Sem2Vec: Semantic Word Vectors with Bidirectional Constraint Propagations

Taygun Kekeç, David. M. J. Tax

Abstract—Word embeddings learn a vector representation of words, which can be utilized in a large number of natural language processing applications. Learning these vectors shares the drawback of unsupervised learning: representations are not specialized for semantic tasks. In this work, we propose a full-fledged formulation to effectively learn semantically specialized word vectors (Sem2Vec) by creating shared representations of online lexical sources such as Thesaurus and lexical dictionaries. These shared representations are treated as semantic constraints for learning the word embeddings. Our methodology addresses size limitation and weak informativeness of these lexical sources by employing a bidirectional constraint propagation step. Unlike raw unsupervised embeddings that exhibit low stability and easily subject to changes under randomness, our semantic formulation learns word vectors that are quite stable. An extensive empirical evaluation on the word similarity task comprised of eleven word similarity datasets is provided where our vectors suggest notable performance gains over state of the art competitors. We further demonstrate the merits of our formulation in document text classification task over large collections of documents.

Index Terms—Word Embeddings, Semantic Embeddings, Embedding Stability, Thesaurus, Constraint Propagation

1 Introduction

Along with the development of modern computational devices, the amount of digitalized data is increasing fast-paced. The improvements of the electronic storage and processing technology has increased our capacity to collect, create, filter and distribute more and more information. Nevertheless, the price to pay for this accumulation is the information overload [1], the difficulty of analyzing and interpreting the vast amount of data and making effective long-term decisions. Therefore, the development of sophisticated natural language tools is indispensable to overcome this overload and enable easier decision making.

Document classification is an excellent example for illustrating the information overload in which the task is to categorize documents into a predefined set of labels given a large collection of documents. Considering the size of document collection, employing human labour for answering these questions is mostly daunting, exhaustive and highly inefficient. Combatting aforementioned information intensive tasks using computational methodologies requires the development of word representations. Traditional natural language processing was centered around naive, frequency based features to represent the words [2]. Nevertheless, these models are primitive and do not generalize to many lingual tasks due to their simple counting based nature. The main drawback of these frequency based features is that they do not consider the context of words explicitly. The importance and necessity of considering the context were introduced with the notion of the Distributional Hypothesis [3] [4] which claims that words mostly acquire their meanings through the context they are used in. Foundation of this hypothesis provided a formal ground to learn novel representations that do more than simple counting and addressed the contextual cues as well. A stream of unsupervised techniques that learn the cooccurrence statistics of the data was developed. Word embedding approaches (a.k.a. vector space learning) [5] are such techniques which representations of words are optimized such that these words and their context words are located nearby in the embedding space. Rather than simple counting arguments, these approaches incorporate learning the words and their contexts from the corpora and generalize significantly better than their traditional frequency features. They not only discover intrinsic aspects and variations of the underlying data at hand but also allow variations in the optimization, and embedding prior knowledge. Recent studies have shown that the resulting word vectors are usable for a diverse set of natural language applications. They yield substantial representation power and proven to be much more successful in many lingual tasks than the frequency representations [6].

The optimization of word embedding vectors is usually performed on large unstructured corpora. An efficient word embedding algorithm is expected to learn the structure and regularities in the language without any further guidance from experts. Unfortunately, these algorithms share the common drawback of unsupervised learning: the learned embeddings are not necessarily optimized for the subsequent predictive task [7]. Generally speaking, when one wishes to optimize the vectors for a semantic task of interest, the Distributional Hypothesis is insufficient [8]. Words occurring in similar contexts may exhibit weak or no semantic relevance, and the learned vectors do not necessarily encode features that capture semantic similarities [9]. These limitations naturally ask for the development of novel learning methodologies whilst keeping the prominent benefits of the unsupervised learning.

Many formulations have been proposed to tackle this

• *Taygun Kekeç and David. M. J. Tax are with the Faculty of Electrical Engineering, Mathematics and Computer Science, Pattern Recognition and Bioinformatics Laboratory, Technical University of Delft, The Netherlands, Van Mourik Broekmanweg 6, 2628 XE.*
E-mail: taygunkekec@gmail.com, D.M.J.Tax@tudelft.nl

problem. Incorporation of knowledge graphs [10] to the embedding network, augmenting the objective with extra relatedness annotations [11], and the extraction of word senses from lexical dictionaries [12] are solutions to embed these general purpose vectors to a semantic space. The work in [13] constructs an unsupervised random field over the semantic associations to retrofit (post-process) the word vectors. These works jointly learn embeddings, given a knowledge source, and they show improvements over unsupervised, raw embeddings. Nevertheless, the utilization of semantic sources is not straightforward. Each semantic source has a degree of semantic relevance to the task, and usually, sources with high semantic relevance are scarce. Addressing these points require the design of novel unsupervised objectives that exploit auxiliary semantic content.

In this work, we propose a novel approach to address the aforementioned issues and effectively learn embeddings with semantic specialization. Briefly, the main contributions of this paper are:

- We adopt the view that each lexical source exhibits a different degree of semantic relevance. Thus, we create a shared representation of Thesaurus and online lexical dictionaries, and then fuse their content into the learning. This is done by introducing them as semantic constraints with varying strengths called heavy and light constraints, in order to restrict the original embedding problem.
- We introduce a bidirectional propagation technique over constraint sets where 1) bottom up propagations increase the number of heavy constraints 2) top down propagations improve the overall reliability of light constraints. This strategy constructs a well-behaving objective for learning semantically specialized embeddings. The full pipeline of our learning methodology is visually illustrated in Figure 1.
- It is difficult to train embeddings that take long range dependencies into account. Unsupervised embeddings are known to be highly unstable when trained under large window sizes. Compared to the original embedding problem, embeddings trained with our semantic constraints yield quite stabilized solutions for all query sets.
- Our empirical findings suggest significant improvements on semantic tasks. More precisely, we measure the word similarity performance on a wide set of word embedding baselines using a test collection of eleven datasets. The weighted average of Spearman correlation score shows a 4.3% improvement upon the state of the art solutions. When embeddings are trained on a smaller subset of Wikipedia 2017, the improvement over the competitors is even 7.4%. Although semantic vectors were not specifically trained for optimizing a document classification objective, we further evaluate our vectors on the text document classification, and obtain noticeable performance gain in multi-class classification tasks.

The rest of this paper is structured as follows: in Section 2, we first detail the limitations of the distributional hypothesis for specializing to semantics, and then explain our pipeline of learning semantic embeddings. In Section 3, we provide our experimental setup, model selection routines, followed by stability and quantitative results on the evaluation tasks. In Section 4, we conclude our work and provide discussions.

2 Semantic Word Vectors with Bidirectional Constraint Propagations

In this section, we introduce the preliminary word-context learning problem, followed by construction of our heavy and light constraints. We then conclude by detailing our bidirectional constraint propagations.

2.1 Word Vector Models

A large set of word embedding approaches use the following generalized objective function:

$$J(w,c) = \ell(w,c) - \sum_{c_N \in V_c} \ell(w,c_N) \tag{1}$$

where w is a target word in vocabulary V_w, and c is a context word in a vocabulary set V_c. Further we define $\vec{\boldsymbol{w}}$ and $\vec{\boldsymbol{c}}$ as the vector representations of w and c, respectively. Then $\ell(w,c) = \log\left(1+\exp(-\vec{\boldsymbol{w}}^T\vec{\boldsymbol{c}})\right)$ is the logistic loss function. The first logistic loss term in the objective penalizes the dissimilarity of $\vec{\boldsymbol{w}}$ and $\vec{\boldsymbol{c}}$. The second term normalizes the first quantity by making comparisons over different contexts. In practice, evaluating the normalization over all possible contexts is impractical, usually manageable sized approximations have resorted.

Skip Gram approximation [14], approximates the second term of this objective function by randomly sampling some negative context c_N which are forced to have a vector $\vec{\boldsymbol{c}}_N$ that is most dissimilar to $\vec{\boldsymbol{w}}$. The total loss over the corpus is then simply the sum of i.i.d. (w,c) word context pairs. Without loss of generality, this objective is an application of the distributional hypothesis: if a word w occurs together with context c, they should have similar vectors. This relation gets stronger if they co-occur more in the observed corpus.

