Unsupervised Morphology Induction Using Word Embeddings

Radu Soricut
Google Inc.
rsoricut@google.com

Franz Och[*]
Human Longevity Inc.
och@humanlongevity.com

Abstract

We present a language agnostic, unsupervised method for inducing morphological transformations between words. The method relies on certain regularities manifest in high-dimensional vector spaces. We show that this method is capable of discovering a wide range of morphological rules, which in turn are used to build morphological analyzers. We evaluate this method across six different languages and nine datasets, and show significant improvements across all languages.

1 Introduction

Word representations obtained via neural networks (Bengio et al., 2003; Socher et al., 2011a) or specialized models (Mikolov et al., 2013a) have been used to address various natural language processing tasks (Mnih et al., 2009; Huang et al., 2014; Bansal et al., 2014). These vector representations capture various syntactic and semantic properties of natural language (Mikolov et al., 2013b). In many instances, natural language uses a small set of concepts to render a much larger set of meaning variations via morphology. We show in this paper that morphological transformations can be captured by exploiting regularities present in word-representations as the ones trained using the SkipGram model (Mikolov et al., 2013a).

In contrast to previous approaches that combine morphology with vector-based word representations (Luong et al., 2013; Botha and Blunsom, 2014), we do not rely on an external morphological analyzer, such as Morfessor (Creutz and Lagus, 2007). Instead, our method automatically induces morphological rules and transformations, represented as vectors in the same embedding space.

At the heart of our method is the SkipGram model described in (Mikolov et al., 2013a). We further exploit the observations made by Mikolov et al (2013b), and further studied by (Levy and Goldberg, 2014; Pennington et al., 2014), regarding the regularities exhibited by such embedding spaces. These regularities have been shown to allow inferences of certain types (e.g., *king* is to *man* what *queen* is to *woman*). Such regularities also hold for certain morphological relations (e.g., *car* is to *cars* what *dog* is to *dogs*). In this paper, we show that one can exploit these regularities to model, in a principled way, prefix- and suffix-based morphology. The main contributions of this paper are as follows:

1. provides a method by which morphological rules are learned in an unsupervised, language-agnostic fashion;

2. provides a mechanism for applying these rules to known words (e.g., *boldly* is analyzed as *bold+ly*, while *only* is not);

3. provides a mechanism for applying these rules to rare and unseen words;

We show that this method improves state-of-the-art performance on a word-similarity rating task using standard datasets. We also quantify the impact of our morphology treatment when using large amounts of training data (tens/hundreds of billions of words).

The technique we describe is capable of inducing transformations that cover both typical, regular morphological rules, such as adding suffix *ed*

[*] Work done at Google, now at Human Longevity Inc.

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 1627–1637,
Denver, Colorado, May 31 – June 5, 2015. ©2015 Association for Computational Linguistics

to verbs in English, as well as exceptions to such rules, such as the fact that pluralization of words that end in *y* require substituting it with *ies*. Because each such transformation is represented in the high-dimensional embedding space, it therefore captures the semantics of the change. Consequently, it allows us to build vector representations for any unseen word for which a morphological analysis is found, therefore covering an unbounded (albeit incomplete) vocabulary.

Our empirical evaluations show that this language-agnostic technique is capable of learning morphological transformations across various language families. We present results for English, German, French, Spanish, Romanian, Arabic, and Uzbek. The results indicate that the induced morphological analysis deals successfully with sophisticated morphological variations.

2 Previous Work

Many recent proposals in the literature use word-representations as the basic units for tackling sentence-level tasks such as language modeling (Mnih and Hinton, 2007; Mikolov and Zweig, 2012), paraphrase detection (Socher et al., 2011a), sentiment analysis (Socher et al., 2011b), discriminative parsing (Collobert, 2011), as well as similar tasks involving larger units such as documents (Glorot et al., 2011; Huang et al., 2012; Le and Mikolov, 2014). The main advantage offered by these techniques is that they can be both trained in an unsupervised manner, and also tuned using supervised labels. However, most of these approaches treat words as units, and fail to account for phenomena involving the relationship between various morphological forms that affect word semantics, especially for rare or unseen words.

Previous attempts at dealing with sub-word units and their compositionality have looked at explicitly-engineered features such as stems, cases, POS, etc., and used models such as factored NLMs (Alexandrescu and Kirchhoff, 2006) to obtain representations for unseen words, or compositional distributional semantic models (Lazaridou et al., 2013) to derive representations for morphologically-inflected words, based on the composing morphemes. A more recent trend has seen proposals that deal with morphology using vector-space representations (Luong et al., 2013; Botha and Blunsom, 2014). Given word morphemes (affixes, roots), a neural-network architecture (recursive neural networks in the work of Luong et al (2013), log-bilinear models in the case of Botha and Blunsom (2014)), is used to obtain embedding representations for existing morphemes, and also to combine them into (possibly novel) embedding representations for words that may not have been seen at training time.

