

Ultradense Word Embeddings by Orthogonal Transformation

Sascha Rothe and Sebastian Ebert and Hinrich Schütze

Center for Information and Language Processing

LMU Munich, Germany

{sascha|ebert}@cis.lmu.de

Abstract

Embeddings are *generic* representations that are useful for many NLP tasks. In this paper, we introduce DENSIFIER, a method that learns an *orthogonal transformation* of the embedding space that focuses the information relevant for a task in an *ultradense subspace* of a dimensionality that is smaller by a factor of 100 than the original space. We show that ultradense embeddings generated by DENSIFIER reach state of the art on a lexicon creation task in which words are annotated with three types of lexical information – sentiment, concreteness and frequency. On the SemEval2015 10B sentiment analysis task we show that no information is lost when the ultradense subspace is used, but training is an order of magnitude *more efficient* due to the compactness of the ultradense space.

1 Introduction

Embeddings are useful for many tasks, including word similarity (e.g., Pennington et al. (2014)), named entity recognition (NER) (e.g., Collobert et al. (2011)) and sentiment analysis (e.g., Kim (2014), Kalchbrenner et al. (2014), Severyn and Moschitti (2015)). Embeddings are generic representations, containing different types of information about a word. Statistical models can be trained to make best use of these generic representations for a specific application like NER or sentiment analysis (Ebert et al., 2015).

Our hypothesis in this paper is that the information useful for any given task is contained in an *ultradense subspace* E_u . We propose the new method

DENSIFIER to identify E_u . Given a set of word embeddings, DENSIFIER learns an *orthogonal transformation* of the original space E_o on a task-specific training set. The orthogonality of the transformation can be considered a hard regularizer.

The benefit of the proposed method is that embeddings are most useful if learned on unlabeled corpora and performance-enhanced on a broad array of tasks. This means we should try to keep all information offered by them. Orthogonal transformations “reorder” the space without adding or removing information and preserve the bilinear form, i.e., Euclidean distance and cosine. The transformed embeddings concentrate all information relevant for the task in E_u .

The benefits of E_u compared to E_o are (i) *high-quality* and (ii) *efficient* representations. (i) DENSIFIER moves non-task-related information outside of E_u , i.e., into the orthogonal complement of E_u . As a result, E_u provides *higher-quality representations* for the task than E_o ; e.g., noise that could result in overfitting is reduced in E_u compared to E_o . (ii) E_u has a *dimensionality smaller by a factor of 100* in our experiments. As a result, training statistical models on these embeddings is much faster. These models also have many fewer parameters, thus again helping to prevent overfitting, especially for complex, deep neural networks.

We show the benefits of ultradense representations in two text polarity classification tasks (SemEval2015 Task 10B, Czech movie reviews).

In the most extreme form, ultradense representations – i.e., E_u – have a single dimension. We exploit this for creating lexicons in which words are

annotated with lexical information, e.g., with sentiment. Specifically, we create high-coverage lexicons with up to 3 million words (i) for three lexical properties: for *sentiment*, *concreteness* and *frequency*; (ii) for five languages: *Czech*, *English*, *French*, *German* and *Spanish*; (iii) for two domains, *Twitter* and *News*, in a domain adaptation setup.

The main advantages of this method of lexicon creation are: (i) We need a training lexicon of only a few hundred words, thus making our method effective for new domains and languages and requiring only a minimal manual annotation effort. (ii) The method is applicable to any set of embeddings, including phrase and sentence embeddings. Assuming the availability of a small hand-labeled lexicon, DENSIFIER automatically creates a domain dependent lexicon based on a set of embeddings learned on a large corpus of the domain. (iii) While the input lexicon is *discrete* – e.g., positive (+1) and negative (-1) sentiment – the output lexicon is *continuous* and this more fine-grained assessment is potentially more informative than a simple binary distinction.

We show that lexicons created by DENSIFIER beat the state of the art on SemEval2015 Task 10E (determining association strength).

One of our goals is to make embeddings more interpretable. The work on sentiment, concreteness and frequency we present in this paper is a first step towards a *general decomposition of embedding spaces into meaningful, dense subspaces*. This would lead to cleaner and more easily interpretable representations – as well as representations that are more effective and efficient.

2 Model

Let $Q \in \mathbb{R}^{d \times d}$ be an orthogonal matrix that transforms the original word embedding space into a space in which certain types of information are represented by a small number of dimensions. Concretely, we learn Q such that the dimensions $D^s \subset \{1, \dots, d\}$ of the resulting space correspond to a word’s sentiment information and the $\{1, \dots, d\} \setminus D^s$ remaining dimensions correspond to non-sentiment information. Analogously, the sets of dimensions D^c and D^f correspond to a word’s concreteness information and frequency information, respectively. In this paper, we assume that these

properties do not correlate and therefore the ultradense subspaces do not overlap, e.g., $D^s \cap D^c = \emptyset$. However, this might not be true for other settings, e.g., sentiment and semantic information.