2.2 Semantic Word Vector Specializations

Learning embeddings using Equation 1 attained reasonable success for the general tasks. However, when we want to specialize embeddings for semantic relations, we notice several problems with this approach. First, it is unlikely to observe a word and its semantic partner (e.g. its hypernym, hyponym or synonym) together in a local window. The semantic partner usually occurs with it usually only through long-range dependencies [15]. The second difficulty is that unsupervised objective has no preference over any pairs. Without explicitly telling the model which loss pairs are semantically valuable, most of the loss pairs are those that do not necessarily have strong semantic informativeness. For instance, consider the sentence: *"my dog is cute but aggressive, and likes to eat high quality food"*. According to the Distributional Hypothesis, the meaning of *dog* and surrounding words like *aggressive*, *is* should be closer to *dog* since they occur in close context. Although this argument is partially true, the semantic relation between *dog* and *food*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2942021, IEEE Transactions on Knowledge and Data Engineering

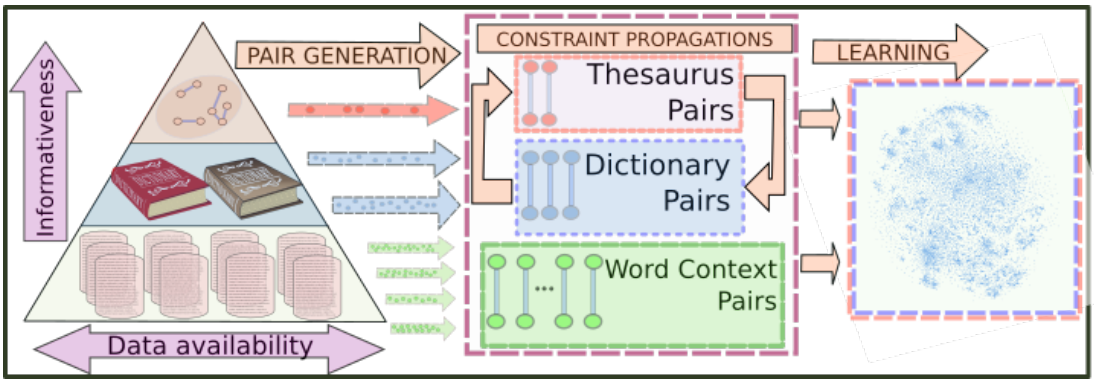


Fig. 1: Our proposed word embedding pipeline (best viewed in color). We first generate various levels of word context pairs using a triplet of sources: unsupervised corpora, lexical dictionaries and Thesaurus. We then treat upper lexical sources as optimization constraints and perform bidirectional propagations between the constraint sets to maximize the learning efficiency. Our final word embeddings are highly suited for semantic tasks.

gets weaker due to being far away, or even lost after long-ranges.

These two problems arising from the hypothesis can be addressed by designing an objective function, such that it weights semantically valuable pairs heavier than the rest. This is possible by leveraging auxiliary semantic information that specifies feasible regions of the objective function in Equation 1. But which pairs are more semantically valuable? From a computational linguistics point of view [16], semantic value is understood via the concept of *Information Content* which suggests that general entities present less information than the more specialized entities and relations. In other words, abstract relations of semantics has high information content whereas raw cooccurrences provide significantly less amount of semantic content.

Consider the relations of two words w and c. These words can cooccur in a domain such as raw noisy corpora, a dictionary, or in a thesaurus. As information content suggests, these relations differ in their semantic abstraction level: there is a clear distinction between the raw text co-occurrence relation and a dictionary sense relation, the latter being a stronger one.

Lexical Dictionary. The lexical dictionary is a rich source containing sense definitions of the words where one can extract significant clues what the meaning of the word is with respect to other words. For example, consider the definition of word *tower* in Table 1. There are commonalities across the definitions of the same word. For our purposes, we extract all word-context pairs from the dictionary definitions and denote an extracted elements as *sense pairs*. Let $\mathbf{D}$ be the dictionary. We formulate a sense pair as a constraint to the semantic similarity of (w, c). We penalize the dissimilarity of $\vec{\boldsymbol{w}}$ and $\vec{\boldsymbol{c}}$ under the logistic loss, and form a constraint $\big(\ell(w, c) \leq \tau\big)\, \mathbb{1}_{\mathbf{D}}(w, c)$. We then use standard Karush Kuhn Tucker (KKT) conditions [17] to treat this dictionary constraint as an objective term:

$$J_{\mathbf{D}}(w, c) = \big(\ell(w, c)\mathbb{1}_{\mathbf{D}}(w, c)\big) \tag{2}$$

where τ disappeared since it neither depends on w nor c.

TABLE 1: Dictionary and Thesaurus content for the query word *tower*.

Source	Content
Dict [1]	*a building or structure high in proportion to its lateral dimensions, either isolated or forming part of a building.*
Dict [2]	*A tower is a tall, narrow building, that either stands alone or forms part of another building such as a church or castle.*
Thesaurus [3]	*belfry - castle - citadel - column - fort - fortification - fortress - keep - lookout ...*

[1] http://www.dictionary.com [2] http://en.oxforddictionaries.com [3] http://www.thesaurus.com

Here, $\mathbb{1}_{\mathbf{D}}(w, c)$ is the dictionary indicator function:

$$\mathbb{1}_{\mathbf{D}}(w, c) = \begin{cases} 1 & (w, c) \in \mathbf{D} \\ 0 & otherwise \end{cases}$$

for the cooccurence of (w, c) in dictionary set $\mathbf{D}$. Since dictionary pairs are considered as constraints to raw cooccurrences: we call $J_{\mathbf{D}}(w, c)$ as the *light constraint* objective.

Thesaurus. Thesaurus is a reference source where a word is explained in a concise manner using a small subset of related words. In contrast to a dictionary, thesaurus does not treat words in alphabetic order. Also, dictionary definitions can contain syntactic or semantic relevance, yet Thesaurus only accounts for semantic relations. These relations are very abstract and may contain synonyms and antonyms. The pure semantic nature of the Thesaurus means that pairs generated from it have higher information content than dictionary pairs. For the word *tower* the last row of Table 1 shows the query result from a thesaurus. We see that the Thesaurus definition of *tower* is much condensed compared to dictionary content, and mostly includes concrete building objects having structural similarities.

Similarly to the dictionary definitions, we extract pairs and denote $\mathbf{T}$ as the set of Thesaurus pairs to further constrain the embedding problem. That is we penalize the dissimilarity under the logistic loss and form the heavy constraint $\big(\ell(w, c) \leq \kappa\big)\, \mathbb{1}_{\mathbf{T}}(w, c)$ and use it in the objective term through the Thesaurus:

$$J_{\mathbf{T}}(w, c) = \big(\ell(w, c)\mathbb{1}_{\mathbf{T}}(w, c)\big) \tag{3}$$

where $\mathbb{1}_{\mathbf{T}}(w,c)$ is the indicator function for the constraint pair in Thesaurus $\mathbf{T}$. Here we denote constraint pairs in $J_{\mathbf{T}}(w,c)$ as *heavy* constraints meaning that they have to be strongly satisfied during the optimization.

Since Thesaurus content is semantically more informative than lexical dictionary content, the reader may question why hard optimization constraints with equality conditions are not employed in our formulation. Such hard-constraint strategies introduce problematic issues when we have hundred thousands of constraints in the learning problem. The probability of constraint violation, and yielding an infeasible problem gets higher with a large set of hard constraints. Furthermore, we do not specifically require hard constraints since we don't want words to be identical to their synonyms. This is because there are already nuances between different synonyms of a particular word because only a subset of synonyms can be substituted for a word in a context [18]. Thus, our heavy constraints are still mathematically soft constraints like light constraints, and can still preserve nuances of synonyms, rather than removing them.

2.3 Bidirectional Constraint Propagations

In the last subsection, we constructed light constraints from the lexical dictionary and heavy constraints from a Thesaurus source. These constraints restrict the maximization of the objective function over a subspace that semantic relations hold. Unfortunately, sets with high information content are very much limited in size as Figure 1 demonstrates. On the other hand, dictionary pairs are relatively less informative but potentially yield to an order of magnitude more constraints. The main idea in this subsection is that these two lexical sources can mutually benefit from each other. Promoting reliable sense pairs can increase the number of heavy constraints and Thesaurus can create some new constraints for the dictionary to increase its average informativeness. For promoting a light constraint to heavy, we define two rules:

- *definitional symmetry*. The dictionary sense definition pair is denoted as symmetric if $(w,c) \in \mathbf{D}$, and $(c,w) \in \mathbf{D}$. This indicates a very strong semantic relation, and we promote this pair to be an element of $\mathbf{T}$.
- *expert agreement*. Assume we have d dictionaries collected from independent sources representing our large dictionary set $\mathbf{D} = \{\mathbf{D}_1, \mathbf{D}_2, ...\mathbf{D}_d\}$. If the definition of word w contains c in multiple dictionaries, then (w,c) pair is an expert sense. According to this rule, the word *tower* in Table 1 has *building* in its definition across multiple dictionaries. Hence, *tower-building* is an expert agreed sense. We augment $\mathbf{T}$ with these pairs.

In the next step, we query elements of $\mathbf{T}$ and stochastically apply semantic association rules to form new pairs. While there exists ontology knowledge based association rule techniques [19], we adopt a low complexity association rule that is if a pair (w_1,c) are (w_2,c) both in $\mathbf{T}$, we then create (w_1,w_2) pair and add it to the set $\mathbf{D}$. We perform these associations for a tiny number of KNN neighbourhood and increase the average information content of the light constraint set.

