Sparse Coding of Neural Word Embeddings for Multilingual Sequence Labeling

Gábor Berend
Department of Informatics
University of Szeged
2 Árpád tér, 6720 Szeged, Hungary
berendg@inf.u-szeged.hu

Abstract

In this paper we propose and carefully evaluate a sequence labeling framework which solely utilizes sparse indicator features derived from dense distributed word representations. The proposed model obtains (near) state-of-the art performance for both part-of-speech tagging and named entity recognition for a variety of languages. Our model relies only on a few thousand sparse coding-derived features, without applying any modification of the word representations employed for the different tasks. The proposed model has favorable generalization properties as it retains over 89.8% of its average POS tagging accuracy when trained at 1.2% of the total available training data, i.e. 150 sentences per language.

1 Introduction

Determining the linguistic structure of natural language texts based on rich hand-crafted features has a long-going history in natural language processing. The focus of traditional approaches has mostly been on building linguistic analyzers for a *particular kind of analysis*, which often leads to the incorporation of extensive linguistic and/or domain knowledge for defining the feature space. Consequently, traditional models easily become language and/or task specific resulting in improper generalization properties.

A new research direction has emerged recently, that aims at building more general models that require far less feature engineering or none at all. These advancements in natural language processing, pioneered by Bengio et al. (2003), followed by Collobert and Weston (2008), Collobert et al. (2011), Mikolov et al. (2013a) among others, employ a different philosophy. The objective of these works is to find representations for linguistic phenomena in an unsupervised manner by relying on large amounts of text.

Natural language phenomena are extremely sparse by their nature, whereas continuous word embeddings employ dense representations of words. In our paper we empirically verify via rigorous experiments that turning these dense representations into a much sparser (yet denser than one-hot encoding) form can keep the most salient parts of word representations that are highly suitable for sequence models.

Furthermore, our experiments reveal that our proposed model performs substantially better than traditional feature-rich models in the absence of abundant training data. Our proposed model also has the advantage of performing well on multiple sequence labeling tasks without any modification in the applied word representations thanks to the sparse features derived from continuous word representations.

Our work aims at introducing a novel sequence labeling model solely utilizing features derived from the sparse coding of continuous word embeddings. Even though sparse coding had previously been utilized in NLP prior to us (Faruqui et al., 2015; Chen et al., 2016), to the best of our knowledge, we are the first to propose a sequence labeling framework incorporating it with the following contributions:

- We show that the proposed sparse representation is general as sequence labeling models trained on them achieve (near) state-of-the-art performances for both POS tagging and NER.

Transactions of the Association for Computational Linguistics, vol. 5, pp. 247–261, 2017. Action Editor: Hinrich Schütze.
Submission batch: 12/2015; Revision batch: 5/2016; 11/2016; Published 7/2017.

- We show that the representation is general in the other sense, that it produces reasonable results for more than 40 treebanks for POS tagging,
- rigorously compare different sparse coding approaches in conjunction with differently trained continuous word embeddings,
- highlight the favorable generalization properties of our model in settings when access to a very limited training corpus is assumed,
- release the sparse word representations determined for our experiments at `https://begab.github.io/sparse_embeds` to ensure the replicability of our results and to foster further multilingual NLP research.

2 Related work

The line of research introduced in this paper relies on distributed word representations (Al-Rfou et al., 2013) and dictionary learning for sparse coding (Mairal et al., 2010) and also shows close resemblance to (Faruqui et al., 2015).

2.1 Distributed word representations

Distributed word representations assign some relatively low-dimensional, dense vectors to each word in a corpus such that words with similar context and meaning tend to have similar representations. From an algebraic point of view, the embedding of word i having index idx_i in a vocabulary V can be thought of as the result of a matrix-vector multiplication $W\mathbf{1}_i$, where the i^{th} column of matrix $W \in \mathbb{R}^{k \times |V|}$ contains the k-dimensional ($k \ll |V|$) embedding for word i and vector $\mathbf{1}_i \in \mathbb{R}^{|\mathbb{V}|}$ is the one-hot representation of word i. The one-hot representation of word i is such a vector, which contains zeros for all of its entries except for index idx_i where it stores a one. Depending on how the columns of W (i.e. the word embeddings) get determined, we could distinguish a plethora of approaches (Bengio et al., 2003; Lebret and Collobert, 2014; Mnih and Kavukcuoglu, 2013; Collobert and Weston, 2008; Mikolov et al., 2013a; Pennington et al., 2014).

Prediction-based distributed word embedding approaches such as `word2vec` (Mikolov et al., 2013a) have been conjectured to have superior performance over count-based word representations (Baroni et al., 2014). However, as Lebret and Collobert (2015), Levy et al. (2015) and Qu et al. (2015) point out count-based distributional models can perform on par with prediction-based distributed word embedding models. Levy et al. (2015) illustrate that the effectiveness of neural word embeddings largely depend on the selection of model hyperparameters and other design choices.

According to these findings, in order to avoid any hassles of tuning the hyperparameters of the word embedding model employed, we primarily use the publicly available pre-trained `polyglot` word embeddings of Al-Rfou et al. (2013) instead, without any task specific modification for our experiments. A key thing to note is that `polyglot` word embeddings are not tailored toward any specific language analysis task such as POS tagging or NER. These word embeddings are instead trained in a manner favoring the word analogy task introduced by Mikolov et al. (2013c). The `polyglot` project distributes word embeddings for more than 100 languages. Al-Rfou et al. (2013) also report results on POS tagging, however, word representations they apply for these experiments are different from the task-agnostic representations they made publicly available.

There has been previous research on training neural networks for learning distributed word representations for various specific language analysis tasks. Collobert et al. (2011) propose neural network architectures to four natural language processing tasks, i.e. POS tagging, named entity recognition, semantic role labeling and chunking. Collobert et al. (2011) train word representations on large amounts of unannotated texts from Wikipedia, then update the pretrained word representations for the individual tasks. Our approach is different in that we do not update our word representations for the different tasks and most importantly that we use successfully the features derived from sparse coding in a log-linear model instead of a neural network architecture. A final difference to (Collobert et al., 2011) is that we experiment with a much wider range of languages while they report results for English only.

Qu et al. (2015) evaluate the impacts of choosing different embedding methods on four sequence labeling tasks, i.e. POS tagging, NER, syntactic

chunking and multiword expression identification. The hand-crafted features they employ for POS tagging and NER are the same as in Collobert et al. (2011) and Turian et al. (2010).