Common to these proposals is the fact that the morphological analysis of words is treated as an external, preprocessing-style step. This step is done using off-the-shelf analyzers such as Morfessor (Creutz and Lagus, 2007). As a result, the morphological analysis happens within a different model compared to the model in which the resulting morphemes are consequently used. In contrast, the work presented here uses the same vector-space embedding to achieve both the morphological analysis of words and to compute their representation. As a consequence, the morphological analysis can be justified in terms of the relationship between the resulting representation and other words that exhibit similar morphological properties.

3 Morphology Induction using Embedding Spaces

The method we present induces *morphological transformations* supported by evidence in terms of regularities within a word-embedding space. We describe in this section the algorithm used to induce such transformations.

3.1 Morphological Transformations

We consider two main transformation types, namely prefix and suffix substitutions. Other transformation types can also be considered, but we restrict the focus of this work to morphological phenomena that can be modeled via prefixes and suffixes.

We provide first a high-level description of our algorithm, followed by details regarding the individual steps. The following steps are applied to monolingual training data over a finite vocabulary V:

1. Extract candidate prefix/suffix rules from V
2. Train embedding space $E^n \subset \mathbb{R}^n$ for all words in V

3. Evaluate quality of candidate rules in E^n
4. Generate lexicalized morphological transformations

We provide more detailed descriptions next.

Extract candidate rules from V

Starting from $(w_1, w_2) \in V^2$, the algorithm extracts all possible prefix and suffix substitutions from w_1 to w_2, up to a specified size[1]. We denote such substitutions using triplets of the form `type:from:to`. For instance, triplet `suffix:ed:ing` denotes the substitution of suffix *ed* with suffix *ing*; this substitution is supported by many word pairs in an English vocabulary, e.g. (*bored, boring*), (*stopped, stopping*), etc. We call these triplets candidate rules, because they form the basis of an extended set from which the algorithm extracts morphological rules.

At this stage, the candidate rules set contains both rules that reflect true morphology phenomena, e.g. `suffix:s:`ϵ (replace suffix *s* with the null suffix, extracted from (*stops, stop*), (*weds, wed*), etc.), or `prefix:un:`ϵ (replace prefix *un* with the null prefix, from (*undone, done*), etc.), but also rules that simply reflect surface-level coincidences, e.g. `prefix:S:`ϵ (delete *S* at the beginning of a word, from (*Scream, cream*), (*Scope, cope*), etc.).

Train embedding space

Using a large monolingual corpus, we train a word-embedding space E^n of dimensionality n for all words in V using the SkipGram model (Mikolov et al., 2013a). For the experiments reported in this paper, we used our own implementation of this model (which varies only slightly from the publicly-available `word2vec` implementation[2]).

Evaluate quality of candidate rules

The extracted candidate rules set is evaluated by using, for each proposed rule r, its support set:

$$S_r = \{(w_1, w_2) \in V^2 | w_1 \xrightarrow{r} w_2\}$$

The notation $w_1 \xrightarrow{r} w_2$ means that rule r applies to word w_1 (e.g., for rule `suffix:ed:ing`, word w_1 ends with suffix *ed*), and the result of applying the rule to word w_1 is word w_2. To speed up computation, we downsample the sets S_r to a large-enough number of word pairs (1000 has been used in the experiments in this paper).

rule	hit rate	Example $\uparrow d_w$
`suffix:er:o`	0.8	$\uparrow d_{\text{Vot}\underline{\text{er}}}$
`suffix:ton:`ϵ	1.1	$\uparrow d_{\text{Gale}\underline{\text{ton}}}$
`prefix:S:`ϵ	1.6	$\uparrow d_{\underline{\text{S}}\text{DK}}$
`prefix:`ϵ`:in`	28.8	$\uparrow d_{\text{\_competent}}$
`suffix:ly:`ϵ	32.1	$\uparrow d_{\text{official}\underline{\text{ly}}}$
`prefix:`ϵ`:re`	37.0	$\uparrow d_{\text{\_sited}}$
`prefix:un:re`	39.0	$\uparrow d_{\underline{\text{un}}\text{made}}$
`suffix:st:sm`	52.5	$\uparrow d_{\text{egoi}\underline{\text{st}}}$
`suffix:ted:te`	54.9	$\uparrow d_{\text{imita}\underline{\text{ted}}}$
`suffix:ed:ing`	68.1	$\uparrow d_{\text{procur}\underline{\text{ed}}}$
`suffix:y:ies`	69.6	$\uparrow d_{\text{foundr}\underline{\text{y}}}$
`suffix:t:ts`	73.0	$\uparrow d_{\text{pugilis}\underline{\text{t}}}$
`suffix:sed:zed`	80.1	$\uparrow d_{\text{seriali}\underline{\text{sed}}}$

Table 1: Candidate rules evaluated in E^n.