If $e_w \in E_o \subset \mathbb{R}^d$ is the original embedding of word w , the transformed representation is Qe_w . We use $*$ as a placeholder for s , c and f and call $d^* = |D^*|$ the dimensionality of the ultradense subspace of $*$. For each ultradense subspace, we create $P^* \in \mathbb{R}^{d^* \times d}$, an identity matrix for the dimensions in $D^* \subset \{1, \dots, d\}$. Thus, the ultradense representation $u_w^* \in E_u \subset \mathbb{R}^{d^*}$ of e_w is defined as:

$$u_w^* := P^* Q e_w \quad (1)$$

2.1 Separating Words of Different Groups

We assume to have a lexicon resource l in which each word w is annotated for a certain information as either $l^*(w) = +1$ (positive, concrete, frequent) or $l^*(w) = -1$ (negative, abstract, infrequent). Let L_\neq^* be a set of word index pairs (v, w) for which $l^*(v) \neq l^*(w)$ holds. We want to maximize:

$$\sum_{(v,w) \in L_\neq^*} \|u_v^* - u_w^*\| \quad (2)$$

Thus, our objective is given by:

$$\operatorname{argmax}_Q \sum_{(v,w) \in L_\neq^*} \|P^* Q (e_w - e_v)\| \quad (3)$$

or, equivalently, by:

$$\operatorname{argmin}_Q \sum_{(v,w) \in L_\neq^*} -\|P^* Q (e_w - e_v)\| \quad (4)$$

subject to Q being an orthogonal matrix.

2.2 Aligning Words of the Same Group

Another goal is to minimize the distance of two words of the same group. Let L_\sim^* be a set of word index pairs (v, w) for which $l^*(v) = l^*(w)$ holds. In contrast to Eq. 3, we now want to minimize each distance. Thus, the objective is given by:

$$\operatorname{argmin}_Q \sum_{(v,w) \in L_\sim^*} \|P^* Q (e_w - e_v)\| \quad (5)$$

subject to Q being an orthogonal matrix.

The intuition behind the two objectives is graphically depicted in Figure 1.

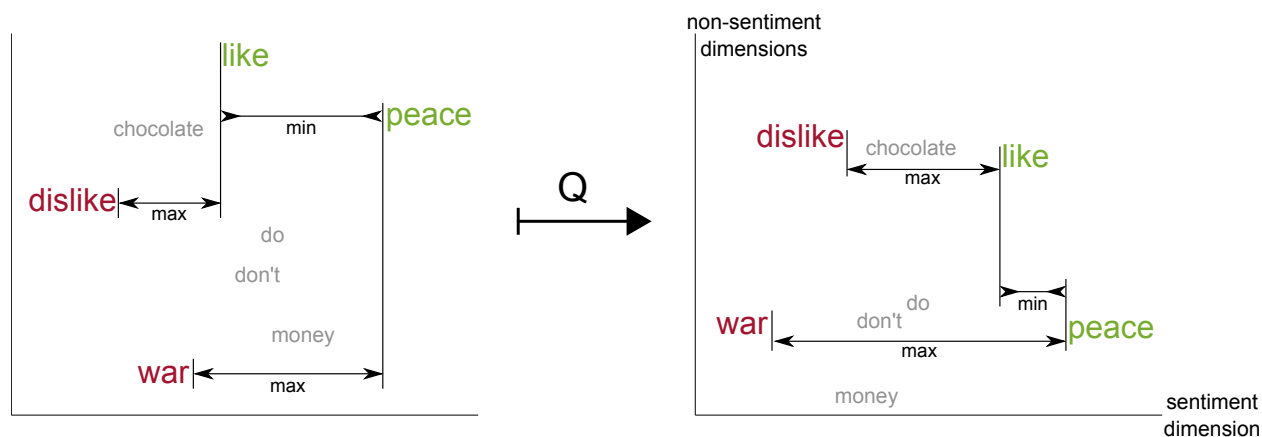


Figure 1: The original word embedding space (left) and the transformed embedding space (right). The training objective for Q is to *minimize* the distances in the sentiment dimension between words of the same group (e.g., positive/green: “like” & “peace”) and to *maximize* the distances between words of different groups (e.g., negative/red & positive/green: “war” & “peace”; not necessarily antonyms).

2.3 Training

We combine the two objectives in Eqs. 3/5 for each subspace, i.e., for sentiment, concreteness and frequency, and weight them with α^* and $1 - \alpha^*$. Hence, there is one hyperparameter α^* for each subspace. We then perform stochastic gradient descent (SGD). Batch size is 100 and starting learning rate is 5, multiplied by .99 in each iteration.