2.2 Sparse coding

The general goal of sparse coding is to express signals in the form of *sparse* linear combination of basis vectors and the task of finding an appropriate set of basis vectors is referred to as the dictionary learning problem (Mairal et al., 2010). Generally, given a data matrix $X \in \mathbb{R}^{k \times n}$ with its i^{th} column $\mathbf{x}_i$ representing the i^{th} k-dimensional signal, the task is to find $D \in \mathbb{R}^{k \times m}$ and $\alpha \in \mathbb{R}^{m \times n}$, such that $X \approx D\alpha$. This can be formalized into an ℓ_1-regularized linear least-squares minimization problem having the form

$$\min_{D \in \mathcal{C}, \alpha} \frac{1}{2n} \sum_{i=1}^{n} \left(\|\mathbf{x}_i - D\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \right), \quad (1)$$

with $\mathcal{C}$ being the convex set of matrices of column vectors having an ℓ_2 norm at most one, matrix D acting as the shared dictionary across the signals, and the columns of the sparse matrix α containing the coefficients for the linear combinations of each of the n observed signals.

Performing sparse coding of word embeddings has recently been proposed by Faruqui et al. (2015), however, the objective function they optimize differs from (1). In Section 4, we compare the effects of employing different sparse coding paradigms including the ones in (Faruqui et al., 2015).

In their work, Yogatama et al. (2015) proposed an efficient learning algorithm for determining hierarchically organized sparse word representations using stochastic proximal methods. Most recently, Sun et al. (2016) have proposed an online learning algorithm using regularized dual averaging to directly obtain ℓ_1 regularized continuous bag of words (CBOW) representations (Mikolov et al., 2013a) without the need to determine dense CBOW representations first.

3 Sequence labeling framework

This section introduces the sequence labeling framework we use for both POS tagging and NER. Since our goal is to measure the effectiveness of sparse word embeddings alone, we do not apply any features based on gazetters, capitalization patterns or character suffixes.

As described previously, word embedding methods turn a high-dimensional (i.e., as many dimensions as words in the vocabulary) and extremely sparse (i.e. containing only one non-zero element at the vocabulary index of the word it represents) one-hot encoded representation of words into a dense embedding of much lower dimensionality k.

In our work, instead of using the low dimensional dense word embeddings, we use a dictionary learning approach to obtain sparse codings for the embedded word representations. Formally, given the lookup matrix $W \in \mathbb{R}^{k \times |V|}$ which contains the embedding vectors, we learned $D \in \mathbb{R}^{k \times m}$ being the dictionary matrix shared across all the embedding vectors and $\alpha \in \mathbb{R}^{m \times |V|}$ containing sparse linear combination coefficients for each of the word embeddings so that $\|W - D\alpha\|_F^2 + \lambda \|\alpha\|_1$ is minimized.

Once the dictionary matrix D is learned, the sparse linear combination coefficients $\boldsymbol{\alpha}_i$ can easily be determined for a word embedding vector $\mathbf{w}_i$ by solving an ℓ_1-regularized linear least-squares minimization problem (Mairal et al., 2010). We define features based on vector $\boldsymbol{\alpha}_i$ by taking the signs and indices of its non-zero coefficients, that is

$$f(\mathbf{w}_i) = \{sign(\boldsymbol{\alpha}_i[j])j \mid \boldsymbol{\alpha}_i[j] \neq 0\}, \quad (2)$$

where $\boldsymbol{\alpha}_i[j]$ denotes the j^{th} coefficient in the sparse vector $\boldsymbol{\alpha}_i$. The intuition behind this feature is that words with similar meaning are expected to use an overlapping set of basis vectors from dictionary D. Incorporating the signs of coefficients into the feature function can help to distinguish cases when a basis vector takes part in the reconstruction of a word representation "destructively" or "constructively".

When assigning features to a target word at some position within a sentence, we determine the same set of feature functions for the target word itself and its neighboring words of window size 1. Experiments with window size 2 were also performed. However, we omit these results for brevity as they do not substantially differ from those obtained with a window size of 1.

We then use the previously described set of features in a linear chain CRF (Lafferty et al., 2001)

using CRFsuite (Okazaki, 2007) with its default settings for hyperparameters, i.e., the coefficients of 1.0 and 0.001 for ℓ_1 and ℓ_2 regularization, respectively.

4 Experiments

We rely on the SPArse Modeling Software[1] (SPAMS) (Mairal et al., 2010) for performing sparse coding of distributed word representations. For dictionary learning as formulated in Equation 1, one should choose m and λ, controlling the number of the basis vectors and the regularization coefficient affecting the sparsity of α, respectively. Starting with $m = 256$ and doubling it at each iteration, our preliminary investigations showed a steady growth in the usefulness of sparse word representations as a function of m, plateauing at $m = 1024$. We set m to that value for further experiments.

4.1 Baseline methods

Brown clustering Various studies have identified Brown clustering (Brown et al., 1992) as a useful source of feature generation for sequence labeling tasks (Ratinov and Roth, 2009; Turian et al., 2010; Owoputi et al., 2013; Stratos and Collins, 2015; Derczynski et al., 2015). We should note that sparse coding can also be viewed as a kind of clustering that – unlike Brown clustering – has the capability of assigning word forms to multiple clusters at a time (corresponding to the non-zero coefficients in α).

We thus define a linear chain CRF relying on features from the Brown cluster identifier of words as one of our baseline approach. Since Brown clustering defines a hierarchical clustering over words, cluster supersets can easily function as features. We generate features from length-p ($p \in \{4, 6, 10, 20\}$) prefixes of Brown cluster identifiers similar to Ratinov and Roth (2009) and Turian et al. (2010).

In our experiments we use the implementation by Liang (2005) for performing Brown clustering[2]. We provide the very same Wikipedia articles as input text for determining Brown clusters that are used for training the `polyglot`[3] word embeddings. We also set the number of Brown clusters to be identified to 1024, which is the number of basis vectors applied during sparse coding (cf. $D \in \mathbb{R}^{64\times1024}$).

#	Level	Feature name	
1	char	$\texttt{isNumber}(w_t)$	
2	char	$\texttt{isTitleCase}(w_t)$	
3	char	$\texttt{isNonAlnum}(w_t)$	
4	char	$\texttt{prefix}(w_t, i)$	$1 \le i \le 4$
5	char	$\texttt{suffix}(w_t, i)$	$1 \le i \le 4$
6	word	w_{t+j}	$-2 \le j \le 2$
7	word	$w_t \oplus w_{t+i}$	$1 \le i \le 9$
8	word	$w_t \oplus w_{t-i}$	$1 \le i \le 9$
9	word	$\oplus_{i=t+j}^{t+j+1} w_i$	$-2 \le j \le 1$
10	word	$\oplus_{i=t+j}^{t+j+2} w_i$	$-2 \le j \le 0$
11	word	$\oplus_{i=t+j-1}^{t+j+2} w_i$	$-1 \le j \le 0$
12	word	$\oplus_{i=t-2}^{t+2} w_i$	

Table 1: Features and feature templates applied by our feature-rich baseline for target word w_t. $\oplus$ is a binary operator forming a feature from words and their relative positions by combining them together.