We define a generic evaluation function Ev^F over paired couples in $S_r \times S_r$, using a function $F : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, as follows:

$$\begin{gathered} Ev^F((w_1, w_2), (w, w')) = F_E(w_2, w_1 + \uparrow d_w) \quad (1) \\ (w_1, w_2), (w, w') \in S_r, \ \uparrow d_w = w' - w \end{gathered}$$

Word-pair combinations in $S_r \times S_r$ are evaluated using Eq. 1 to assess the meaning-preservation property of rule r. We use as F_E function rank$_E$, the cosine-similarity rank function in E^n. We can quantitatively measure the assertion "*car* is to *cars* what *dog* is to *dogs*", as rank$_E$($cars$, car+$\uparrow d_{dog\_}$). We use a single threshold t^0_{rank} to capture meaning preservation (all the experiments in this paper use $t^0_{\text{rank}} = 100$): for each proposed rule r, we compute a hit rate based on the number of times Eq. 1 scores above t^0_{rank}, over the number of times it has been evaluated. In Table 1 we present some of these candidate rules and their hit rate.

We note that rules that are non-meaning–preserving receive low hit rates, while rules that are morphological in nature, such as `suffix:ed:ing` (verb change from past/participle to present-continuous) and `suffix:y:ies` (pluralization of *y*–ending nouns), receive high hit rates.

[1] A maximum size of 6 is used in our experiments.

[2] At `code.google.com/p/word2vec`.

w_1	w_2	rank	cosine	transformation
create	created	0	0.58	suffix:ϵ:d:$\uparrow$dethrone
create	creates	0	0.65	suffix:te:tes:$\uparrow$evaluate
create	creates	1	0.62	suffix:ϵ:s:$\uparrow$contradict
created	create	0	0.65	suffix:ed:e$\uparrow$eroded
creation	create	0	0.52	suffix:ion:e:$\uparrow$communication
creation	created	0	0.54	suffix:ion:ed:$\uparrow$disruption
recreations	recreate	2	0.59	suffix:ions:e:$\uparrow$translations
recreations	recreating	1	0.53	suffix:ions:ing:$\uparrow$constructions
recreations	Recreations	81	0.64	prefix:r:R:$\uparrow$remediation

Table 2: Examples of lexicalized morphological transformations evaluated in E^n using rank and cosine.

Generate lexicalized morphological transformations

The results in Table 1 indicate the need for creating *lexicalized* transformations. For instance, rule suffix:ly:ϵ (drop suffix *ly*, a perfectly reasonable morphological transformation in English) is evaluated to have a hit rate of 32.1%. While such transformations are desirable, we want to avoid applying them when firing without yielding meaning-preserving results (the rest of 67.9%), e.g., for word-pair (*only, on*). We therefore create lexicalized transformations by restricting the rule application to the vocabulary subset of V which passes the meaning-preservation criterion.

The algorithm also computes best direction vectors $\uparrow d_w$ for each rule support set S_r. It greedily selects a direction vector $\uparrow d_{w_0}$ that explains (based on Equation 1) the most pairs in S_r. After subset $S_r^{w_0}$ is computed for direction vector $\uparrow d_{w_0}$, it applies recursively on set $S_r - S_r^{w_0}$. This yields a new best direction vector $\uparrow d_{w_1}$, and so on. The recursion stops when it finds a direction vector $\uparrow d_{w_k}$ that explains less than a predefined number of words (we used 10 in all the experiments from this paper).

We consider multiple direction vectors $\uparrow d_{w_i}$ because of the possibly-ambiguous nature of a morphological transformation. Consider rule suffix:ϵ:s, which can be applied to the noun *walk* to yield plural-noun *walks*; this case is modeled with a transformation like *walk* + $\uparrow d_{invention\_}$, since $\uparrow d_{invention\_}$=*inventions*$-$*invention* is a direction that our procedure deems to explain well noun pluralization; it can also be applied to the verb *walk* to yield the 3rd-person singular form of the verb, in which case it is modeled as *walk* + $\uparrow d_{enlist\_}$, since $\uparrow d_{enlist\_}$=*enlists*$-$*enlist* is a direction that our procedure deems to explain well 3rd-person singular verb forms. In that sense, our algorithm goes beyond proposing simple surface-level morphemes, with direction vectors encoding well-defined semantics for our morphological analysis.