2.4 Orthogonalization

Each step of SGD updates Q . The updated matrix Q' is in general no longer orthogonal. We therefore reorthogonalize Q' in each step based on singular value decomposition:

$$Q' = USV^T$$

where S is a diagonal matrix, and U and V are orthogonal matrices. The matrix

$$Q := UV^T$$

is the nearest orthogonal matrix to Q' in both the 2-norm and the Frobenius norm (Fan and Hoffman, 1955). (Formalizing our regularization directly as projected gradient descent would be desirable. However, gradient descent includes an additive operation and orthogonal matrices are not closed under summation.)

SGD for this problem is sensitive to the learning rate. If the learning rate is too large, a large jump

results and the reorthogonalized matrix Q basically is a random new point in the parameter space. If the learning rate is too small, then learning can take long. We found that our training regime of starting at a high learning rate (5) and multiplying by .99 in each iteration is effective. Typically, the cost initially stays about constant (random jumps in parameter space), then cost steeply declines in a small number of about 50 iterations (sweet spot); the curve flattens after that. Training Q took less than 5 minutes per experiment for all experiments in this paper.

3 Lexicon Creation

For lexicon creation, the input is a set of *embeddings* and a lexicon *resource* l , in which words are annotated for a lexical information such as sentiment, concreteness or frequency. DENSIFIER is then trained to produce a *one-dimensional ultra-dense subspace*. The output is an *output lexicon*. It consists of all words covered by the embedding set, each associated with its one-dimensional ultra-dense subspace representation (which is simply a real number), an indicator of the word’s strength for that information.

The embeddings and lexicon resources used in this paper cover five languages and three domains (Table 1). The Google News embeddings for English¹ and the FrWac embeddings for French² are

¹<https://code.google.com/p/word2vec/>

²<http://fauconnier.github.io/>

			train			test			τ
	#tokens	#types	resource	\cap	#words	resource	\cap	#words	
1	sent CZ web	2.44 3.3	SubLex 1.0	2,492	4,125	SubLex 1.0	319	500	.580
2	sent DE web	1.34 8.0	German PC	10,718	37,901	German PC	573	1,000	.654
3	sent ES web	0.37 3.7	full-strength	824	1,147	full-strength	185	200	.563
4	sent FR web	0.12 1.6	FEEL	7,496	10,979	FEEL	715	1,000	.544
5	sent EN twitter	3.34 5.4	WHM all	12,601	19,329	Trial 10E	198	200	.661
6	sent EN news	3.00 100.0	WHM train	7,633	10,270	WHM val	952	1,000	.622
7	conc EN news	3.00 100.0	BWK	14,361	29,954	BWK	8,694	10,000	.623
8	freq EN news	3.00 100.0	word2vec order	4,000	4,000	word2vec order	1,000	1,000	.361
9	freq FR web	0.12 1.6	word2vec order	4,000	4,000	word2vec order	1,000	1,000	.460

Table 1: Results for lexicon creation. #tokens: size of embedding training corpus (in billion). #types: size of output lexicon (in million). For each resource, we give its size (“#words”) and the size of the intersection of resource and embedding set (“ \cap ”). Kendall’s τ is computed on “ \cap ”.

publicly available. We use word2vec to train 400-dimensional embeddings for English on a 2013 Twitter corpus of size 5.4×10^9 . For Czech, German and Spanish, we train embeddings on web data of sizes 3.3, 8.0 and 3.8×10^9 , respectively. We use the following lexicon resources for sentiment: SubLex 1.0 (Veselovská and Bojar, 2013) for Czech; WHM for English [the combination of MPQA (Wilson et al., 2005), Opinion Lexicon (Hu and Liu, 2004) and NRC Emotion lexicons (Mohammad and Turney, 2013)]; FEEL (Abdaoui et al., 2014) for French; German Polarity Clues (Waltinger, 2010) for German; and the sentiment lexicon of Pérez-Rosas et al. (2012) for Spanish. For concreteness, we use BWK, a lexicon of 40,000 English words (Brybaert et al., 2014). For frequency, we exploit the fact that word2vec stores words in frequency order; thus, the ranking provided by word2vec is our lexicon resource for frequency.

For a resource/embedding-set pair (l, E) , we intersect the vocabulary of l with the top 80,000 words of E to filter out noisy, infrequent words that tend to have low quality embeddings and we do not want them to introduce noise when training the transformation matrix.

For the sentiment and concreteness resources, $l^*(w) \in \{-1, 1\}$ for all words w covered. We create a resource l^f for frequency by setting $l^f(w) = 1$ for the 2000 most frequent words and $l^f(w) = -1$ for words at ranks 20000-22000. 1000 words randomly selected from the 5000 most frequent are the test set.³ We designate three sets of dimen-

sions D^s , D^c and D^f to represent sentiment, concreteness and frequency, respectively, and arbitrarily set (i) $D^c = \{11\}$ for English and $D^c = \emptyset$ for the other languages since we do not have concreteness resources for them, (ii) $D^s = \{1\}$ and (iii) $D^f = \{21\}$. Referring to the lines in Table 1, we then learn six orthogonal transformation matrices Q : for CZ-web (1), DE-web (2), ES-web (3), FR-web (4, 9), EN-twitter (5) and EN-news (6, 7, 8).