Feature-rich representation We report results relying on linear chain CRFs that assign standard state-of-the-art feature-rich representation to sequences. We apply the very same features and feature templates included in the POS tagging model of CRFSuite[4]. We summarize these features in Table 1, where $\oplus$ denotes the binary operator which defines features as a combination of word forms at different (not necessarily contiguous) positions of a sentence.

We use the same pool of features described in Table 1 for both POS tagging and NER. The reason why we do not adjust the feature-rich representation employed as our baseline for the different tasks is that we do not alter our representation in any way when using our sparse coding-based model either.

Note that features #1 through #5 in Table 1 operate at character-level, whereas our proposed framework solely uses features derived from the sparse coding of word forms. We thus distinguish two feature-rich baselines, i.e. $\text{FR}_{\text{w+c}}$ including both word and character-level features and FR_{w} treating word forms as atomic units to derive features from.

Using dense word representations As our ultimate goal is to demonstrate the usefulness of sparse

[1] `http://spams-devel.gforge.inria.fr/`

[2] `https://github.com/percyliang/brown-cluster`

[3] `https://sites.google.com/site/rmyeid/projects/polyglot`

[4] `http://github.com/chokkan/crfsuite/blob/master/example/pos.py`

features derived from dense word representations, it is important to address the question of whether sparse word representations are more beneficial for sequence labeling tasks compared to their dense counterparts. To this end, we developed a similar model to the one proposed in Section 3, except for using the original dense word representations for inducing features.

According to this modification, we made the following change in our feature function: instead of calculating Equation (2) for some word i, the modified feature function we use for this baseline is

$$f(\mathbf{w}_i) = \{j : \mathbf{w}_i[j] \mid \forall j \in \{1, \ldots, k\}\}.$$

That is, instead of relying on the nonzero values in $\boldsymbol{\alpha}_i$, each word is characterized by its k real-valued coordinates in the embedding space. In order to notationally distinguish sparse and dense representations, we add subscript SC when we refer to a sparse coded version of some word embedding (e.g. SG_{SC}).

4.2 POS tagging experiments

Even though it is reasonable to assume that languages share a common coarse set of linguistic categories, linguistic resources had their own notations for part-of-speech tags. The first notable attempt to canonize the multiple tag sets was the Google universal part-of-speech tags introduced by Petrov et al. (2012) in which the POS tags of various tagging schemes were mapped to 12 language-independent part-of-speech tags.

The recent initiative of universal dependencies (UD) (Nivre, 2015) aims to provide a unified notation for multiple linguistic phenomena, including part-of-speech tags as well. The POS tag set proposed for UD has 17 categories which partially overlap with those defined by Petrov et al. (2012).

4.2.1 Experiments using CoNLL 2006/07 data

We use 12 treebanks in the CoNLL-X format from the CoNLL-2006/07 (Buchholz and Marsi, 2006; Nivre et al., 2007) shared tasks. The complete list of the treebanks included in our experiments is presented in Table 2.

We rely on the official scripts released by Petrov et al. (2012)[5] for mapping the treebank specific POS tags to the Google universal POS tags in order to obtain results comparable across languages. For our experiments we used the original CoNLL-X train/test splits of the treebanks.

[5] https://github.com/slavpetrov/universal-pos-tags

Language	Source
bg	BTB/CoNLL06 (2005)
da	DDT/CoNLL06 (2004)
de	Tiger/CoNLL06 (2002)
en	Penn Treebank (1993)
es	Cast3LB/CoNLL06 (2008)
hu	Szeged Treebank/CoNLL07 (2005)
it	ISST/CoNLL07 (2003)
nl	Alpino/CoNLL06 (2002)
pt	Floresta Sint(c)tica/CoNLL06 (2002)
sl	SDT/CoNLL06 (2006)
sv	Talbanken05/CoNLL06 (2006)
tr	METU-Sabanci/CoNLL07 (2003)

Table 2: Treebanks used for POS tagging experiments from the CoNLL 2006/07 shared task.

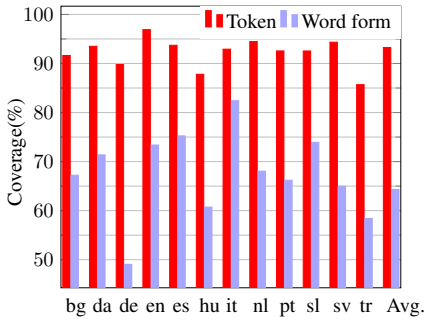


Figure 1: Token and word form-level coverages of the word vectors against the combined train/test sets of the CoNLL-2006/07 POS tagging datasets.

A key factor for the efficiency of our proposed model resides in the coverage of word embeddings, i.e. the proportion of tokens/word forms for which distributed representation is determined. Figure 1 depicts these coverage scores calculated over the merged training and test sets for the different languages. Figure 1 reveals that a substantial amount of tokens has distributed representation defined for (around 90% for the majority of languages, except for Turkish where it is 5 point less). Token coverages of the word embeddings are most likely affected by the morphological richness of the languages and the elaborateness of the corresponding Wikipedia articles used for training word embeddings.