Lexicalized rules enhanced with direction vectors are called *morphological transformations*. For each morphological transformation, we evaluate again how well it passes a proximity test in E^n for the words it applies to. As evaluation criteria, we use two instances of Eq 1, with F_E instantiated to rank$_E$ and cosine$_E$, respectively. We apply more stringent criteria in this second pass, using thresholds on the resulting rank (t_{rank}) and cosine (t_{cosine}) values to indicate meaning preservation (we used $t_{\text{rank}} = 30$ and $t_{\text{cosine}} = 0.5$ in all the experiments in this paper). We present in Table 2 a sample of the results of this procedure. For instance, word *create* can be transformed to *creates* using two different transformations: suffix:te:tes:$\uparrow$evaluate and suffix:ϵ:s:$\uparrow$contradict, passing the meaning-preservation criteria with rank=0, cosine=0.65, and rank=1, cosine=0.62, respectively.

Lexicalized morphological transformations over a vocabulary V have a graph-based interpretation: words represent nodes, transformations represent edges in a labeled, weighted, cyclic, directed multi-graph (weights are (r, c) pairs, rank and cosine values; multiple direction vectors create multiple edges between two nodes; cycles may exist, see

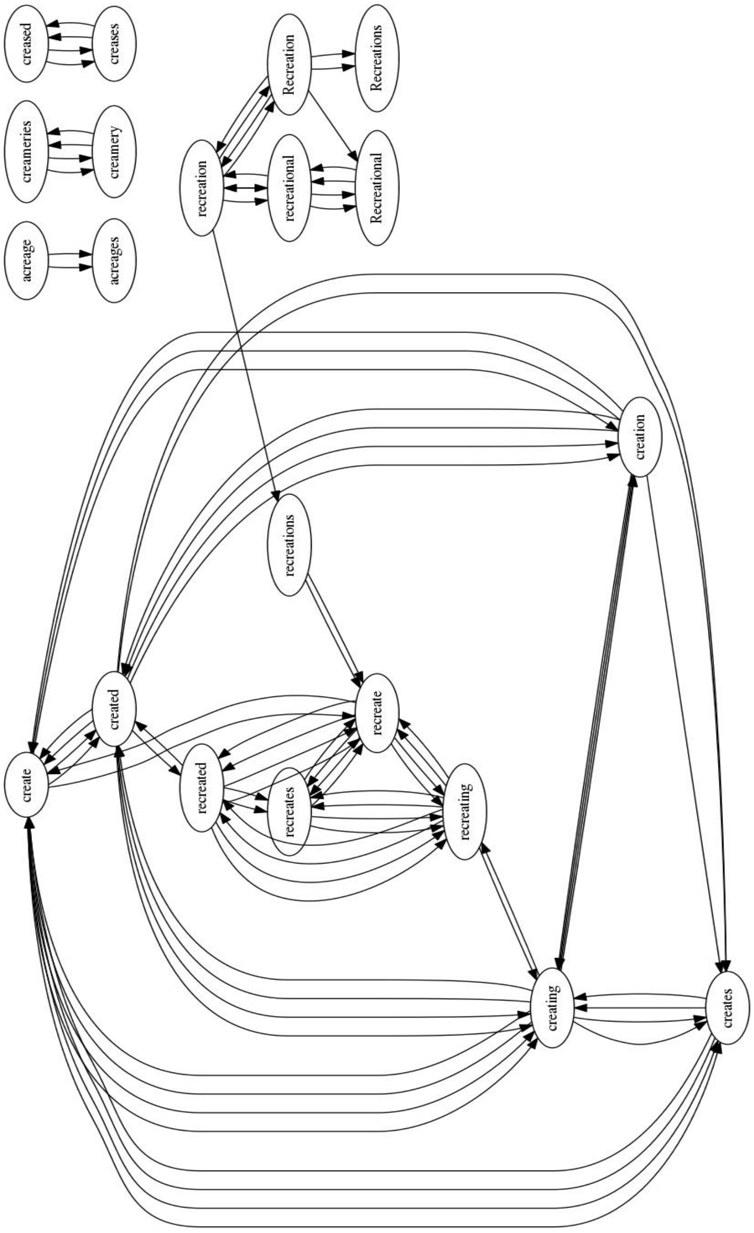

Figure 1: A few strongly connected components of a G^V_{Morph} graph for English.

e.g. *created→create→created* in Table 2). We use the notation G^V_{Morph} to denote such a graph. G^V_{Morph} usually contains many strongly connected components, with components representing families of morphological variations. As an illustration, we present in Figure 1 a few strongly connected components obtained for an English embedding space (for illustration purposes, we show only a maximum of 2 directed edges between any two nodes in this multigraph, even though more may exist).