4 Evaluation

4.1 Top-Ranked Words

Table 2 shows the top 10 positive/negative words (i.e., most extreme values on dimension D^s) when we apply the transformation to the corpora EN-twitter, EN-news and DE-web and the top 10 concrete/abstract words (i.e., most extreme values on dimension D^c) for EN-news. For EN-twitter (leftmost double column), the selected words look promising: they contain highly domain-specific words such as hashtags (e.g., #happy). This is surprising because there is not a single hashtag in the lexicon resource WHM that DENSIFIER was trained on. Results for the other three double columns show likewise extreme examples for the corresponding information and language. This initial evaluation indicates that our method effectively learns high quality lexicons for new domains. Figure 3 depicts values for selected words for the three properties. Illustrative examples are “brother” / “brotherhood” for concreteness and “hate” / “love” for sentiment.

³ τ is low even in a setup that is optimistic due to train/test overlap; presumably it would be even lower without overlap.

³The main result of the frequency experiment below is that

EN-twitter		EN-news		EN-news		DE-web	
positive	negative	positive	negative	concrete	abstract	positive	negative
#blessed	rape	expertise	angry	tree	fundamental	herzlichen	gesperrt
inspiration	racist	delighted	delays	truck	obvious	kenntnisse	droht
blessed	horrible	honored	worse	kitchen	legitimate	hervorragende	verurteilt
inspiring	nasty	thank	anger	dog	reasonable	ideale	gefahr
foundation	jealousy	wonderful	foul	bike	optimistic	bestens	falsche
provide	murder	commitment	blamed	bat	satisfied	glückwunsch	streit
wishes	waste	affordable	blame	garden	surprising	optimale	angst
dedicated	mess	passion	complained	homer	honest	anregungen	krankheit
offers	disgusting	exciting	bad	bed	regard	freuen	falschen
#happy	spam	flexibility	deaths	gallon	extraordinary	kompetenzen	verdacht

Table 2: Top 10 words in the output lexicons for the domains Twitter and News (English) and Web (German).

4.2 Quality of Predictions

Table 1 presents experimental results. In each case, we split the resource into train/test, except for Twitter where we used the trial data of SemEval2015 Task 10E for test. We train DENSIFIER on train and compute Kendall’s τ on test. The size of the lexicon resource has no big effect; e.g., results for Spanish (small resource; line 3) and French (large resource; line 4) are about the same. See Section 5.2 for a more detailed analysis of the effect of resource size.

The quality of the output lexicon depends strongly on the quality of the underlying word embeddings; e.g., results for French (small embedding training corpus; line 4) are worse than results for English (large embedding training corpus; line 6) even though the lexicon resources have comparable size.

In contrast to sentiment/concreteness, τ values for frequency are low (lines 8-9). For the other three languages we obtain $\tau \in [.34, .46]$ for frequency (not shown). This suggests that word embeddings represent sentiment and concreteness much better than frequency. The reason for this likely is the learning objective of word embeddings: modeling the context. Infrequent words can occur in frequent contexts. Thus, the frequency information in a single word embedding is limited. In contrast negative words are likely to occur in negative contexts.

The nine output lexicons in Table 1 – each a list of words annotated with predicted strength on one of three properties – are available at www.cis.lmu.de/~sascha/Ultradense/.

		τ	
system		all	\cap
1	Amir et al. (2015)	.626 [†]	
2	Hamdan et al. (2015)	.621 [†]	
3	Zhang et al. (2015)	.591 [†]	
4	Özdemir and Bergler (2015)	.584 [†]	
5	Plotnikova et al. (2015)	.577 [†]	
6	DENSIFIER	.654	.650
7	Sentiment140	.508 [†]	.538 [†]
8	DENSIFIER, trial only	.627 [†]	

Table 3: Results for Lexicon Creation. The first column gives the correlation with the entire test lexicon of SemEval2015 10E, the last column only on the intersection of our output lexicon and Sentiment140. Of the 1315 words of task 10E, 985 and 1308 are covered by DENSIFIER and Sentiment140, respectively. †: significantly worse than the best (bold) result in the same column ($\alpha = .05$, Fisher z-transformation).

4.3 Determining Association Strength

We also evaluate lexicon creation on SemEval2015 Task 10E. As before, the task is to predict the sentiment score of words/phrases. We use the trial data of the task to tune the hyperparameter, $\alpha^s = .4$. Out-of-vocabulary words were predicted as neutral (7/1315). Table 3 shows that the lexicon computed by DENSIFIER (line 5, Table 1) has a τ of .654 (line 6, column *all*), significantly better than all other systems, including the winner of SemEval 2015 ($\tau = .626$, line 1). DENSIFIER also beats Sentiment140 (Mohammad et al., 2013), a widely used semi-automatic sentiment lexicon. The last column is τ on the intersection of DENSIFIER and Sentiment140. It shows that DENSIFIER again performs significantly better than Sentiment140.