2.4 Generalized Negative Sampling

Our learning step is similar to the way a majority of the embedding methods are trained [20], [21], [22]. We adopt Negative Sampling formulation in which a noise distribution generates random word context pairs, denoted as negative samples. In this manner, the learning consists of discriminating between positive word-context pairs and negative pairs. Negative sampling contribution term is:

$$J_N(w) = \sum_{c_N \in V_c} \ell(w, c_N)$$

where c_N is a random context word sampled from the negative distribution. In the standard negative sampling, there is a probability that all word context samples can be negative samples. However, we must ensure that random contexts are guaranteed to be negative samples. In other words, we must ensure that w and c_N are not related. Since we know that pairs obtained from $\mathbf{T}$ and $\mathbf{D}$ are strongly related, there is still a non-zero probability to sample such pair. To circumvent this issue, we perform Generalized Negative Sampling and do not negative sample a context word if the pair is in the lexical sources:

$$J_N(w) = \sum_{\substack{c_N \in V_c \\ (w,c_N) \notin \mathbf{T}, (w,c_N) \notin \mathbf{D}}} \ell(w, c_N) \tag{4}$$

This generalized sampling strategy discards a small fraction of the negative samples[4] from the objective but we have experimentally found out that it yields better learning. Our final objective function is the sum of the pair loss, J_N, J_T and J_D:

$$J(w,c) = \ell(w,c) + \lambda_{\mathbf{T}} J_T(w,c) + \lambda_{\mathbf{D}} J_D(w,c) - J_N(w) \tag{5}$$

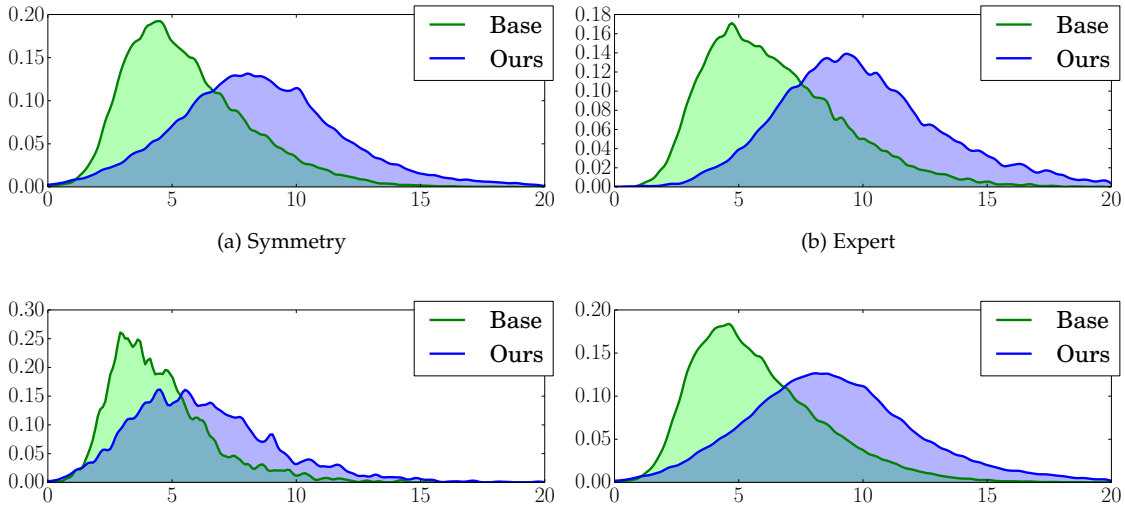
Practically, we distinguish between our two soft constraint sets by using $\lambda_{\mathbf{T}}$ such that $\lambda_{\mathbf{T}} \geq \lambda_{\mathbf{D}}$ holds for any case. We then obtain the global objective by simply summing over all i.i.d (w,c) pairs in the corpus.

3 Experimental Results

We train our embedding models using the latest Wikipedia 2017 July snapshot containing 4.5B tokens. We extract the vocabulary from the corpus which gives us approximately a vocabulary of 2.3M words. Our corpus processing follows the state of the art practices for Wikipedia which we use the standard preprocessing scripts and remove XML and HTML tags to obtain the raw text [23]. For a fair evaluation, all common embedding training parameters are set as in [24], where we remove words that occur less than 5 times, set the window size to 5, number of negative samples to 5, and the corpus is processed for 5 epochs. The initial learning rate is set to the same value for the methods and Stochastic Gradient Descent is used as the optimization algorithm.

We use the publicly available Cambridge, Oxford, Collins, Dictionary.com and Longman English dictionaries to obtain word definitions. We crawl each dictionary with web requests and parse the HTML contents using regular expressions to get word definitions from Cambridge, Oxford, Collins, Dictionary.com. Unlike other dictionaries, the

4. Normalization by the number of negative samples is usually employed in $J_N(w)$, which is omitted here for notational convenience.



(a) Symmetry (b) Expert

(c) Association (d) Symmetry + Expert + Association

Fig. 2: Kernel Density Estimate fits to inner products for SG (base) and our model. x-axis is the inner product value and y-axis is the density estimate. Fits to the inner product for the a) symmetry pairs b) expert pairs and c) association pairs d) distribution for an aggregation of these constraints. Learning with our model corrects the inherent skew, and yields a Gaussian-peaked concentration for the inner products.

Longman Dictionary provides an Application Programming Interface allowing to directly get the word definitions. The definition texts are preprocessed similarly to the input corpus such that only alphanumeric characters are present. For obtaining more informative pairs, we reduce the redundancy by removing the stop-words from dictionary definitions. After collection of all definitions from all dictionaries, as the purpose is not word sense disambiguation, we concatenate all senses into a single list. For a Thesaurus source, we crawl the contents of Online Thesaurus[5] where each word is provided a list of synonyms. After the initial construction of our heavy and light objective terms using pairs from our sources, we apply the bidirectional constraint propagations.

Our performance benchmark includes comparison of the following word embedding architectures:

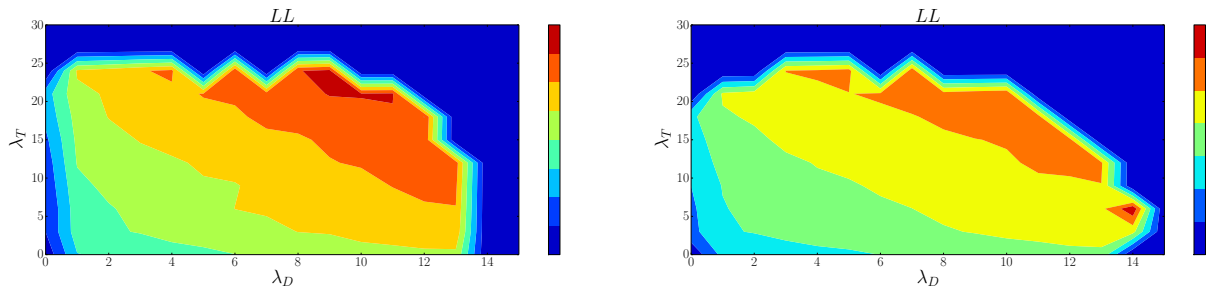
- Skip Gram (SG) [23]: The vanilla baseline using Skip Gram architecture of Word2Vec which word vectors are trained by predicting the context word from a target word of a sentence. Default parameters are used.
- Continuous Bag of Words (CBoW) [14]: state of the art architecture representing the context vectors as the bag of words around the target word. This architecture is faster to train than SG, and competitive to beat in large scale datasets.
- Dict2Vec (D2V) [12]: embedding architecture that uses dictionary definitions. As their approach requires a preliminary training step of word embeddings, we first pretrain the embeddings to obtain initial vectors. We then follow the necessary steps: use pretrained vectors to specify and promote the constraint pairs and set parameters to the best reported results.
- FastText (FT) [24]: embedding architecture in which each word is represented as a bag of character N-grams. This is one more extra layer of word representation where vectors enjoy the additional shared knowledge of N-Grams. For parameter specification, we use the default suggested settings for their bucket length, N-Gram sizes and update rates.
- Ours (S2V). After creation of Dictionary and Thesaurus collections, we perform Bidirectional Constraint Propagations to extend our constraint sets. To achieve an efficient hyperparameter optimization for $\lambda_{\mathbf{D}}$ and $\lambda_{\mathbf{T}}$, we apply a guided parameter sweep (grid search) algorithm and use the same hyperparameters during different experiments.

5. http://www.thesaurus.com

3.1 Quantitative Results

To measure how these word pairs are affected when we apply our model, we fit a Kernel Density Estimate to the cosine distances of pairs for symmetry, expert, association and depict the results in Figure 2. Satisfying our expectations, learning with our model causes all densities to undergo a mean shift and yield a higher average inner product. The density shift is relatively larger in expert pairs compared to symmetry and associations, suggesting that the expert agreement has the strongest impact on our constraints. Furthermore, observe that original densities for these word pairs are right (positive) skewed. This is logical when there is no prior knowledge available for semantics, most of the pairs are tend to have low cosine similarity. Learning with our model corrects the inherent skew, and yields a Gaussian-peaked concentration for inner products.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2942021, IEEE Transactions on Knowledge and Data Engineering



(a) No Propagation (b) Constraint Propagations

Fig. 3: Log Likelihood landscapes for $\{\lambda_{\mathbf{T}}, \lambda_{\mathbf{D}}\}$ where a) No Propagations are applied b) Bidirectional Constraint Propagations are applied. In both cases, the optimization landscape has a global minima. Constraint Propagations increase the smoothness of the optimization landscape and causes contour edges to yield smoother transitions. This smooth and well-behaving property suggests that optimization for such an objective landscape is easier.