3.2 Inducing 1-to-1 Morphological Mappings

The induced graph G^V_{Morph} encodes a lot of information about words and how they relate to each other. For some applications, however, we want to normalize away morphological diversity by mapping to a canonical surface form. This amounts to selecting, from among all the candidate morphological transformations generated, specific 1-to-1 mappings. In graph terms, this means building a labeled, weighted, acyclic, directed graph D^V_{Morph} starting

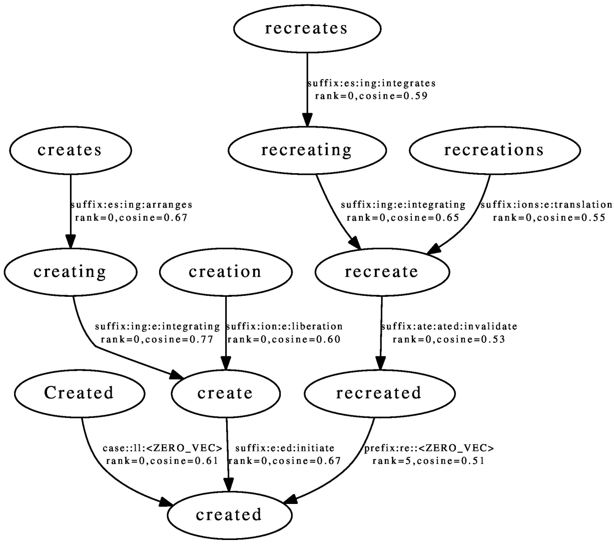

Figure 2: A part of a D^V_{Morph} graph, with the morphological family for the normal-form *created*.

from G^V_{Morph}, using the nodes from G^V_{Morph} and retaining only edges that meet certain criteria.

For the experiments presented in Section 4, we build a directed graph D^V_{Morph} as follows:

1. edge $w_1 \overset{(r,c)}{\rightarrow} w_2$ in G^V_{Morph} is considered only if count$(w_1) \leq$ count(w_2) in V;

2. if multiple such edges exist, chose the one with minimal rank r;

3. if multiple such edges still exist, chose the one with the maximal cosine c.

The interpretation we give is word-normalization: a normalization of w to w' is guaranteed to be meaning preserving (using the direction-vector semantics), and to a more frequent form. A snippet of the resulting graph D^V_{Morph} is presented in Figure 2.

One notable aspect of this normalization procedure is that these are not "traditional" morphological mappings, with morphology-inflected words mapped to their linguistic roots. Rather, our method produces morphological mappings that favor frequency over linguistic normalization. An example of this can be seen in Figure 2, where the root form *create* is morphologically-explained by mapping it to the form *created*. This choice is purely based on our desire to favor the accuracy of the word-representations for the normal forms; different choices regarding how this pruning procedure is performed lead to different normalization procedures, including some that are more linguistically-motivated (e.g., length-based).

3.3 Morphological Transformations for Rare and Unknown Words

For some count threshold C, we define $V_C = \{w \in V | C \leq \text{count}(w)\}$. The method we presented up to this point induces a morphology graph $D^{V_C}_{Morph}$ that can be used to perform morphological analysis for any words in V_C. We analyze the rest of the words we may encounter (i.e., rare words and OOVs) by mapping them directly to nodes in $D^{V_C}_{Morph}$.

We extract such mappings from $D^{V_C}_{Morph}$ using all the sequences of edges that start at nodes in the graph and end in a normal-form (i.e., nodes that have out-degree 0). The result is a set of rule sequences denoted *RS*. A count cutoff on the rule sequence counts is used, since low-count sequences tend to be less reliable (in the experiments reported in this paper we use a cutoff of 50). We also denote with *R* the set of all edges in D_{Morph}. Using sets *RS* and *R*, we map $w \notin V_C$ to a node $w' \in D^{V_C}_{Morph}$, as follows:

1. for rule-sequences $s \in RS$ from highest-to-lowest count, if $w \xrightarrow{s} w'$ and $w' \in D^{V_C}_{Morph}$, then s is the morphological analysis for w;

2. if no s is found, do breadth-first search in $D^{V_C}_{Morph}$ using $r \in R$, up to a predefined[3] depth d; for $k \leq d$, word w' with $w \xrightarrow{r_1 \ldots r_k} w' \in D^{V_C}_{Morph}$ and the highest count in V_C is the morphological analysis for w.