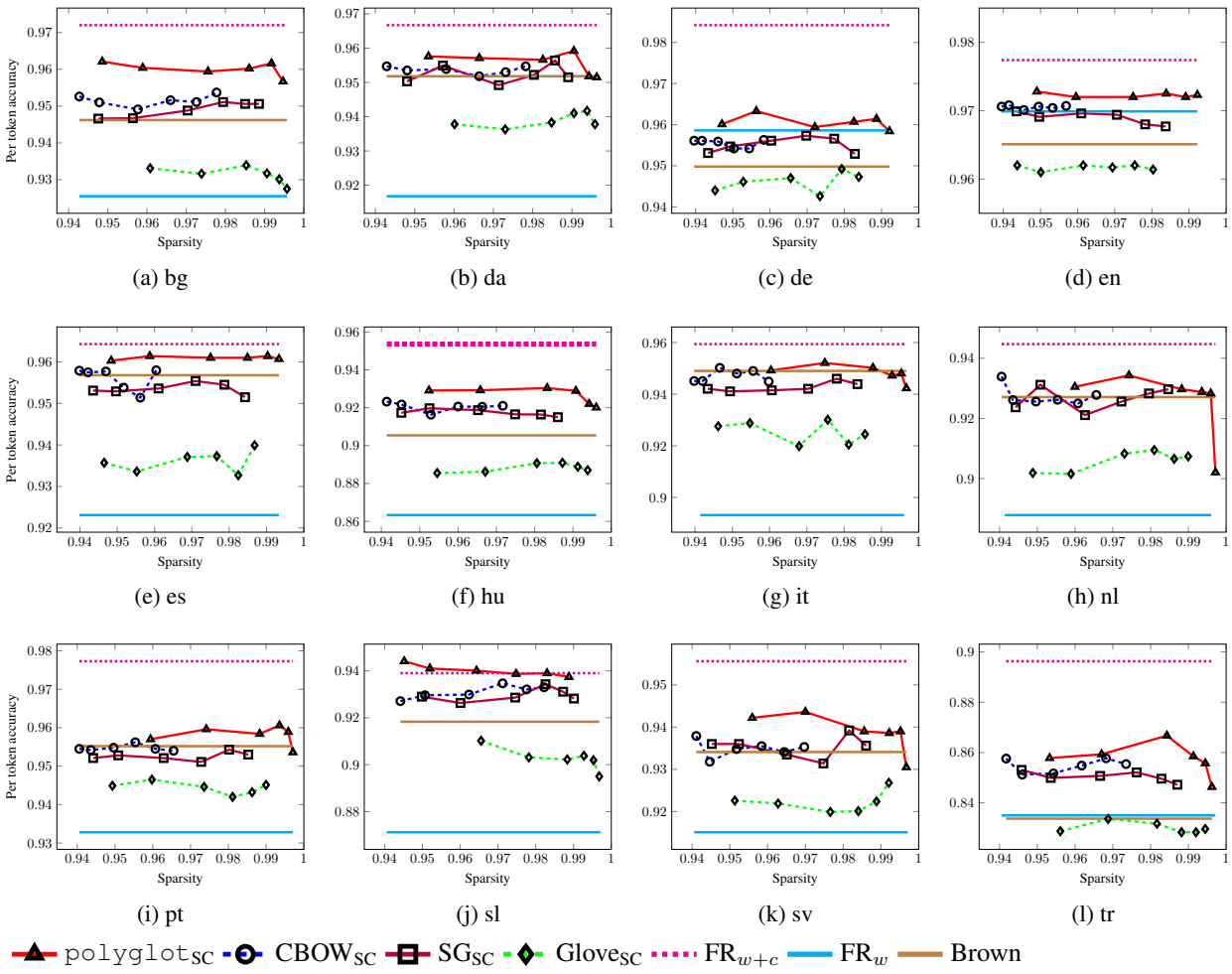


Figure 2: POS tagging results on the CoNLL 2006/07 treebanks evaluating against universal POS tags. Ticks are placed for $\lambda = 0.05, 0.1, 0.2, 0.3, 0.4, 0.5$. The x-axis shows the sparsity of the representations.

Comparing word embeddings Our motivation for choosing `polyglot` word embeddings as input to sparse coding is that they are publicly available for a variety of languages. However, distributed word representations trained in any other reasonable manner can serve as input to our approach. In order to investigate if some of the popular word embedding techniques seem favorable for our algorithm, we conduct experiments using alternatively trained embeddings, i.e. skip-gram (SG), continuous bag-of-words (CBOW) and Glove.

In order that the utility of different word embeddings not to be conflated with other factors, we train them on the same Wikipedia dumps used for training the `polyglot` word vectors. We choose further hyperparameters identically to `polyglot`, i.e. we train 64 dimensional dense word representations using a symmetric context window of size 2 for both SG/CBOW[6] and Glove[7].

Figure 2 includes POS tagging accuracies over the 12 treebanks from the CoNLL 2006/07 shared tasks evaluated against Google Universal POS tags. Instead of reporting results as a function of λ, we rather present accuracies as a function of the different sparsity levels induced by different λ values. Figure 2 demonstrates that POS tagging performance is quite insensitive to the choice of λ unless it yields some extreme sparsity level (>99.5%).

Figure 2 also reveals that the usage of

[6] `https://code.google.com/archive/p/word2vec/`

[7] `http://nlp.stanford.edu/projects/glove/`

	bg	da	de	en	es	hu	it	nl	pt	sl	sv	tr	Avg.
polyglot$_{SC}$	96.04	95.71	96.33	97.20	96.14	92.92	95.21	93.43	95.96	94.10	94.36	85.93	94.44
CBOW$_{SC}$	95.10	95.35	95.61	97.08	95.75	92.17	94.51	92.61	95.42	92.96	93.18	85.12	93.74
SG$_{SC}$	94.67	95.49	95.47	96.91	95.29	91.97	94.11	93.12	95.28	92.63	93.60	84.99	93.63
Glove$_{SC}$	93.16	93.63	94.61	96.10	93.36	88.62	92.88	90.16	94.65	90.31	92.19	83.36	91.92

(a) Results obtained using sparse word representations ($\lambda = 0.1, m = 1024$).

	bg	da	de	en	es	hu	it	nl	pt	sl	sv	tr	Avg.
polyglot	92.11	93.03	93.10	94.80	94.64	89.23	92.90	90.07	94.36	89.36	89.14	81.33	91.17
CBOW	90.19	90.36	88.46	91.22	91.55	86.07	87.11	88.09	92.45	87.82	87.00	79.30	88.30
SG	88.10	88.84	86.48	90.19	91.34	84.38	85.09	85.11	91.77	88.17	84.48	78.72	86.89
Glove	83.10	81.95	83.07	86.64	84.65	77.34	79.98	78.54	86.62	80.91	78.77	76.77	81.53

(b) Results obtained using dense word representations.

Table 3: Performances of sparse and dense word representations for POS tagging over the 12 CoNLL-X datasets.

polyglot$_{SC}$ word representations tend to produce superior results over all alternative representations we experiment with. Furthermore, models using polyglot$_{SC}$ consistently outperform the FR$_w$ and Brown clustering-based baselines.

Models relying on SG$_{SC}$ and CBOW$_{SC}$ representations have an average tagging accuracy of 93.74 and 93.63, respectively, and they typically perform better than the baseline using Brown clustering with an average tagging performance of 93.27. Although utilizing Glove embeddings produce the lowest scores (91.92 on average), its scores still surpass those of the FR$_w$ baseline for all languages except for Turkish.

The average tagging performance over the 12 languages when relying on features based on polyglot$_{SC}$ is only 1.3 points below that of FR_{w+c} (i.e. 94.4 versus 95.7). Recall that FR_{w+c} uses a feature-rich representation, whereas our proposed model uses only $O(m)$ features, i.e. it is tied to the number of the basis vectors employed for sparse coding. Furthermore, our model does not employ word identity features, nor does it rely on character-level features of words.

Analyzing the effects of window size Hyperparameters for training word representations can greatly impact their quality as also concluded by Levy et al. (2015). We thus investigate if providing a larger context window size during the training of CBOW, SG and Glove embeddings can improve their performance in our model.

According to Figure 3 applying context window sizes of 2 for training the word embeddings tend to produce better overall POS tagging accuracies than applying a larger window size of 10. Differences are the most pronounced in case of skip-gram representation, confirming the findings of Lin et al. (2015), i.e. embedding models that model short-range context are more effective for POS tagging.