Some random pairs obtained from our bidirectional propagation step are shown in Table 2. Symmetry and expert agreement pairs highlight strong semantic relevancies. As there seems to be a low deviation in the conveyed meaning for some of these pairs, arguably, these can even be used as meaning-preserving substitutes for training a lexical substitution system (e.g. "examination-test", "forbidden-taboo"). Unlike symmetry and expert pairs, association pairs instead depict gripping cases. We observe that associations are effective at highlighting particular lost dimensions of word meanings. For instance, some generated pairs like "science-aesthetic" which captures a usually omitted dimension of the word science, suggests that "science" is not only functional but also contains an aesthetics regard. Associations can also generate real concepts that word embedding model does not explicitly address. "international-alphabet" pair in Table 2 is such an example depicting how simple associations on word pairs can also point to phrasal concepts such as the *Phonetic Alphabet*. Note that such phrasal concepts can only be included in the vocabulary after a stage of word to phrase modeling [25]. Associations in our model in some sense implicitly form these links to further tune word level embeddings and circumvents phrase-word conversation problem. Compared to symmetry and expert relations, associations introduce potentially valuable semantics that we do not observe in the corpus and corrects some amount of lost information due to the imprecise modelling.

TABLE 2: Example word pairs from propagation sets.

Symmetry	Expert	Association
coal-fuel	forbidden-taboo	time-atomic
examination-test	hit-serve	abroad-disperse
gold-jewellery	crack-open	natural-harmony
carry-serve	microscobic-small	society-tandem
medicine-surgery	existence-produce	art-witchcraft
address-addressed	disrupt-prevent	black-gathering
break-disrupt	cave-hill	science-aesthetic
short-summary	pond-water	dignity-quality
box-wagon	fall-shower	international-alphabet
college-institution	cache-hidden	language-grammatical

3.2 Model Selection

For model selection purposes, we analyze the likelihood of multiple instances of our model. We form a large validation set containing millions of words and then evaluate the predictive likelihood of each model instance on this set. Since exact computation is not feasible, similarly to stochastic computations in [26], we compute a stochastic likelihood with sampling few context words around the target word and randomly perform multiple repetitions.

Figure 3a and Figure 3b depicts the likelihood LL contours over the $\{\lambda_{\mathbf{T}}, \lambda_{\mathbf{D}}\}$ grid without and with Constraint Propagations. We observe landscapes exhibit a unique maximum on both settings. The figure shows that an advantage of Constraint Propagations is increasing the smoothness of the optimization landscape in which contour edges yield smoother transitions. This means for any optimization algorithm, it is faster to discover a better maximum when new constraints are formed using these propagation rules.

In particular, the slope of the contours also shows that heavy constraints of Thesaurus are much more informative compared to the ones obtained from Dictionaries. The orientation of the contours suggests that there is a linear relationship between $\lambda_{\mathbf{T}}$ and $\lambda_{\mathbf{D}}$, which can be interpreted as the relative weighting of these sources. This is a remarkable observation which can drive efficient data-collection for learning word embeddings. Grounding on our embedding model for learning semantics, we observe that one Thesaurus is able to minimize the Log Likelihood similarly to ten dictionaries.

3.3 Embedding Stability

In this section, we want to measure the stabilization effects of using our embedding technique. To be able to capture long-range dependencies of word cooccurrences, large window sizes have to be used [15]. Nevertheless, experimental evidence [27] shows that embeddings obtained from such training conditions are shown to be highly unstable. To understand the behavior of the models, we simply train multiple randomly initialized embeddings and check how the nearest neighbors of the query words are subject to variations. We first train multiple random embeddings and store the nearest neighbors of query words using cosine similarity. Then, similarly to [28] we use a stability measure based on the Jaccard Index for comparing the similarity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2942021, IEEE Transactions on Knowledge and Data Engineering

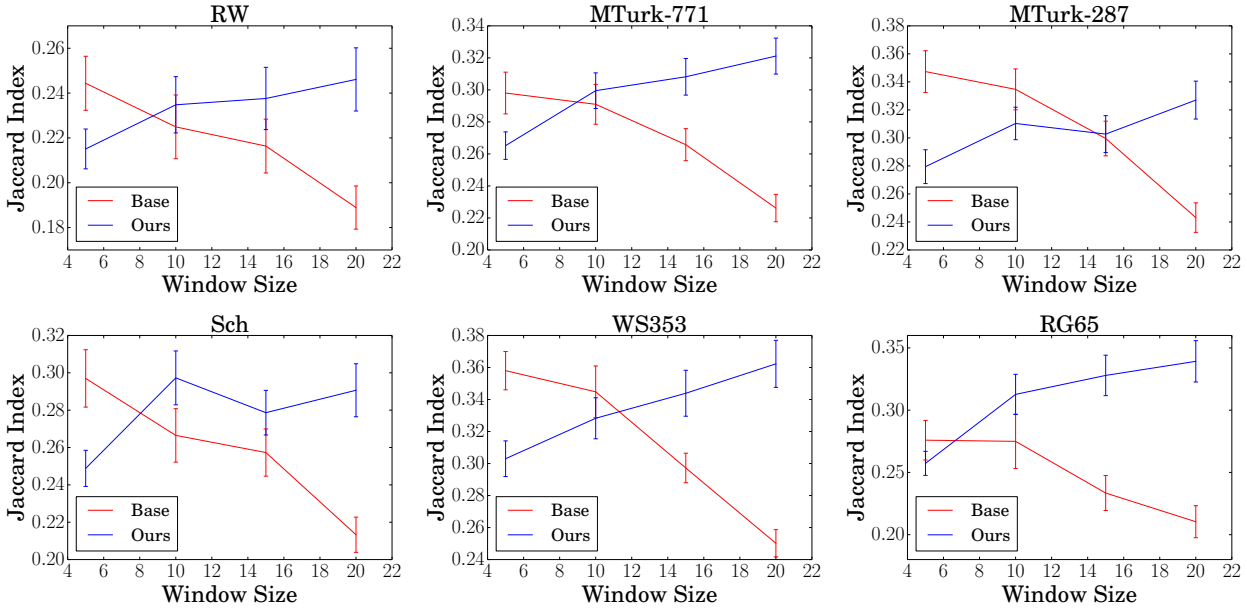


Fig. 4: Jaccard Stability Index on different query inventories. Base means having no semantic constraints. Despite the traditional approach, the stability does not detoriate with our approach. The embeddings yields to be highly stable especially for the large window sizes.

TABLE 3: Word Similarity performances of embeddings trained on first 50 Million words, and the Full version of Wikipedia 2017. We report Spearman's Correlation Coefficient measure.

	Wiki50M					Wiki50M+				
	SG	CBoW	D2V	FT	Ours	SG	CBoW	D2V	FT	Ours
MC-30	69.9	64.2	**74.5**	74.1	72.0	76.7	72.9	75.3	**78.5**	77.6
MEN	69.5	65.3	71.1	70.4	**72.1**	71.7	66.7	72.0	72.1	**72.3**
MTurk-287	65.4	65.5	66.6	66.0	**68.5**	65.6	65.3	66.6	67.6	**68.0**
MTurk-771	61.4	56.3	65.6	59.9	**70.2**	64.7	60.9	67.6	64.5	**70.9**
RG-65	70.0	67.5	76.8	69.9	**80.6**	80.3	75.3	82.0	78.0	**83.9**
RW	40.9	31.2	43.4	44.9	**49.2**	46.9	40.4	47.9	49.1	**50.9**
SimVerb	20.8	15.5	29.8	19.7	**43.5**	30.0	23.4	35.7	28.6	**47.1**
WS	69.9	62.7	**74.2**	67.2	71.6	72.2	64.1	**73.6**	68.3	72.7
WSR	64.6	55.7	**67.9**	62.9	61.5	65.6	56.3	**67.3**	63.3	63.5
WSS	75.6	68.6	77.8	72.4	**77.9**	77.8	71.1	78.0	75.2	**78.9**
YP-130	39.8	32.5	56.0	46.3	**67.5**	54.7	47.2	58.7	59.1	**67.6**
W. Average	46.9	41.1	51.7	47.4	**57.9**	52.4	46.5	54.9	52.3	**59.7**
	WikiFull					WikiFull+				
	SG	CBoW	D2V	FT	Ours	SG	CBoW	D2V	FT	Ours
MC-30	78.6	66.4	78.5	73.4	**79.6**	**79.3**	76.0	78.2	77.9	**79.3**
MEN	71.3	67.1	72.6	71.5	**74.5**	72.5	68.7	72.0	74.4	**75.3**
MTurk-287	65.4	65.5	64.8	**67.2**	66.5	64.0	63.9	64.2	**69.1**	66.7
MTurk-771	61.7	57.2	66.2	60.1	**73.2**	64.7	60.1	67.5	68.1	**74.2**
RG-65	74.6	70.9	79.2	69.7	**83.6**	79.1	77.5	81.2	79.9	**85.6**
RW	43.1	37.4	45.8	46.5	**53.1**	47.6	43.5	49.2	**54.5**	53.6
SimVerb	20.9	15.7	29.6	19.0	**43.6**	26.7	23.0	33.7	35.7	**46.8**
WS	70.3	62.7	72.9	67.2	**74.2**	71.0	63.1	**73.2**	71.4	**73.2**
WSR	63.9	55.8	66.2	62.1	**66.3**	64.9	56.3	65.4	**65.7**	64.6
WSS	76.4	69.6	78.2	72.1	**80.8**	76.9	70.2	78.3	76.5	**79.9**
YP-130	33.2	24.3	50.7	46.5	**68.1**	47.5	40.2	57.3	63.3	**69.2**
W. Average	47.9	42.8	52.4	47.8	**59.8**	51.5	47.4	54.4	56.9	**61.2**

and diversity of sample sets. The index is defined as the size of the intersection divided by the size of the union of the sample sets: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ where A and B are embedding sets for a set of word queries. For query sets, we use word similarity datasets as well as the recently proposed Sch dataset of [29], which is calibrated well according to word frequencies, and also considers parts-of-speech and abstractness of words into account.