For example, this procedure uses the RS sequence s=$\texttt{prefix}:\texttt{un}:\epsilon, \texttt{suffix}:\texttt{ness}:\epsilon$ to perform the OOV morphological analysis *unassertiveness* $\xrightarrow{s}$*assertive*. We perform an in-depth analysis of the performance of this procedure in Section 4.2.

4 Empirical Results

In this section, we evaluate the performance of the procedure described in Section 3. Our evaluations aim at answering several empirical questions: how

[3] We use *d*=1 in the experiments reported in Section 4.2.

Lang	$\lvert \text{Tokens} \rvert$	$\lvert V \rvert$	$\lvert G^{V}_{Morph} \rvert$	$\lvert D^{V}_{Morph} \rvert$
EN	1.1b	1.2m	780k	75,823
DE	1.2b	2.9m	3.7m	169,017
FR	1.5b	1.2m	1.8m	92,145
ES	566m	941k	2.2m	82,379
RO	1.7b	963k	3.8m	141,642
AR	453m	624k	2.4m	114,246
UZ	850m	2.0m	5.6m	194,717

Table 3: Statistics regarding the size of the training data and the induced morphology graphs.

well does our method capture morphology, and how does it compare with previous approaches that use word-representations for morphology? How well does this method handle OOVs? How does the impact of morphology analysis change with training data size? We provide both qualitative and quantitative answers for each of these questions next.

4.1 Quality of Morphological Analysis

We first evaluate the impact of our morphological analysis on a standard word-similarity rating task. The task measures word-level understanding by comparing the correlation between human-produced similarity ratings for word pairs, e.g. (*intraspecific, interspecies*), with those produced by an algorithm. For the experiments reported here, we train SkipGram models[4] using a dimensionality of $n = 500$. We denote a system using only SkipGram model embeddings as SG. To evaluate the impact of our method, we perform morphological analysis for words below a count threshold C. For a word $w \in D^{V_C}_{Morph}$, we simply use the SkipGram vector-representation; for a word $w \notin D^{V_C}_{Morph}$, we use as word-representation its mapping in $D^{V_C}_{Morph}$; we denote such a system SG+Morph. For both SG and SG+Morph systems, we compute the similarity of word-pairs using the cosine distance between the vector-representations.

Data

We train both the SG and SG+Morph models from scratch, for all languages considered. For English, we use the Wikipedia data (Shaoul and Westbury, 2010). For German, French, and Spanish, we use the monolingual data released as part of the WMT-2013 shared task (Bojar et al., 2013). For Arabic we use the Arabic GigaWord corpus (Parker et al., 2011). For Romanian and Uzbek, we use collections of News harvested from the web and cleaned (boilerplate removed, formatting removed, encoding made consistent, etc.). All SkipGram models are trained using a count cutoff of 5 (all words with count less than the cutoff are ignored). Table 3 presents statistics on the data and vocabulary size, as well as the size of the induced morphology graphs. These numbers illustrate the richness of the morphological phenomena present in languages such as German, Romanian, Arabic, and Uzbek, compared to English.

As test sets, we use standard, publicly-available word-similarity datasets. Most relevant for our approach is the Stanford English Rare-Word (RW) dataset (Luong et al., 2013), consisting of 2034 word pairs with a higher degree of English morphology compared to other word-similarity datasets. We also use for English the WS353 (Finkelstein et al., 2002) and RG65 datasets (Rubenstein and Goodenough, 1965). For German, we use the Gur350 and ZG222 datasets (Zesch and Gurevych, 2006). For French we use the RG65 French version (Joubarne and Inkpen, 2011); for Spanish, Romanian, and Arabic we use their respective versions of WS353 (Hassan and Mihalcea, 2009).

Results

We present in Table 4 the results obtained across 6 language pairs and 9 datasets, using a count threshold for SG+Morph of $C = 100$. We also include the results obtained by two previously-proposed methods, LSM2013 (Luong et al., 2013) and BB2014 (Botha and Blunsom, 2014), which share some of the characteristics of our method.

Even in the absence of any morphological treatment, our word representations are better than previously used ones. For instance, LSM2013 uses exactly the same EN Wikipedia (Shaoul and Westbury, 2010) training data, and achieves 26.8 and 34.4 Spearman ρ correlation on RW, with and without morphological treatment, respectively. The word representations we train yield a ρ of 35.8 for SG, and a ρ of 41.8 for SG+Morph (+7.4 improve-

[4] Additional settings include a window-size of 5 and negative sampling set to 5. Unseen words receive a zero-vector embedding and a cosine score of 0.