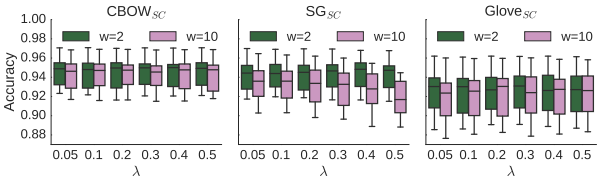


Figure 3: Overview of POS tagging accuracies over the 12 CoNLL-X datasets when relying on sparse coded versions of alternative word embeddings trained with context window size of 2 and 10.

Comparing dense and sparse representations Unless stated otherwise, we use $\lambda = 0.1$ for the experiments below in accordance to Figure 2. Table 3 demonstrates that performances obtained by models using dense word representations as features are consistently inferior to those models relying on sparse word representations.

In Table 3b, we can see that polyglot embeddings perform the best for dense representations as well. When using dense features, the CBOW representation-based model tends to produce results better than by a 1.4 points margin on average compared to SG embeddings. This performance gap between the two word2vec variants vanishes, however, when dense representations are replaced by their sparse counterparts. Table 3 also reveals that

model	bg	da	de	en	es	hu	it	nl	pt	sl	sv	tr	Avg.
polyglot$_{SC}$	96.04	95.71	96.33	97.20	96.14	92.92	95.21	93.43	95.96	**94.10**	94.36	85.93	94.44
FR$_{w}$	92.55	91.68	95.86	96.99	92.31	86.33	89.32	88.79	93.28	87.12	91.51	83.50	90.77
FR$_{w+c}$	**97.20**	**96.67**	**98.42**	**97.74**	**96.43**	**95.36**	**95.94**	**94.47**	**97.73**	93.90	**95.56**	**89.63**	**95.75**
#train sents.	12823	5190	39216	39832	3306	6035	3110	13349	9071	1534	11042	4997	12458

(a) Results obtained with different models when all the training corpora was used.

model	bg	da	de	en	es	hu	it	nl	pt	sl	sv	tr	Avg.
polyglot$_{SC}$	88.20	**94.04**	93.47	**95.76**	**95.63**	91.15	**94.19**	**87.28**	94.60	**94.12**	**91.14**	83.23	**91.90**
FR$_{w}$	79.63	87.75	85.58	90.93	89.87	80.01	86.60	74.40	89.13	86.93	80.16	77.59	85.05
FR$_{w+c}$	**88.71**	93.52	**95.77**	94.59	95.42	**92.74**	93.66	84.94	**95.13**	93.82	88.56	**84.92**	91.82
train sents. %	11.70	28.90	3.82	3.77	45.37	24.86	48.23	11.24	16.54	97.78	13.58	30.02	12.04

(b) Results obtained with different models when the first 1,500 sentences of the training corpora were used.

model	bg	da	de	en	es	hu	it	nl	pt	sl	sv	tr	Avg.
polyglot$_{SC}$	**76.46**	**89.51**	88.29	**90.46**	**91.32**	**86.51**	**89.13**	**75.24**	**90.74**	**86.67**	**82.50**	**71.17**	**84.83**
FR$_{w}$	62.44	74.88	72.46	78.10	77.80	67.20	75.45	56.67	79.38	72.46	65.13	61.38	70.28
FR$_{w+c}$	74.87	83.34	**89.64**	85.75	85.88	83.54	84.99	69.28	87.52	83.88	76.71	67.40	81.07
train sents. %	1.17	2.89	0.38	0.38	4.54	2.49	4.82	1.12	1.65	9.78	1.36	3.00	1.20

(c) Results obtained with different models when the first 150 sentences of the training corpora were used.

Table 4: Comparison of models based on different amount of training data. Bold numbers indicate the best results for a given training regime (i.e. either training on 150/1,500/all training sentences). polyglot$_{SC}$ uses $m = 1024, \lambda = 0.1$.

sparse word representations improve average POS tagging accuracy by 3.3, 5.4, 6.7 and 10.4 points for polylgot, CBOW, SG and Glove word representations, respectively.

Comparing the effects of training corpus size We also investigate the generalization characteristics of the proposed representation by training models that have access to substantially different amounts of training data per language. We distinguish three scenarios, i.e. when using only the first **150**, the first **1,500** and **all** the available training sentences from each corpus. Figure 4 illustrates the average POS tagging accuracy over the 12 CoNLL-X datasets for different amounts of training data and models.

Table 4 further reveals that the average performance of polyglot$_{SC}$ is 14.55 and 3.76 points better compared to the FR$_{w}$ and FR$_{w+c}$ baselines when using only 1.2% of all the available training data, i.e. 150 sentences per language. By discarding 98.8% of the training data polyglot$_{SC}$ obtains 89.8% of its average performance compared to the scenario when it has access to all the training sentences. However, under the same scenario the FR$_{w+c}$ and FR$_{w}$ models only manage to preserve 85% and 77% of their original performance, respectively.

Our model performs on par with FR$_{w+c}$ and has a 6.85 points advantage over FR$_{w}$ with a training corpus of 1,500 sentences. FR$_{w+c}$ has an average of 1.3 points advantage over polyglot$_{SC}$ when we provide access to all training data during training, nevertheless FR$_{w}$ still underperforms polyglot$_{SC}$ in that setting by 3.67 points.

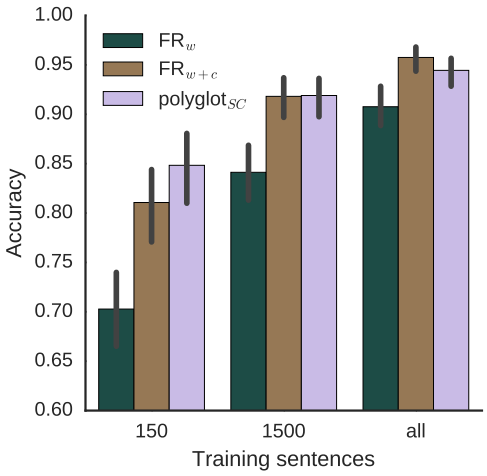


Figure 4: Average tagging accuracies over the 12 CoNLL-X languages using varying amount of training sentences.

Comparing sparse coding techniques Next, we compare different sparse coding approaches on the

pre-trained `polyglot` word representations. The recent work of Faruqui et al. (2015) formulated alternative approaches to determine sparse word representations. One of the objective functions Faruqui et al. (2015) apply is

$$\min_{D,\alpha} \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{x}_i - D\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 + \tau\|D\|_2^2. \quad (3)$$

The main difference in Eq. 1 and 3 is that the latter does not explicitly constrain D to be a member of the convex set of matrices comprising of column vectors having a pre-defined upper bound on their norm. In order to implicitly control for the norms of the basis vectors Faruqui et al. (2015) apply an additional regularization term affected by an extra parameter τ in their objective function.

