiple-domain sentiment analysis-identifying domain dependent/independent word polarity., in: AAAI, 2011.
[38] G. Zhou, Y. Zhou, X. Guo, X. Tu, T. He, Cross-domain sentiment classification via topical correspondence transfer, Neurocomputing 159 (2015) 298–305.
[39] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1) (1959) 269–271.
[40] R. Bartusiak, T. Kajdanowicz, SparklingGraph: large scale, distributed graph processing made easy, 2016, http://sparkling.ml.
[41] J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, 2013, pp. 165–172. ACM ISBN 978-1-4503-2409-0.
[42] D.C. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization, Math. Program. 45 (1–3) (1989) 503–528.
[43] R.H. Randles, Wilcoxon signed rank test, Encyclopedia of Statistical Sciences, 1988.
[44] R. Navigli, S.P. Ponzetto, BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artif. Intell. 193 (2012) 217–250.

Roman Bartusiak is a Ph.D. candidate in Computer Science at the Wroclaw University of Science and Technology, Poland. He received MSc in Computer Science from the Wroclaw University of Technology in 2015 with distinction. Roman works in the area of distributed procesing of networked data with main interest in graph partitioning. He is also a successful entrepreneur.

Łukasz Augustyniak is a Ph.D. student in Computer Science at the Wroclaw University of Science and Technology, Poland. He received MSc in Computer Science from the Wroclaw University of Technology in 2013 with distinction. He also received an MA in Law from Wroclaw University in 2014. Lukasz works in the area of Natural Language Processing and applies Machine Learning methods to solve business problems.



Tomasz Kajdanowicz received his M.Sc. and Ph.D. degrees in computer science with honours, both from Wroclaw University of Technology, Poland, in 2008 and 2012, respectively. His PhD was given The Best Polish Ph.D. Dissertation Award in 2012/2013, European Association for Artificial Intelligence – Polish Chapter, 2014. Recently, he serves as an assistant professor of Wroclaw University of Science and Technology at the Department of Computational Intelligence, Poland. He was a co-chair and a member of organizing committee of multiple workshops and summer schools on Data Science, Artificial Intelligence, Machine Learning and Big Data. He regularly serves as a member of international programme committees and the reviewer for prestige international journals and scientific conferences in the area of Social Networks, Machine Learning, Prallel Computing, Data and Knowledge Management, Big Data, Intelligent Systems. He has authored over 70 research articles in a variety of areas related to Ensemble Classification, Collective Classification, Machine Learning, Social Network Analysis, Collaborative Systems, Data Mining, Recommender Systems and Big Data, including principals of Map-Reduce and Bulk Synchronous Parallel paradigms. He also initialized and led 7 research projects in cooperation with commercial companies, including large international corporations. Tomasz holds ITIL v3 Foundation Certificate and multiple professional certificates from project management and data science.