Figure 4 depicts the mean and the variance of the Jaccard Index for each query inventory. The stability significantly deteriorates on large window sizes with the typical embedding learning approach. The mean deterioration trend is mostly linear for RW and Sch datasets, and variances are comparably similar. Our approach does not deteriorate on large window sizes, instead yields increased stability. The stability results strongly suggest that learning the embeddings do possess high degrees of freedom in the optimization, maybe even more than necessary, carrying the risk of forming random neighbors for words in each training instance. Our constraint pairs serve as a stabilizer for avoiding these solutions.

In Figure 5, we project the word vectors to 2D space using TSNE dimensionality reduction [30] and show how the close proximity of a randomly sampled word (”feasible” in this case) changes. The first row shows the results of the SG model, and the second row shows vectors obtained with our model. Each column shows the neighborhood of the word and is obtained from a random training instance. The circle radius' indicates how many times a neighboring word appears in all four training instances. We observe more consistent neighbors when training includes our constraints, and the number of stable neighbors is higher due to our constraints. Embeddings trained with our semantic constraints favour stabilized solutions for all query sets compared to the original embedding problem and might be also utilized when the task of interest asks for large window dependency modelling.

3.4 Word Similarity Measurements

We report the word similarity results of all trained embeddings both on the first 50M words that represent the scarce data regime, and also on the full version of the Wikipedia corpora representing the big data regime. Since our method uses information from multiple lexical sources, we would like to perform a fair comparison against all other baselines. For this purpose, we also concatenate the collected dictionary definitions and Thesaurus to the training data so that other methods can also benefit from the information of these extra sources. Wiki 50M denotes the raw training corpus whereas Wiki 50M+ is the corpus with the aforementioned dictionary and Thesaurus concatenations. To increase the confidence of the experiments, we repeat each experiment with different seeds and report the averages.

We test our embeddings on a large set of test collections. As a standard extrinsic benchmark of [29], we compute the Spearman Correlation Coefficient of cosine distances of word pairs to measure how much embeddings can predict the expert annotated similarities. Since dataset overall performance might also be interest, we also report the weighted average result by weighting each dataset with its query inventory size. Our test suite contains the following datasets: MC-30 [31], MEN [32], MTurk-287 [33], MTurk-771 [34], RG65 [35], RW [36], SimVerb-3500 [37], WordSim-353 [38] and YP-130 [39].

Table 3 shows that for models trained on Wiki50M corpus, the gain of our approach over FastText reaches 10.5%, and Dict2Vec by 6.2% on dataset average. On a dataset basis, our method obtains highest gains for SimVerb and YP-130 datasets. For models trained on the concatenated Wiki50M+ corpus, other methods yield an average of 4.75% increased performance, whereas our model obtains 1.2% extra on the Wiki50M corpus. It turns out that concatenation of dictionary and Thesaurus pairs from the semantic sources as training input can benefit all models only for a few percents. The contribution of our model is the largest for the RW and Simverb datasets. Here, SimVerb contains many pairs for the syntactic and semantic similarities of verb meanings. RW dataset contains query pairs that are observed only a few times in the corpus. We understand that leveraging pairwise constraints helps most for learning the verb meanings, and also for out of vocabulary queries. Our observations are similar when training on the full version, except that a few percents extra performance is obtained, with FastText gaining the most from the concatenation routine.

Our approach leverages information from different sources to learn the embeddings. On the surface, our method exhibits similarities to Dict2Vec, which also leverages information from lexical dictionaries. The reader must note that there are few key distinctions in which Sem2Vec deviates from such dictionary learning work. First, our work focuses on combining multiple semantic informativeness into the same learning framework whereas Dict2Vec aims at introducing the dictionaries to word embedding learning optimally. In this sense, our purpose is to specialize the vectors to semantics whereas Dict2Vec aims to obtain vectors that have a balance between syntactic and semantic accuracy. Secondly, Dict2Vec requires a pretraining stage where they assume that initial word vectors are already trained and available. This assumption is questionable since the quality of the final embeddings heavily relies on how well the initial pretrained embeddings are. A variation of this multi-step approach is proposed in [40] which embeddings are further refined offline by a retrofitting step. We deviate from such multi-step approaches that advocate pretraining or postprocessing the word vectors. With our learning framework, we avoid both approaches and train the whole embedding pipeline in one-shot.

Word similarity results highlight that other word embedding architectures can also enjoy the additional performance when they are exposed to the dictionary and Thesaurus content. Here we ask the question of whether treating these semantic sources as samples are profitable. Is it sufficient to apply sample duplications or extending the formulation with our constraints is a necessity?

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2942021, IEEE Transactions on Knowledge and Data Engineering

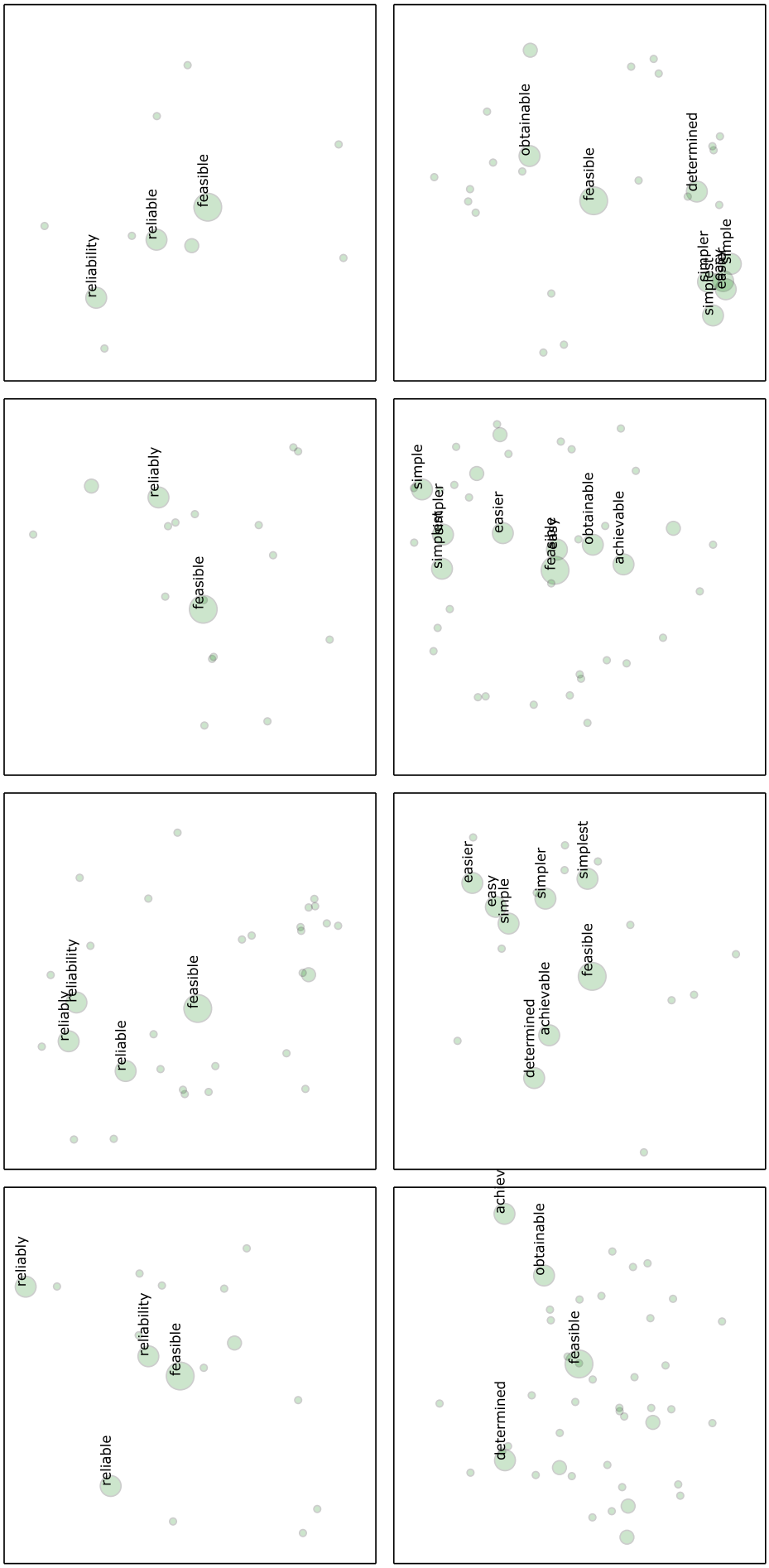


Fig. 5: Neighbouring proximity of the word *feasible*. First row is the SG, and second row is our approach. Each figure in the columns, is obtained from a random training instance and shows the resulting neighbourhood of the word. The circle radius' indicates of how many times that word appears in all four instances. Our constraints preserves many of the neighbouring words across random training instances, and outputs much more stable embeddings.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2942021, IEEE Transactions on Knowledge and Data Engineering