Faruqui et al. (2015) also formulated a constrained objective function of the form

$$\min_{\substack{D\in\mathbb{R}_{\geq 0}^{k\times m} \\ \alpha\in\mathbb{R}_{\geq 0}^{k\times|V|}}} \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{x}_i - D\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 + \tau\|D\|_2^2, \quad (4)$$

for which a non-negativity constraint on the elements of α (but no constraint on D) is imposed. When using the objective functions introduced by Faruqui et al. (2015), we use the default $\tau = 10^{-5}$ value. Notationally, we distinguish the sparse coding approaches based on the equation they use as their objective function, i.e. SC-*i*, $i \in \{1, 3, 4\}$.

We applied $\lambda = 0.05$ for SC-1 and $\lambda = 0.5$ for SC-3 and SC-4 in order to obtain word representations of comparable average sparsity levels across the 12 languages, i.e. 95.3%, 94.5% and 95.2%, respectively (cf. the left of Figure 5). The right of Figure 5 further illustrates the spread of POS tagging accuracies over the 12 CoNLL-X treebanks when using models that rely on different sparse coding strategies with comparable sparsity levels.

Although Murphy et al. (2012) mentions non-negativity as a desired property of word representations for cognitive plausibility, Figure 5 reveals that our sequence labeling model cannot benefit from it as the average POS tagging accuracy for SC-4 is 0.7 points below that of SC-3 approach. The average performances when applying SC-1 and SC-3 are nearly identical with a 0.18 point difference between the two.

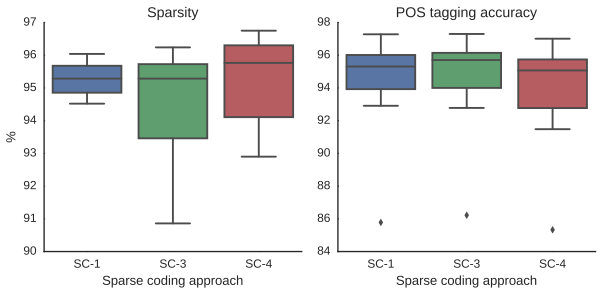
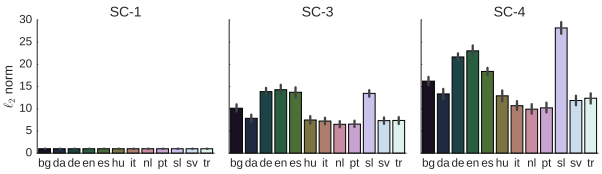
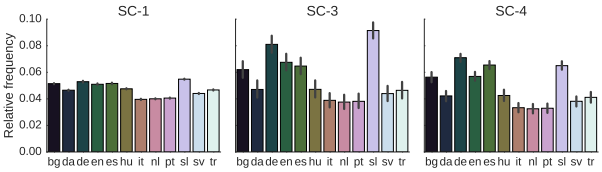


Figure 5: Comparison of the POS tagging accuracies of different sparse coding techniques with comparable average sparseness levels over the 12 CoNLL-X languages.



(a) ℓ_2 norms



(b) Relative frequencies

Figure 6: Characteristics of the different sparse coding techniques over the 12 CoNLL-X languages.

It is instructive to analyze the patterns different sparse coding approaches exhibit. Even though the objective functions used by the different approaches are similar, decompositions obtained by them convey rather different sparsity structures.

Figure 6a illustrates that there exist substantial variation in the length of the basis vectors obtained by SC-3 and SC-4 both within and across languages. However, SC-1 produces practically no variation in the length of the basis vectors comprising D due to the constraint present in the objective function it employs. Figure 6b shows similar differences about the relative frequency of basis vectors taking part in the reconstruction of word embeddings.

Figure 7 shows a strong correlation between the ℓ_2 norm of basis vectors and the relative number of times a non-zero coefficient is assigned to them in α for SC-3 and SC-4 but not for SC-1.

It can be further noted from Figure 7 that the norm

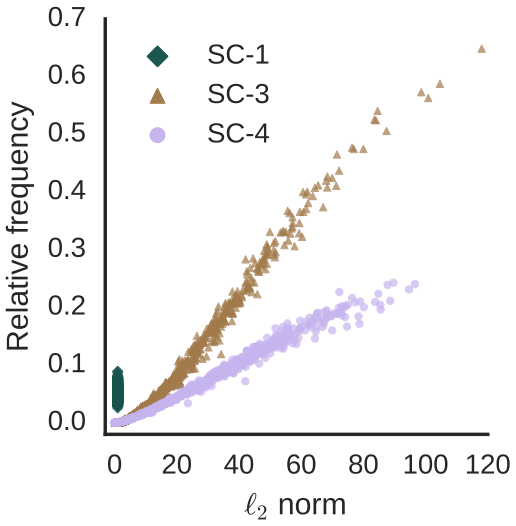


Figure 7: Relative frequency of basis vectors receiving nonzero coefficients in α as a function of their ℓ_2 norm.

of the basis vectors determined by SC-3 and SC-4 are often orders of magnitude larger than those determined by SC-1. This effect, however, can be naturally mitigated by increasing τ.

Overall, the different approaches convey comparable POS tagging accuracies but different decompositions due to the differences in the objective functions they employ. Experiments described below are conducted using the objective function in Eq. 1.

4.2.2 Experiments using UD treebanks

For POS tagging we also experiment with UD v1.2 (Nivre et al., 2015) treebanks. We used the default train-test splits of the treebanks not utilizing the development sets for fine tuning performance on any of the languages during our experiments. We omitted the Japanese treebank as words in it are stripped off due to licensing issues. Also there is no `polyglot` vector released for Old Church Slavonic and Gothic. Even though `polyglot` word representations are released for Arabic, it was of no practical use as it contained unvocalized surface forms of tokens in contrast to the vocalized forms in UD v.1.2. For this reason, we discarded the Arabic treebank as less than 30% of its tokens could be associated with a representation. By omitting these 4 languages from our experiments we are finally left with 33 treebanks for 29 languages. We note that for Ancient Greek treebanks (grc*) we use word embeddings trained on Modern Greek.

We should add that there are 4 languages (related to 6 treebanks) for which `polyglot` word vectors are accessible, however, the Wikipedia dumps used for training them are not distributed. For this reason, Brown clustering-based baselines are missing for the affected treebanks.

We report our results on UD v1.2 in Table 5. Recall that the default behavior of our sparse coding-based models (SC in Table 5) is that they do not handle word identity as an explicit feature. We now investigate how much contribution word identity features convey on their own and also when used in conjunction with sparse coding-derived features. For this end we introduce a simple linear chain CRF model generating features solely on the identity of the current word and the ones surrounding it (WI in Table 5). Likewise, we define a model that relies on WI and SC features simultaneously (WI+SC). Table 5 reveals that SC outperforms WI by a large margin and that combining the two feature sets together yields some further improvements over SC scores.