	Spearman ρ								
Language	EN			DE		FR	ES	RO	AR
Testset	RW	WS	RG	Gur	ZG	RG	WS	WS	WS
System									
LSM2013 w/o morph	26.8	62.6	62.8	-	-	-	-	-	-
LSM2013 w/ morph	34.4	64.6	65.5	-	-	-	-	-	-
BB2014 w/o morph	18.0	32.0	47.0	36.0	6.0	33.0	26.0	-	-
BB2014 w/ morph	30.0	40.0	41.0	56.0	**25.0**	45.0	28.0	-	-
SG	35.8	71.2	75.1	62.4	16.6	63.6	36.5	51.7	37.1
SG+Morph	**41.8**	**71.2**	**75.1**	**64.1**	21.5	**67.3**	**47.3**	**53.1**	**43.1**
# pairs	2034	353	65	350	222	65	353	353	353

Table 4: Performance of previously proposed methods, compared to SG and SG+Morph trained on Wiki1b. LSM2013 uses exactly the same training data for EN, whereas BB2014 uses the same training data for DE, FR, ES.

ment under the morphology condition). The morphological treatment used by LSM2013 also has a small effect on the words present in the English WS and RG sets; our method does not propose any separate morphological treatment for the words in these datasets, since all of them have been observed more than our $C = 100$ threshold in the training data (therefore have reliable representations). The SG word-representations for all the other languages (German, French, Spanish, Romanian, and Arabic) also perform well on this task, with much higher Spearman scores obtained by SG compared with the previously-reported scores.

The results in Table 4 also show that our morphology treatment provides consistent gains across all languages considered. For morphologically-rich languages, all datasets reflect the impact of morphology treatment. We observe significant gains between the performance of the SG and SG+Morph systems, on top of the high correlation numbers of the SG system. For German, the relatively small increase we observe is due to the fact the German noun-compounds are not covered by our morphological treatment. For French, Spanish, Romanian, and Arabic, the gains by the SG+Morph support the conclusion that our method, while completely language-agnostic, handles well the variety of morphological phenomena present in these languages.

4.2 Quality of Morphological Analysis for Unknown/Rare Words

In this section, we quantify the accuracy of the morphological treatment for OOVs presented in Section 3.3. We assume that the statistics for unseen words (with respect to their morphological make-up) are similar with the statistics for low-frequency words. Therefore, for some relatively-low counts L and H, the set $V_{[L,H)} = V_L - V_H$ is a good proxy for the population of OOV words that we see at run-time. We evaluate OOV morphology as follows:

1. Run the procedure for morphology induction on V_L, resulting in $D^{V_L}_{Morph}$;
2. Run the procedure for morphology induction on V_H, resulting in $D^{V_H}_{Morph}$;
3. Apply OOV morphology using $D^{V_H}_{Morph}$ for each $w \in V_{[L,H]}$; evaluate resulting $w \rightarrow w'$ against reference $w \rightarrow w'_{ref}$ from $D^{V_L}_{Morph}$, as normal-form$(w') \equiv$ normal-form(w'_{ref}).

To make the analysis more revealing, we split the entries in $V_{[L,H)}$ in two: type T1 entries are those that have in-degree > 0 in $D^{V_L}_{Morph}$ (i.e., words that have a morphological mapping in the reference graph); type T2 entries are those that have 0 in-degree in $D^{V_L}_{Morph}$ (i.e., words with no morphological mapping in the reference, e.g., proper-nouns in English). Note that the T1/T2 distinction reflects a recall/precision trade-off: T1-words should be morphologically analyzed, while T2-words should not; a method that over-analyses has poor performance on T2, while one that under-analyses performs poorly on T1.

We use the same datasets as the ones presented in Section 4.1, see Table 3. The results for all the languages are shown in Table 6, with all rows using

	EN (RW testset)				DE (RG testset)			
	|Unmapped|		Spearman ρ		|Unmapped|		Spearman ρ	
	Wiki1b	News120b	Wiki1b	News120b	WMT2b	News20b	WMT2b	News20b
SG	80	177	35.8	44.7	0	20	62.4	62.1
SG+Morph	1	0	41.8	52.0	0	0	64.1	69.1

Table 5: Comparison between models SG and SG+Morph at different training-data sizes.