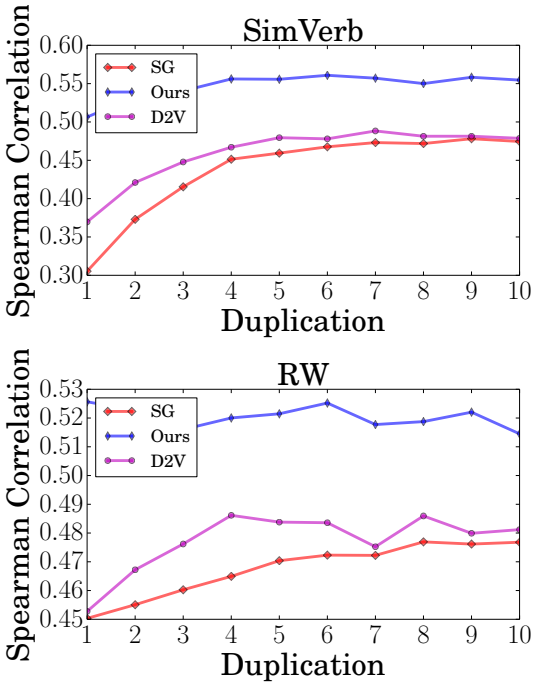


Fig. 6: Word Similarity performances when semantic sources are concatenated multiple times to the training corpus. The gain for other embedding architectures quickly saturates.

We answer this question in Figure 6 where we simply extract all pairs from dictionary and Thesaurus sources and concatenate them multiple times to the available corpus. We observe that the first few duplications raise the performance greatly, but gain saturates around 10 duplications where no additional benefit is observed. In contrast, duplications serve as random noise fluctuations for our approach. We conclude that treating these extra sources as sample duplications are an alternative approach to embed semantic knowledge to the learning problem while introducing little extra training time. However, the performance gain is far away from optimal.

So far we assumed that the model has access to the highest level semantic source during training. Under some conditions, this assumption might be too optimistic since for many languages these semantic sources might not be either available or accessible. We name this condition as *economical scenario* for the word embedding learning. In Figure 7, we demonstrate how word similarity performance varies when we are only left with a dictionary source and lose access to the Thesaurus content. On all datasets, losing access to the Thesaurus harms the performance. We observe significant performance losses on the RW and SimVerb datasets. Comparably, the drop is less significant for easy datasets containing very frequent words such as RG-65 and WSS. This suggests that learning word similarities can be done using only lexical dictionaries, given that test sets query relatively easy word pairs. On the other hand, if test sets contain pairs that are rare, exploiting a higher level of semantic source appears to be indispensable.

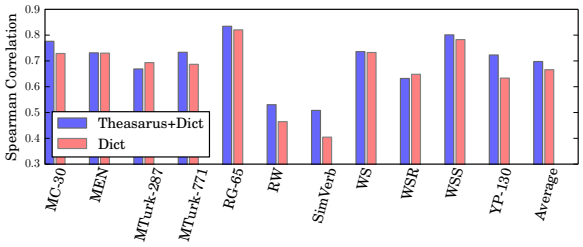


Fig. 7: Word similarity performances when high level semantic source is unavailable. Significant performance losses are observed on the challenging RW and SimVerb datasets. Losses are less significant for datasets with very frequent words.

3.5 Document Text Classification

We follow the standard text classification evaluation similarly to the [41] and evaluate our vectors on the Agnews [42] dataset which was compiled from 2000 different text sources providing news articles. It has documents of 4 classes, randomly split into 120k training and 7k test documents. Compared to Agnews, Dbpedia [43] dataset is a larger corpus and has a split of 560k training documents and 70k test documents from 14 classes.

For all baseline models, we train the vectors first on the unsupervised corpora. We then construct document representations by computing the average word vector of each document. This document embedding is then plugged as an input to a standard Multilayer Perceptron (MLP) with a single hidden layer with ReLu activation functions on the neurons. The network is then trained with the stochastic ADAM optimizer [44] until convergence with an adaptively decaying learning rate. To yield a fully fair comparison of different word embedding vectors, we fix the embedding weights and do not allow word vector layer to change during the classification so that we can accurately quantify the performance gain from each vector set.

As the performance measure, we report first the standard F1-Score classification score:

$$F_1 = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6}$$

which can be interpreted as the harmonic mean of the precision and recall. As, the prediction on the both Dbpedia and Agnews datasets is a multi-class classification problem, and there is almost no class imbalance in the test datasets, we use a macro averaging to compute a single F1 score. We also report the Multi-class Receiver Operating Characteristic (ROC) curves and Area under Curve (AUC) for each class in the training sets.

Confusion matrices in Figure 8 show that objects from Business and World classes in Agnews are relatively harder where many false positives are encountered. The F1 scores for both datasets are reported in the Figure 9. Addressing the classification problem with our word vectors obtains much higher performance compared to the baseline methods especially when the amount of data is relatively limited. The ROC curves of Agnews dataset are reported in Figure 10.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2942021, IEEE Transactions on Knowledge and Data Engineering

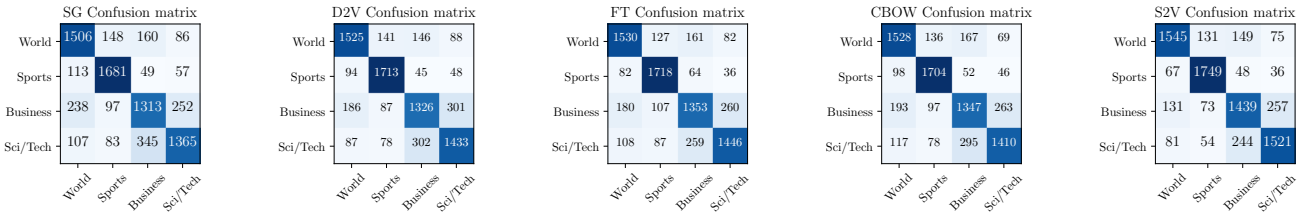


Fig. 8: Agnews Confusion Matrices. Matrix rows indicate true classes of the objects, and matrix columns indicate predicted classes.