We also present in Table 5 the state-of-the-art results of the bidirectional LSTM models by Plank et al. (2016) for comparative purposes. Note that the authors reported results only on a subset of UD v1.2 (i.e. treebanks with at least 60k tokens), for which reason we can include their results on 21 treebanks. Out of these 21 UD v1.2 treebanks there are 15 and 20 cases, respectively, for which SC and WI+SC produces better results than bi-LSTM$_{\text{w}}$. Only FR$_{\text{w+c}}$ and bi-LSTM$_{\text{w+c}}$, models which enjoy the additional benefit of employing character-level features besides word-level ones, are capable of outperforming SC and WI+SC.

4.3 Named entity recognition experiments

Besides the POS tagging experiments, we investigated if the very same features as the ones applied for POS tagging can be utilized in a different sequence labeling task, namely named entity recognition. In order to evaluate our approach, we obtained the English, Spanish and Dutch datasets from the 2002 and 2003 CoNLL shared tasks on multilingual Named Entity Recognition (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003).

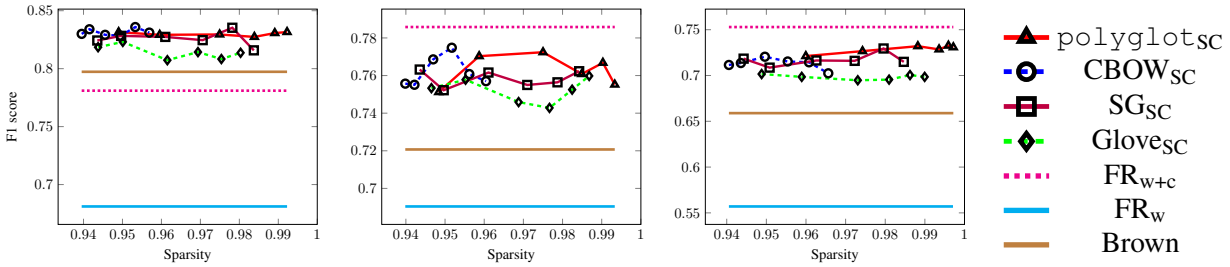
We use the train-test splits provided by the or-

	Baseline using					Word	Sparse		
	Words and characters		Words only			Identity	coding		Token
Treebank	bi-LSTM$_{w+c}$	FR$_{w+c}$	bi-LSTM$_{w}$	FR$_{w}$	Brown	(WI)	(SC)	WI+SC	coverage
bg	98.25	96.88	95.12	90.40	93.36	90.75	**95.33**	**95.63**	92.64
cs	97.93	98.03	93.77	93.09	91.98	93.40	**95.13**	**95.83**	92.42
da	95.94	94.70	91.96	87.41	92.45	87.51	**93.32**	**93.29**	93.96
de	93.11	91.73	90.33	85.73	88.52	85.90	89.11	**90.73**	92.75
el	—	96.77	—	90.91	95.96	91.53	**96.91**	**97.12**	95.80
en	94.61	93.52	92.10	89.28	91.40	89.36	**93.03**	**93.47**	97.61
es	95.34	94.37	93.60	90.93	93.83	91.31	**94.43**	**94.69**	97.08
et	—	84.83	—	75.42	84.52	76.78	**85.56**	**86.30**	80.40
eu	94.91	93.03	88.00	83.36	—	84.83	**90.19**	**90.63**	90.98
fa	96.89	96.13	95.31	93.98	95.04	94.45	**95.91**	**96.11**	97.80
fi	95.18	92.93	87.95	82.31	85.98	83.17	**88.80**	**89.19**	84.37
fi_ftb	—	91.84	—	86.91	82.86	81.57	**86.91**	**87.88**	83.92
fr	96.04	95.30	94.44	92.80	92.42	92.88	93.52	**94.96**	92.06
ga	—	89.64	—	84.32	—	**85.21**	**88.22**	**88.82**	88.80
grc	—	93.57	—	84.35	57.13	**84.44**	70.27	**85.04**	43.58
grc_proiel	—	96.39	—	90.73	49.41	**91.01**	67.17	**91.38**	45.74
he	95.92	93.91	93.37	90.17	93.79	90.33	**94.38**	**95.28**	92.03
hi	96.64	95.96	95.99	94.32	94.61	94.25	95.37	**96.09**	96.40
hr	95.59	94.18	89.24	82.91	92.22	83.52	**92.85**	**93.53**	92.45
hu	—	92.88	—	73.69	91.08	75.63	89.47	89.47	90.07
id	92.79	93.32	90.48	87.29	91.39	88.03	**91.71**	**92.02**	97.09
it	97.64	96.92	96.57	93.62	94.92	93.43	95.70	96.28	94.99
la	—	92.03	—	77.75	—	**79.99**	**85.49**	**86.34**	83.03
la_itt	—	98.78	—	97.69	—	**97.74**	95.43	**97.77**	92.23
la_proiel	—	95.89	—	90.53	—	**90.84**	90.14	**92.42**	85.21
nl	92.07	88.79	84.96	81.11	84.28	81.27	84.32	**85.10**	92.28
no	97.77	96.53	94.39	91.58	94.29	91.87	**95.42**	**95.67**	94.53
pl	96.62	95.27	89.73	84.41	91.13	84.57	**93.57**	**93.95**	94.19
pt	97.48	96.59	94.24	90.69	93.74	91.11	94.00	**95.50**	92.53
ro	—	86.46	—	76.32	89.93	75.96	88.99	88.27	93.06
sl	97.78	95.28	91.09	84.43	90.24	84.92	**92.65**	**92.70**	92.14
sv	96.30	94.94	93.32	88.84	93.50	88.94	**94.46**	**94.62**	92.50
ta	—	85.37	—	68.02	—	**70.69**	**81.25**	**81.80**	85.35
Avg.	95.99	94.76	92.40	88.77	91.95	89.05	93.15	93.73	93.59

Table 5: Per token POS tagging accuracies for 33 UD treebanks. For sparse coding SPAMS is used on `polyglot` vectors with $\lambda = 0.1$ and $m = 1024$. Results in bold are better than any of bi-LSTM$_{w}$, FR$_{w}$ and Brown models (i.e. the baselines using features based on words only). Average is calculated over the 20 highlighted treebanks for which there are results in every column. The bi-LSTM results are from Plank et al. (2016).

ganizers and report our NER results using the F1 scores based on the official evaluation script of the CoNLL shared task. Similar to Collobert et al. (2011) we also apply the 17-tag IOBES tagging scheme during training and inference. The best F1 scores reported for English by Collobert et al. (2011) without employing additional unlabeled texts to enhance their language model is 81.47. When pre-training their neural language model on large amounts of Wikipedia texts they report an F1 score of 87.58.