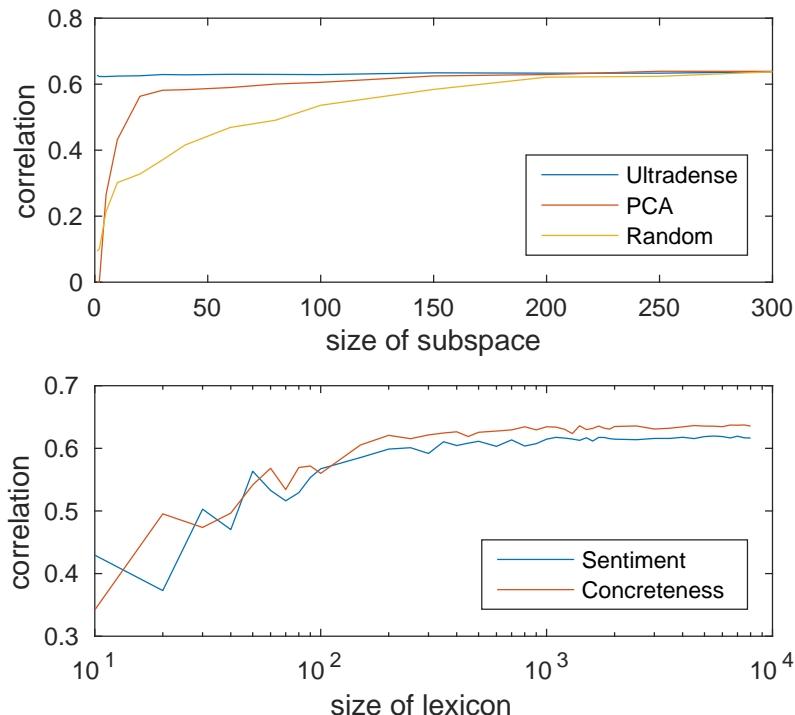


Figure 2: Kendall’s τ versus subspace size (top) and training resource size (bottom). See lines 6 & 8, Table 1, for train/test split.

4.4 Text Polarity Classification

We now show that ultradense embeddings decrease model training times without any noticeable decrease in performance compared to the original embeddings. We evaluate on SemEval2015 Task 10B, classification of Twitter tweets as positive, negative or neutral. We reimplement the linguistically-informed convolutional neural network (lingCNN) of Ebert et al. (2015) that has close to state-of-the-art performance on the task. We do not use sentence-based features to focus on the evaluation of the embeddings. We initialize the first layer of lingCNN, the embedding layer, in three different ways: (i) 400-dimensional Twitter embeddings (Section 3); (ii) 40-dimensional ultradense embeddings derived from (i); (iii) 4-dimensional ultradense embeddings derived from (i). The objective weighting is $\alpha^s = .4$, optimized on the development set.

The embedding layer converts a sentence into a matrix of word embeddings. We also add linguistic features for words, such as sentiment lexicon scores. The combination of embeddings and linguistic features is the input for a convolution layer with filters spanning 2-5 words (100 filters each). This is fol-

lowed by a max pooling layer, a rectifier nonlinearity (Nair and Hinton, 2010) and a fully connected softmax layer predicting the final label. The model is trained with SGD using AdaGrad (Duchi et al., 2011) and ℓ_2 regularization ($\lambda = 5 \times 10^{-5}$). Learning rate is 0.01. Mini-batch size is 100.

We follow the official guidelines and use the SemEval2013 training and development sets as training set, the SemEval2013 test set as development set and the SemEval2015 test set to report final scores (Nakov et al., 2013; Rosenthal et al., 2015). We report macro F_1 of positive and negative classes (the official SemEval evaluation metric) and accuracy over the three classes. Table 4 shows that 40-dimensional ultradense embeddings perform almost as well as the full 400-dimensional embeddings (no significant difference according to sign test). Training time is shorter by a factor of 21 (85/4 examples/second). The 4-dimensional ultradense embeddings lead to only a small loss of 1.5% even though the size of the embeddings is smaller by a factor of 100 (again not a significant drop). Training time is shorter by a factor of 44 (178/4).

We perform the same experiment on CSFD, a

lang.	embeddings	#dim	acc	F_1	ex./sec
en	original	400	.666	.623	4
en	DENSIFIER	40	.662	.620	85
en	DENSIFIER	4	.646	.608	178
cz	original	400	.803	.802	1
cz	DENSIFIER	40	.803	.801	24
cz	DENSIFIER	4	.771	.769	83

Table 4: Performance on Text Polarity Classification

Czech movie review dataset (Habernal et al., 2013), to show the benefits of ultradense embeddings for a low-resource language where only one rather small lexicon is available. As original word embeddings we train new 400 dimensional embeddings on a large Twitter corpus (3.3×10^9 tokens). We use DENSIFIER to create 40 and 4 dimensional embeddings out of these embeddings and SubLex 1.0 (Veselovská and Bojar, 2013). Word polarity features are also taken from SubLex. A simple binary negation indicator is implemented by searching for all tokens beginning with “ne”. Since that includes superlative forms having the prefix “nej”, we remove them with the exception of common negated words, such as “nejsi” – “you are not”. We randomly split the 91,000 dataset instances into 90% train and 10% test and report accuracy and macro F_1 score over all three classes.