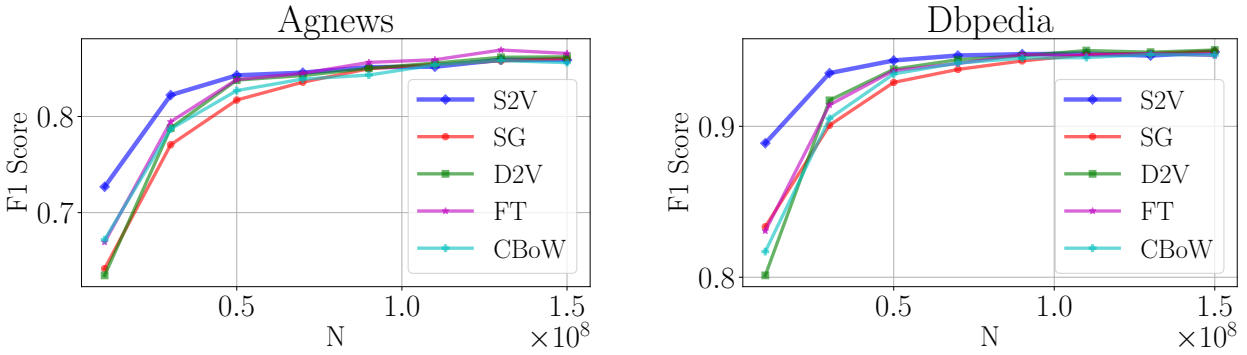
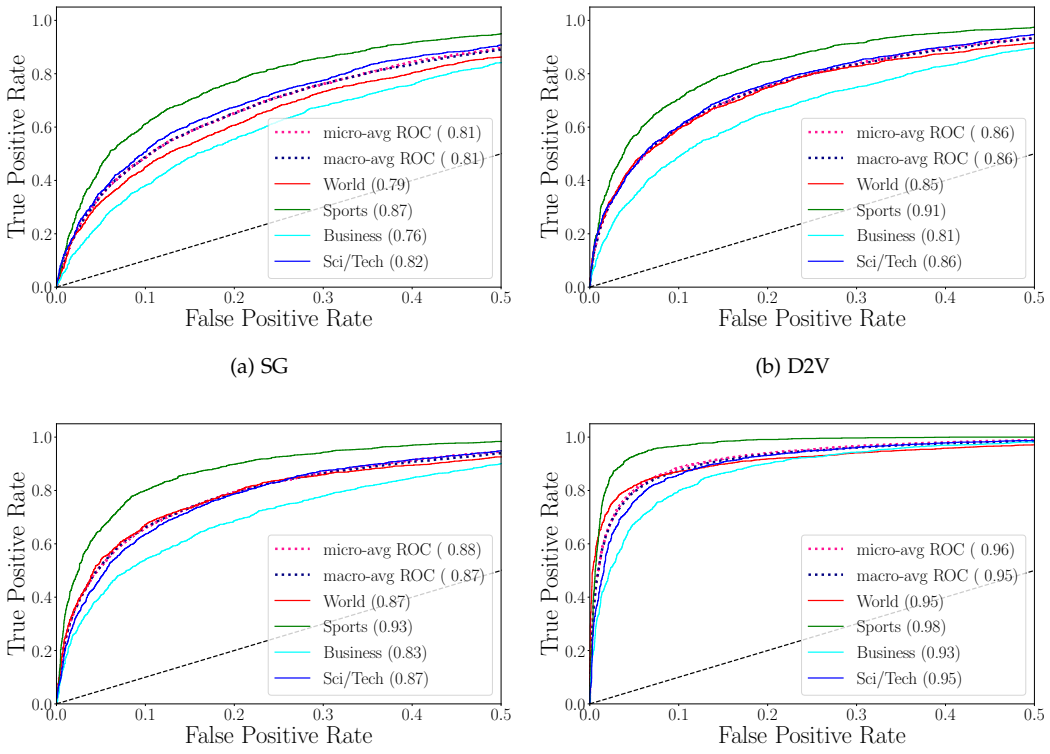


Fig. 9: F1 scores of the methods on the Agnews and Dbpedia classification datasets.

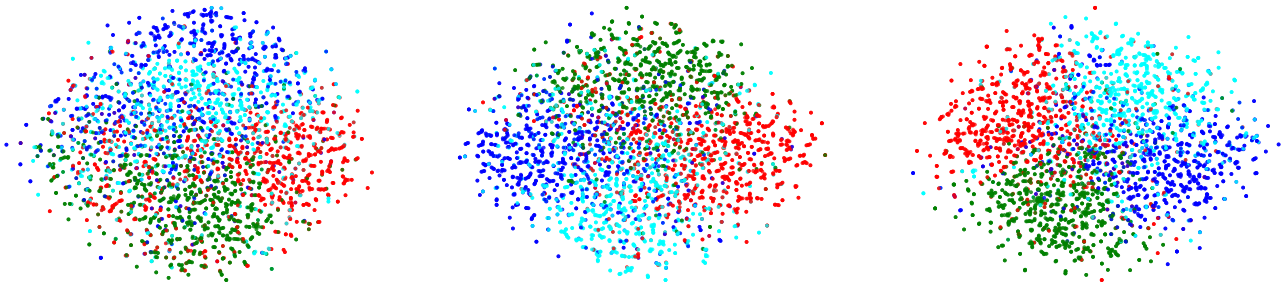


(a) SG (b) D2V

(c) FT (d) S2V

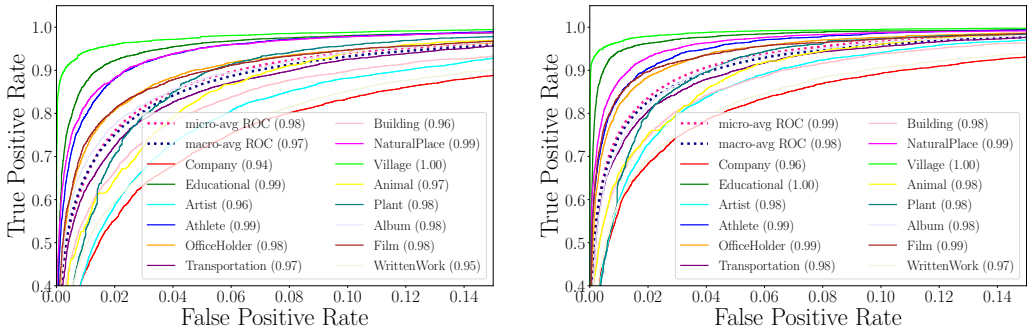
Fig. 10: Multi-class Receiver Operating Characteristic (ROC) curves for each method in the Agnews dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2942021, IEEE Transactions on Knowledge and Data Engineering



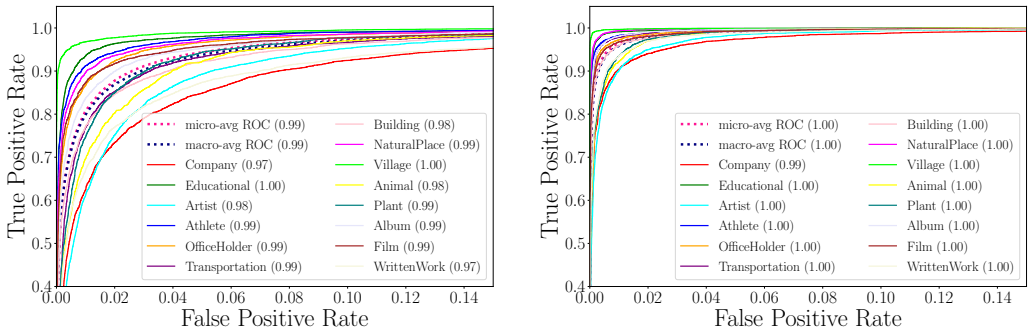
(a) SG-Agnews (b) FT-Agnews (c) S2V-Agnews

Fig. 11: For visualization purposes, we project the document embeddings to two dimensions using T-SNE dimensionality reduction. Each color corresponds to a particular document class in Agnews dataset. a) documents constructed with Skip Gram vectors. b) documents constructed with FT vectors c) documents constructed with S2V vectors. Observe that not only intra-class documents are grouped coherently with our vectors, but also inter-class distances are relatively higher.



(a) SG

(b) D2V



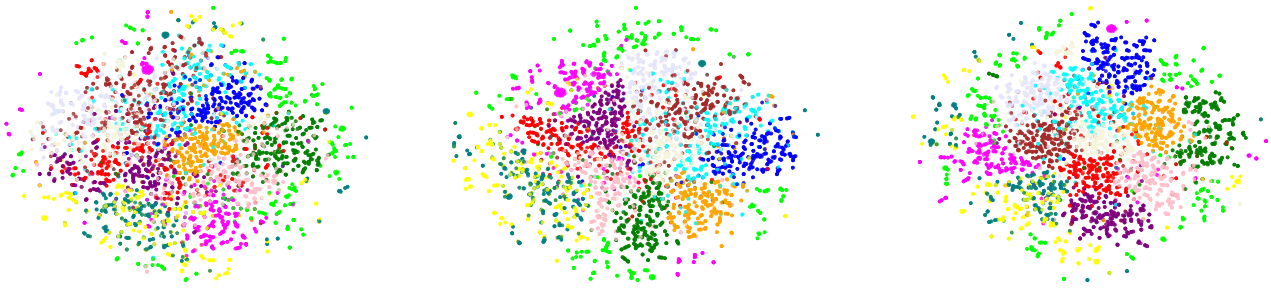
(c) FT

(d) S2V

Fig. 12: Multi-class Receiver Operating Characteristic (ROC) curves for each method in the Dbpedia dataset.

For visualization purposes, we also project the document representations constructed with different word vectors to 2D. In Figure 11, these low dimensional projections confirms the ROC curve in Figure 10 that objects of the Business class (teal colored) is the most difficult to classify. Here, better representation for the classification requires intra-class documents to be close to each other, and inter-class distances to be higher.

We draw similar conclusions for Dbpedia dataset with ROC curves in Figure 12. S2V clearly achieves higher true positive rates. The lower dimensional plots in Figure 13 depict that our semantic constraints become indispensable when the number of classes in the problem increase. Note that the clutter is much prominent in Figure 13a-b, especially in Company class (red points), the complexity of the classification task is higher, showing the necessity of incorporating semantic constraints. Despite the fact that these vectors were not specifically trained for a classification setting, we are able to achieve promising results with them.



(a) SG-Dbpedia (b) FT-Dbpedia (c) S2V-Dbpedia

Fig. 13: For visualization purposes, we project the document embeddings to two dimensions using T-SNE dimensionality reduction. Each color corresponds to a particular document class in Dbpedia dataset. a) documents constructed with Skip Gram vectors. b) documents constructed with FT vectors c) documents constructed with S2V vectors. Observe that not only intra-class documents are grouped coherently with our vectors, but also inter-class distances are relatively higher.