	$\lvert V_{[1000,2000)}\rvert$		Accuracy	
Lang	T1	T2	T1	T2
EN	3421	10617	89.7%	89.6%
DE	10778	21234	90.8%	93.1%
FR	6435	9807	90.3%	90.4%
ES	5724	7412	91.1%	90.3%
RO	11905	9254	86.5%	85.3%
AR	7913	5202	92.4%	69.0%
UZ	11772	9027	81.3%	84.1%

Table 6: Accuracy of Rare&OOV analysis.

Lang	|Tokens|	$\lvert V\rvert$	$\lvert G^V_{Morph}\rvert$	$\lvert D^V_{Morph}\rvert$
EN	120b	1.0m	2.9m	98,268
DE	20b	1.8m	6.7m	351,980

Table 7: Statistics for large training-data sizes.

the same setup. Count $L = 1000$ was chosen such that $D^{V_L}_{Morph}$ is reliable enough to be used as reference. The accuracy results are consistently high (in the 80-90% range) for both T1- and T2-words, even for morphologically-rich languages such as Uzbek. These results indicate that our method does well at both identifying a morphological analysis when appropriate, as well as not proposing one when not justified, and therefore provides accurate morphology analysis for rare and OOV words.

4.3 Morphology and Training Data Size

We also evaluate the impact of our morphology analysis under a regime with substantially more training data. To this end, we use large collections of English and German News, harvested from the web and cleaned (boiler-plate removed, formatting removed, encoding made consistent). Statistics regarding the resulting vocabularies and the induced morphology are presented in Table 7 (vocabulary cutoffs of 400 for EN and 50 for DE). We present results using the word-similarity task using the same Stanford Rare-Word (RW) dataset for EN and RG dataset for DE, compared against the setup using only 1-2 billion training tokens. For SG+Morph, we use count thresholds of 3000 for EN and 100 for DE. The results are given in Table 5. For English, a 100x increase in the training data for EN brings a 10-point increase in Spearman ρ (from 35.8 to 44.7, and from 41.8 to 52.0). The morphological analysis provides substantial gains at either level of training-data size: 6 points in ρ for Wiki1b (from 35.8 to 41.8), and 7.3 points for News120b EN (from 44.7 to 52.0). For German, the increase in training-data size does not bring visible improvements (perhaps due the high vocabulary cutoff), but the morphological treatment has a large impact under the large training-data condition (7 points for News20b DE, from 62.1 to 69.1).

5 Conclusions and Future Work

We have presented an unsupervised method for morphology induction. The method derives a morphological analyzer from scratch, and only requires a monolingual corpus for training, with no additional knowledge of the language. Our evaluation shows that this method performs well across a large variety of language families, and we present here results that improve on current state-of-the-art for the morphologically-rich Stanford Rare-word dataset.

We acknowledge that certain languages exhibit phenomena (such as word-compounds in German) that require a more focused approach for solving them. But techniques like the ones presented here have the potential to exploit vector-based word representations successfully to address such phenomena as well.

References

Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored neural language models. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, Baltimore, MD, USA, Volume 2: Short Papers*, pages 809–815.

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

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jan A. Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. *CoRR*.

Ronan Collobert. 2011. Deep learning for efficient discriminative parsing. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 224–232.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *TSLP*, 4(1).

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520.

Samer Hassan and Rada Mihalcea. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1201.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 873–882.

Fei Huang, Arun Ahuja, Doug Downey, Yi Yang, Yuhong Guo, and Alexander Yates. 2014. Learning representations for weakly supervised natural language processing tasks. *Computational Linguistics*, 40(1):85–120.

Colette Joubarne and Diana Inkpen. 2011. Comparison of semantic similarity for different languages using the google n-gram corpus and second-order co-occurrence measures. In *Advances in Artificial Intelligence - 24th Canadian Conference on Artificial Intelligence*, pages 216–221.

Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositionally derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1517–1526.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 302–308.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, Sofia, Bulgaria.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239.

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

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, pages 746–751.

Andriy Mnih and Geoffrey E. Hinton. 2007. Three new graphical models for statistical language modelling. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference*, pages 641–648.

Andriy Mnih, Zhang Yuecheng, and Geoffrey E. Hinton. 2009. Improving a statistical language model through non-linear prediction. *Neurocomputing*, 72(7-9):1414–1418.

Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maedaet. 2011. Arabic gigaword fifth edition ldc2011t11. In *Linguistic Data Consortium*, Philadelphia.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627633.

Cyrus Shaoul and Chris Westbury. 2010. The Westbury lab Wikipedia corpus.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *25th Annual Conference on Neural Information Processing Systems*, pages 801–809.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011b. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Torsten Zesch and Iryna Gurevych. 2006. Automatically creating datasets for measures of semantic relatedness. In *Workshop on Linguistic Distances*, pages 16–24.