Table 4 shows that what we found for English is also true for Czech. There is only a small performance drop when using ultradense embeddings (not significant for 40 dimensional embeddings) while the speed improvement is substantial.

5 Parameter Analysis

In this section, we analyze two parameters: size of ultradense subspace and size of lexicon resource. We leave an evaluation of another parameter, the size of the embedding training corpus, for future work, but empirical results suggest that this corpus should ideally have a size of several billion tokens.

5.1 Size of Subspace

With the exception of the two text polarity classification experiments, all our subspaces have dimensionality $d^* = 1$. The question arises: does a one-dimensional space perhaps have too low a capacity to encode all relevant information and could we further improve our results by increasing the dimen-

sionality of the subspace to values $d^* > 1$? The lexicon resources that we train and test on are all binary; thus, if we use values $d^* > 1$, then we need to map the subspace embeddings to a one-dimensional scale for evaluation. We do this by training, on the train part of the resource, a linear transformation from the ultradense subspace to the one-dimensional scale (e.g., to the sentiment scale).

Figure 2 compares different values of d^s for three different types of subspaces in this setup, i.e., the setup in which the subspace representations are mapped via linear transformation to a one-dimensional sentiment value: (i) *Random*: we take the first d^s dimensions of the original embeddings; (ii) *PCA*: we compute a PCA and take the first d^s principal components; (iii) *Ultradense* subspace of dimensionality d^s . We use the word embeddings and lexicon resources of line 6 in Table 1. For random, the performance starts dropping when the subspace is smaller than 200 dimensions. For PCA, the performance is relatively stable until the subspace becomes smaller than 100. In contrast, ultradense subspaces have almost identical performance for all values of d^s , even for $d^s = 1$. This suggests that a single dimension is sufficient to encode all sentiment information needed for sentiment lexicon creation. However, for other sentiment tasks more dimensions may be needed, e.g., for modeling different emotional dimensions of polarity: fear, sadness, anger etc.

An alternative approach to create a low-dimensional space is to simply train low-dimensional word2vec embeddings. The following experiment suggests that this does not work very well. We used word2vec to train 60-dimensional twitter embeddings with the same settings as on line 5 in Table 1. While the correlation for 400-dimensional embeddings shown in Table 1 is .661, the correlation of 60-dimensional embeddings is only .568. Thus, even though we show that the information in 400-dimensional embeddings that is relevant for sentiment can be condensed into a single dimension, hundreds of dimensions seem to be needed if we use word2vec to collect sentiment information. If we run word2vec with a small dimensionality, only a subset of available sentiment information is “harvested” from the corpus.