Figure 8 includes our NER results obtained using different word embedding representations as input for sparse coding and different levels of sparsity. Similar to our POS tagging experiments, using `polyglot`$_{SC}$ vectors tend to perform best for NER as well. However, a substantial difference compared to the POS tagging results is that NER performances



(a) en (b) es (c) nl

Figure 8: NER results relying on sparse coding of different word representations. The x-axis shows the sparsity of the representations with ticks at $\lambda = 0.05, 0.1, 0.2, 0.3, 0.4, 0.5$.

	en	es	nl	Avg.
polyglot$_{SC}$	82.92	77.03	72.66	77.54
CBOW$_{SC}$	83.40	75.51	71.36	76.76
SG$_{SC}$	82.83	75.22	70.86	76.30
Glove$_{SC}$	82.31	75.78	69.85	75.98

(a) Sparse ($m = 1024, \lambda = 0.1$)

	en	es	nl	Avg.
polyglot	78.80	70.13	65.58	71.50
CBOW	72.68	64.49	64.80	67.32
SG	74.68	66.17	63.95	68.27
Glove	74.33	65.11	57.73	65.72

(b) Dense

Table 6: Comparison of the performance of sparse and dense word representations for NER.

do not degrade even for extreme levels of sparsity. Also, the sparse coding-based models perform much better when compared to the FR$_{w+c}$ baseline.

In Table 6, we compare the effectiveness of models relying on sparse and dense word representations for NER. In order not to fine-tune hyperparameters for a particular experiment, similarly to our previous choices m and λ are set to 1024 and 0.1, respectively. Results in Table 6 are in line with those reported in Table 3 for POS tagging.

5 Conclusion

In this paper we show that it is possible to train sequence models that perform nearly as well as best existing models on a variety of languages for both POS tagging and NER. Our approach does not require word identity features to perform reliably, furthermore, it is capable of achieving comparable results to traditional feature-rich models. We also illustrate the advantageous generalization property of our model as it retained 89.8% of its original average POS tagging accuracy when trained on only 1.2% of the total accessible training sentences.

As Mikolov et al. (2013b) pointed out the similarities of continuous word embeddings across languages, we think that our proposed model could be employed not in just multi-lingual, but also in cross-lingual language analysis settings. In fact, we investigate its feasibility in our future work. Finally, we have made the sparse coded word embedding vectors publicly available in order to facilitate the reproducibility of our results and to foster multilingual and cross-lingual research.

Acknowledgement

The author would like to thank the TACL editors and the anonymous reviewers for their valuable feedbacks and suggestions.

References

Susana Afonso, Eckhard Bick, Renato Haber, and Diana Santos. 2002. "Floresta sintá(c)tica": a treebank for Portuguese. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, pages 1698–1703. European Language Resources Association (ELRA).

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192. Association for Computational Linguistics.

Nart B. Atalay, Kemal Oflazer, and Bilge Say. 2003. The annotation process in the Turkish treebank. In

Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC), pages 33–38. Association for Computational Linguistics.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 149–164. Association for Computational Linguistics.

Yunchuan Chen, Lili Mou, Yan Xu, Ge Li, and Zhi Jin. 2016. Compressing neural language models by sparse word representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–235. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167. Association for Computing Machinery.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Dóra Csendes, János Csirik, Tibor Gyimóthy, and András Kocsor. 2005. The Szeged Treebank. In *Text, Speech and Dialogue, 8th International Conference, TSD 2005 Proceedings*, pages 123–131.

Leon Derczynski, Sean Chester, and Kenneth Bøgh. 2015. Tune your brown clustering, please. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 110–117. INCOMA Ltd. Shoumen, Bulgaria.

Sašo Džeroski, Tomaž Erjavec, Nina Ledinek, Petr Pajas, Zdeněk Žabokrtský, and Andreja Žele. 2006. Towards a Slovene dependency treebank. In *Proceedings of the Fifth International Language Resources and Evaluation Conference, LREC 2006*, pages 1388–1391. European Language Resources Association (ELRA).

Simonetta Montemagni et al. 2003. Building the Italian syntactic-semantic treebank. In *Building and using Parsed Corpora*, Language and Speech series, pages 189–210. Kluwer.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1491–1500. Association for Computational Linguistics.

Matthias T. Kromann, Line Mikkelsen, and Stine Kern Lynge. 2004. Danish dependency treebank.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289. Morgan Kaufmann Publishers Inc.

Rémi Lebret and Ronan Collobert. 2014. Word embeddings through Hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–490. Association for Computational Linguistics.

Rémi Lebret and Ronan Collobert. 2015. Rehabilitation of count-based models for word vector representations. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 417–429. Springer.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Percy Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised POS induction with word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316. Association for Computational Linguistics.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2010. Online learning for matrix factorization and sparse coding. *The Journal of Machinea Learning Research*, 11:19–60.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013c. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119. Curran Associates Inc.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26*, pages 2265–2273. Curran Associates, Inc.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of COLING 2012*, pages 1933–1950. The COLING 2012 Organizing Committee.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1392–1395. European Language Resources Association (ELRA).

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932. Association for Computational Linguistics.

Joakim Nivre et al. 2015. Universal dependencies 1.2. http://hdl.handle.net/11234/1-1548. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

Joakim Nivre, 2015. *Towards a Universal Grammar for Natural Language Processing*, pages 3–16. Springer International Publishing.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs).

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096. European Language Resources Association (ELRA).

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418. Association for Computational Linguistics.

Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, Nathan Schneider, and Timothy Baldwin. 2015. Big data small data, in domain out-of domain, known word unknown word: The impact of word representations on sequence labelling tasks. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 83–93. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155. Association for Computational Linguistics.

Mariona Taulé M. Antònia Martí Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08)*, pages 96–101. European Language Resources Association (ELRA).

Kiril Simov and Petya Osenova. 2005. Extending the annotation of BulTreeBank: Phase 2. In *The Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, pages 173–184.

Karl Stratos and Michael Collins. 2015. Simple semi-supervised POS tagging. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 79–87. Association for Computational Linguistics.

Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2016. Sparse word embeddings using ℓ_1 regularized online learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intel-*

ligence, pages 2915–2921. AAAI Press / International Joint Conferences on Artificial Intelligence.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394. Association for Computational Linguistics.

Leonoor van der Beek, Gosse Bouma, Jan Daciuk, Tanja Gaustad, Robert Malouf, Gertjan van Noord, Robbert Prins, and Begoa Villada. 2002. Chapter 5. The Alpino dependency treebank. In *Algorithms for Linguistic Processing NWO PIONIER Progress Report*.

Dani Yogatama, Manaal Faruqui, Chris Dyer, and Noah Smith. 2015. Learning word representations with hierarchical sparse coding. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 87–96. PMLR.