4 Conclusion

In this work, we proposed a novel embedding framework to learn word vectors specializing in semantics. Our word embedding pipeline integrated various levels of semantic sources into one unified formulation by treating highly informative lexical sources as heavy constraints, and lexical dictionaries as light constraints to learning. We then utilized the domain knowledge inherent in the lexical sources to further refine our constraint sets by bidirectional constraint propagations, yielding a smoother and better behaving objective function.

Our semantically constrained embedding formulation is notably more stable than the typical word embeddings, especially for training settings on the large window sizes. This is an attractive property, and closes the gap between performance and stability in the field of embeddings. We also empirically evaluated how much gain our model provides for word similarity measurements when trained under scarce and big training data. The practical contribution of our model on the word similarity test suite of eleven datasets is measured, showing significant improvements over the state of the art techniques. Our findings on incorporating semantic knowledge are also supported by the limitations of sample duplication, further supplemented the necessity of a constraint based formulation. Lastly, worst-case economic scenarios in which a semantic source is unavailable is investigated and performance losses are discussed, posing the limitations of our approach.

Perhaps notable merit of our formulation is that it integrates semantic knowledge to the features but follows the conventional word embedding pipeline where training does not require any human in the loop. This is an important remark to obtain vectors in a manageable time since most of the embedding architectures require a human in the loop, which in return significantly slows down the training procedure. Following our experimental evaluation, we conclude that in contrast to our method, state of the art vectors do not have a strong guarantee to learn semantic relevancies especially when the amount of training data is scarce for the given language. Sem2Vec not only provides stability of the end results but also maintains the time-efficiency of the embedding training since the complexity does not increase greatly with the number of constraints.

Acknowledgments

The authors acknowledge funding by the Dutch Organization for Scientific Research (NWO; grant 612.001.301). We would also like to thank Alexander Mey for proofreading the manuscript.

References

[1] C. Speier, J. S. Valacich, and I. Vessey, "The influence of task interruption on individual decision making: An information overload perspective," 1999.

[2] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *CoRR*, vol. abs/1003.1141, 2010.

[3] J. Firth, "A synopsis of linguistic theory 1930-1955," *Studies in linguistic analysis*, pp. 1–32, 1957.

[4] Z. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[5] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, 2003.

[6] M. Baroni, G. Dinu, and 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*, June 2014, pp. 238–247.

[7] D. Kiela, F. Hill, and S. Clark, "Specializing word embeddings for similarity or relatedness." in *EMNLP*, L. Mrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds. The Association for Computational Linguistics, 2015, pp. 2044–2048.

[8] I.-E. Parasca, A. L. Rauter, J. Roper, A. Rusinov, G. Bouchard, S. Riedel, and P. Stenetorp, "Defining words with words: Beyond the distributional hypothesis," in *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Association for Computational Linguistics, 2016, pp. 122–126.

[9] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, no. 4, pp. 327–352, 1977.

[10] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph and text jointly embedding," in *The 2014 Conference on Empirical Methods on Natural Language Processing*, October 2014.

[11] M. Yu and M. Dredze, "Improving lexical embeddings with semantic knowledge." in *ACL (2)*, 2014, pp. 545–550.

[12] J. Tissier, C. Gravier, and A. Habrard, "Dict2vec : Learning word embeddings using lexical dictionaries," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017, pp. 254–263.

[13] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer, "Problems with evaluation of word embeddings using word similarity tasks," *CoRR*, vol. abs/1605.02276, 2016.

[14] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013.

[15] A. B. Dieng, C. Wang, J. Gao, and J. W. Paisley, "Topicrnn: A recurrent neural network with long-range semantic dependency," *CoRR*, vol. abs/1611.01702, 2016.

[16] Y. Jiang, W. Bai, X. Zhang, and J. Hu, "Wikipedia-based information content and semantic similarity computation," *Inf. Process. Manage.*, vol. 53, no. 1, pp. 248–265, Jan. 2017.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[18] U. Topkara, M. Topkara, and M. J. Atallah, "The hiding virtues of ambiguity: Quantifiably resilient watermarking of natural language text through synonym substitutions," in *Proceedings of the 8th Workshop on Multimedia and Security*. New York, NY, USA: ACM, 2006, pp. 164–174.

[19] V. Nebot and R. Berlanga, "Finding association rules in semantic web data," *Know.-Based Syst.*, vol. 25, no. 1, pp. 51–62, Feb. 2012.

[20] D. Mimno and L. Thompson, "The strange geometry of skip-gram with negative sampling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 2873–2878.

[21] T. Kekeç, L. van der Maaten, and D. M. J. Tax, "Pawe: Polysemy aware word embeddings," in *Proceedings of the 2Nd International Conference on Information System and Data Mining*, ser. ICISDM '18. New York, NY, USA: ACM, 2018, pp. 7–13.

[22] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *CoRR*, vol. abs/1402.3722, 2014.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[24] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[25] M. Yu and M. Dredze, "Learning composition models for phrase embeddings," *TACL*, vol. 3, pp. 227–242, 2015.

[26] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2177–2185.

[27] Q. Luo, W. Xu, and J. Guo, "A study on the cbow model's overfitting and stability," in *Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning*, ser. Web-KR '14. ACM, 2014, pp. 9–12.

[28] M. Antoniak and D. Mimno, "Evaluating the stability of embedding-based word similarities," in *Transactions of the Association for Computational Linguistics*, 2017.

[29] T. Schnabel, I. Labutov, D. M. Mimno, and T. Joachims, "Evaluation methods for unsupervised word embeddings." in *EMNLP*, 2015, pp. 298–307.

[30] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[31] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and Cognitive Processes*, vol. 6, no. 1, pp. 1–28, 1991.

[32] E. Bruni, N. K. Tran, and M. Baroni, "Multimodal distributional semantics," *J. Artif. Int. Res.*, vol. 49, no. 1, pp. 1–47, Jan. 2014.

[33] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: Computing word relatedness using temporal semantic analysis," in *Proceedings of the 20th International World Wide Web Conference*, Hyderabad, India, March 2011, pp. 337–346.

[34] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, "Large-scale learning of word relatedness with constraints," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 1406–1414.

[35] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Commun. ACM*, vol. 8, no. 10, pp. 627–633, Oct. 1965.

[36] T. Luong, R. Socher, and C. D. Manning, "Better word representations with recursive neural networks for morphology," in *In Proceedings of the Thirteenth Annual Conference on Natural Language Learning. Tomas Mikolov, Wen-tau*, 2013.

[37] D. Gerz, I. Vulić, F. Hill, R. Reichart, and A. Korhonen, "SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity," in *EMNLP*, 2016.

[38] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," in *Proceedings of the 10th International Conference on World Wide Web*, ser. WWW '01. New York, NY, USA: ACM, 2001, pp. 406–414.

[39] D. Yang and D. M. W. Powers, "Verb similarity on the taxonomy of wordnet," in *In the 3rd International WordNet Conference (GWC-06), Jeju Island, Korea*, 2006.

[40] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith, "Retrofitting word vectors to semantic lexicons," in *Proceedings of NAACL*, 2015.

[41] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016, cite arxiv:1607.01759.

[42] A. Gulli, "The anatomy of a news search engine," in *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, ser. WWW '05. New York, NY, USA: ACM, 2005, pp. 880–881.

[43] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The Semantic Web*, K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, 2014.

Taygun Kekeç received his BSc degree in Computer Science and Engineering from Yildiz Technical University, Istanbul in 2010. He received his MSc degree in Mechatronics Engineering from Sabanci University, Istanbul, in 2013 with specialization of robotics and computer vision. His master thesis is titled 'Developing Object Detection, Tracking and Image Mosaicing Algorithms for Visual Surveillance'. He is currently a PhD student under supervision of David M.J. Tax at Pattern Recognition and Bioinformatics group at Delft University of Technology. In 2017, he was a visiting researcher in Information Sciences Department of Cornell University, Ithaca. His current research interests cover a wide spectrum of machine learning formulations with particular applications to computer vision and natural language processing problems including approximate probabilistic inference, object segmentation and tracking, word vector embeddings and unsupervised text analysis.

David M.J. Tax studied physics at the University of Nijmegen, The Netherlands in 1996, and received Master degree with the thesis "Learning of structure by Many-take-all Neural Networks". After that he had his PhD at the Delft University of Technology in the Pattern Recognition group, under the supervision of R.P.W. Duin. In 2001 he promoted with the thesis 'One-class classification'. After working for two years as a Marie Curie Fellow in the Intelligent Data Analysis group in Berlin, at present he is assistant professor in the Pattern Recognition Laboratory at the Delft university of Technology. His main research interest is in the learning and development of detection algorithms and (one-class) classifiers that optimize alternative performance criteria like ordering criteria using the Area under the ROC curve or a Precision-Recall graph. Furthermore, the problems concerning the representation of data, multiple instance learning, simple and elegant classifiers and the fair evaluation of methods have focus. Good representation, and suitable performance measures should not only lead to good classifiers, but should also help the user in understanding the problems that he is solving.