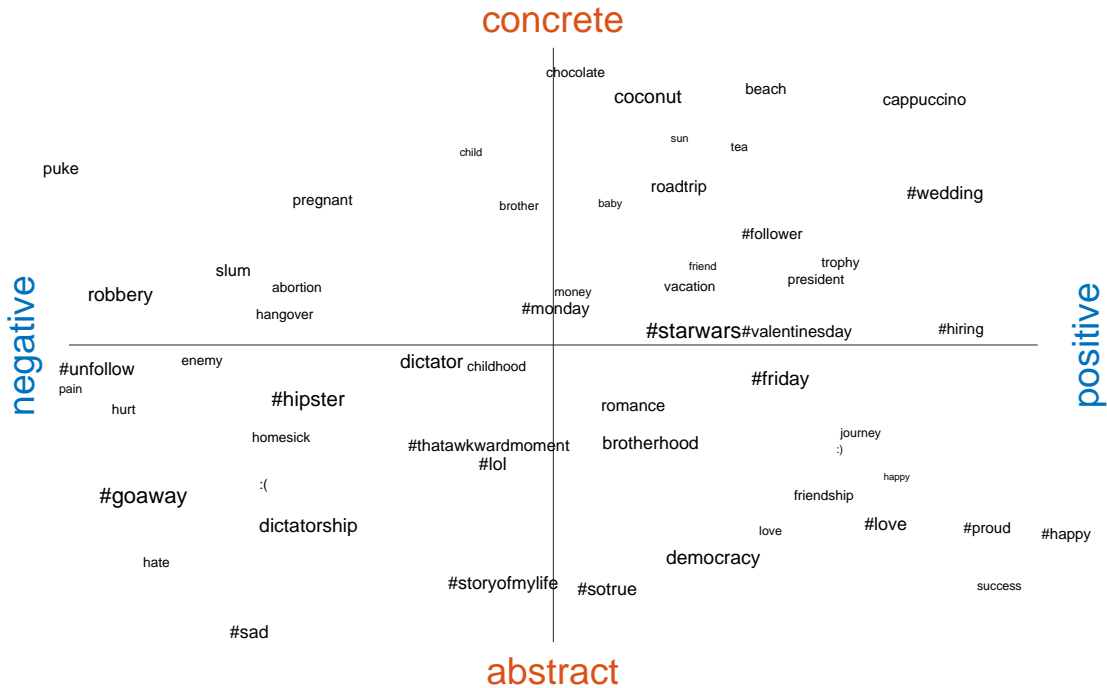


Figure 3: Illustration of EN-twitter output lexicon: DENSIFIER values are x coordinate (sentiment), y coordinate (concreteness) and font size (frequency)

5.2 Size of Training Resource

Next, we analyze what size of training resource is required to learn a good transformation Q . Labeled resources covering many words may not be available or suffer from lack of quality. We use the settings of lines 6 (sentiment) and 7 (concreteness) in Table 1. Figure 2 shows that a small training resource of 300 entries is sufficient for high performance. This suggests that DENSIFIER can create a high quality output lexicon for a new language by hand-labeling only 300 words; and that a small, high-quality resource may be preferable to a large lower-quality resource (semi-automatic or out of domain).

To provide further evidence for this, we train DENSIFIER on only the trial data of SemEval2015 task 10E. To convert the continuous trial data to binary $-1 / 1$ labels, we discard all words with sentiment values between -0.5 and 0.5 and round the remaining values, giving us 39 positive and 38 negative training words. The resulting lexicon has $\tau = .627$ (Table 3, line 8).⁴ This is worse than $\tau =$

⁴Here, we tune α^s on train (equals trial data of SemEval2015 task 10E). This seems to work due to the different

.654 (line 6) for the setup in which we used several large resources, but still better than all previous work. This indicates that DENSIFIER is especially suited for languages or domains for which little training data is available.

6 Related Work

To the best of our knowledge, this paper is the first to train an orthogonal transformation to reorder word embedding dimensions into ultradense subspaces. However, there is much prior work on postprocessing word embeddings.

Faruqui et al. (2015) perform postprocessing based on a semantic lexicon with the goal of fine-tuning word embeddings. Their transformation is not orthogonal and therefore does not preserve distances. They show that their approach optimizes word embeddings for a given application, i.e., word similarity, but also that it worsens them for other applications like detecting syntactic relations. Faruqui et al. (2015)’s approach also does not have the bene-

objectives for training (maximize/minimize difference) and development (correlation).

fit of ultradense embeddings, in particular the benefit of increased efficiency.

In a tensor framework, Rothe and Schütze (2015) transform the word embeddings to sense (synset) embeddings. In their work, all embeddings live in the same space whereas we explicitly want to change the embedding space to create ultradense embeddings with several desirable properties.

Xing et al. (2015) restricted the work of Mikolov et al. (2013) to an orthogonal transformation to ensure that normalized embeddings stay normalized. This transformation is learned between two embedding spaces of different languages to exploit similarities. They normalized word embeddings in a first step, something that did not improve our results.

As a reviewer pointed out, our method is also related to Oriented PCA (Diamantaras and Kung, 1996). However in contrast to PCA a solution for Oriented PCA is not orthogonal.

Sentiment lexicons are often created semi-automatically, e.g., by extending manually labeled seed sets of sentiment words or adding for each word its syno-/antonyms. Alternatively, words frequently cooccurring with a seed set of manually labeled sentiment words are added (Turney, 2002; Kiritchenko et al., 2014). Heerschop et al. (2011) used WordNet together with a PageRank-based algorithm to propagate the sentiment of the seed set to unknown words. Scheible (2010) presented a semi-automatic approach based on machine translation of sentiment lexicons. The winning system of SemEval2015 10E (Amir et al., 2015) was based on structured skip-gram embeddings with 600 dimensions and support vector regression with RBF kernels. Hamdan et al. (2015), the second ranked team, used the average of six sentiment lexicons as a final sentiment score, a method that cannot be applied to low resource languages. We showed that the lexicons created by DENSIFIER achieve better performance than other semi-automatically created lexicons.

Tang et al. (2014b) train sentiment specific embeddings by extending Collobert & Weston’s model and Tang et al. (2014a)’s skip-gram model. The first model automatically labels tweets as positive/negative based on emoticons, a process that cannot be easily transferred to other domains like news. The second uses the Urban Dictionary to expand a small list of 350 sentiment seeds. In our work, we

showed that a training resource of about the same size is sufficient without an additional dictionary. DENSIFIER differs from this work in that it does not need a text corpus, but can transform existing, publicly available word embeddings. DENSIFIER is independent of the embedding learning algorithm and therefore extensible to other word embedding models like GloVe (Pennington et al., 2014), to phrase embeddings (Yu and Dredze, 2015) and even to sentence embeddings (Kiros et al., 2015).

7 Conclusion

We have introduced DENSIFIER, a method that is trained to focus embeddings used for an application to an ultradense subspace that contains the information relevant for the application. In experiments on SemEval, we demonstrate two benefits of the ultradense subspace. (i) Information is preserved even if we focus on a subspace that is smaller by a factor of 100 than the original space. This means that unnecessary noisy information is removed from the embeddings and robust learning without overfitting is better supported. (ii) Since the subspace is 100 times smaller, models that use the embeddings as their input representation can be trained more efficiently and have a much smaller number of parameters. The subspace can be learned with just 80 – 300 training examples, achieving state-of-the-art results on lexicon creation.

We have shown in this paper that up to three orthogonal ultradense subspaces can be created. Many training datasets can be restructured as sets of similar/dissimilar pairs. For instance, in part-of-speech tasks verb/verb pairs would be similar, verb/noun pairs dissimilar. Hence, our objective is widely applicable. In future work, we will explore the possibility of factoring all information present in an embedding into a dozen or so orthogonal subspaces. This factorization would not change the information embeddings contain, but it would make them more compact for any given application, more meaningful and more interpretable.

The nine large DENSIFIER lexicons shown in Table 1 are publicly available.⁵

Acknowledgments. We gratefully acknowledge the support of DFG: grant SCHU 2246/10-1.

⁵www.cis.lmu.de/~sascha/Ultradense/

References

- Amine Abdaoui, Jérôme Azé, Sandra Bringay, and Pascal Poncelet. 2014. FEEL: French Extended Emotional Lexicon: ISLRN: 041-639-484-224-2.
- Silvio Amir, Ramón Astudillo, Wang Ling, Bruno Martins, Mario J. Silva, and Isabel Trancoso. 2015. Inescid: A regression model for large scale twitter sentiment lexicon induction. In *Proceedings of SemEval*.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.
- K. I. Diamantaras and S. Y. Kung. 1996. *Principal Component Neural Networks: Theory and Applications*. John Wiley & Sons, Inc.
- John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159.
- Sebastian Ebert, Ngoc Thang Vu, and Hinrich Schütze. 2015. A Linguistically Informed Convolutional Neural Network. In *Proceedings of WASSA*.
- Ky Fan and Alan J Hoffman. 1955. Some metric inequalities in the space of matrices. volume 6, pages 111–116.
- Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*.
- Ivan Habernal, Tomáš Ptáček, and Josef Steinberger. 2013. Sentiment Analysis in Czech Social Media Using Supervised Machine Learning. In *Proceedings of WASSA*.
- Hussam Hamdan, Patrice Bellot, and Frederic Bechet. 2015. Lsislif: Feature extraction and label weighting for sentiment analysis in twitter. In *Proceedings of SemEval*.
- Bas Heerschop, Alexander Hogenboom, and Flavius Frasincar. 2011. Sentiment lexicon creation from lexical resources. In *Business Information Systems*.
- Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of KDD*.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of ACL*.
- Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of EMNLP*.
- Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *JAIR*, pages 723–762.
- Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of NIPS*.
- Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.
- Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a Word-Emotion Association Lexicon. *Computational Intelligence*, 29(3).
- Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of SemEval*.
- Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of ICML*.
- Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Proceedings of SemEval*.
- Canberk Özdemir and Sabine Bergler. 2015. Clac-sentipipe: Semeval2015 subtasks 10 b,e, and task 11. In *Proceedings of SemEval*.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of EMNLP*.
- Verónica Pérez-Rosas, Carmen Banea, and Rada Mihalcea. 2012. Learning Sentiment Lexicons in Spanish. In *Proceedings of LREC*.
- Nataliia Plotnikova, Micha Kohl, Kevin Volkert, Stefan Evert, Andreas Lerner, Natalie Dykes, and Heiko Ermer. 2015. Klueless: Polarity classification and association. In *Proceedings of SemEval*.
- Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M. Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of SemEval*.
- Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of ACL*.
- Christian Scheible. 2010. Sentiment translation through lexicon induction. In *Proceedings of ACL, Student Research Workshop*.
- Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In *Proceedings of SemEval*.
- Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014a. Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *Proceedings of COLING*.

- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014b. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proceedings of ACL*.
- Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of ACL*.
- Kateřina Veselovská and Ondřej Bojar. 2013. Czech SubLex 1.0.
- Ulli Waltinger. 2010. GermanPolarityClues: A Lexical Resource for German Sentiment Analysis. In *Proceedings of LREC*.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP*.
- Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of NAACL*.
- Mo Yu and Mark Dredze. 2015. Learning composition models for phrase embeddings. *TACL*, 3:227–242.
- Zhihua Zhang, Guoshun Wu, and Man Lan. 2015. Ecnu: Multi-level sentiment analysis on twitter using traditional linguistic features and word embedding features. In *Proceedings of SemEval*.