# Joint Semantic Synthesis and Morphological Analysis of the Derived Word

**Ryan Cotterell**
Department of Computer Science
Johns Hopkins University
ryan.cotterell@jhu.edu

**Hinrich Schütze**
CIS
LMU Munich
inquiries@cislmu.org

## Abstract

Much like sentences are composed of words, words themselves are composed of smaller units. For example, the English word *questionably* can be analyzed as *question+able+ly*. However, this *structural* decomposition of the word does not directly give us a *semantic* representation of the word's meaning. Since morphology obeys the principle of compositionality, the semantics of the word can be systematically derived from the meaning of its parts. In this work, we propose a novel probabilistic model of word formation that captures both the *analysis* of a word $w$ into its constituent segments and the *synthesis* of the meaning of $w$ from the meanings of those segments. Our model jointly learns to *segment* words into morphemes and *compose* distributional semantic vectors of those morphemes. We experiment with the model on English CELEX data and German DErivBase (Zeller et al., 2013) data. We show that jointly modeling semantics increases both segmentation accuracy and morpheme $F_1$ by between 3% and 5%. Additionally, we investigate different models of vector composition, showing that recurrent neural networks yield an improvement over simple additive models. Finally, we study the degree to which the representations correspond to a linguist's notion of morphological productivity.

## 1 Introduction

In most languages, words decompose further into smaller units, termed morphemes. For example, the English word *questionably* can be analyzed as *question+able+ly*. This *structural* decomposition of the word, however, by itself is not a *semantic* representation of the word's meaning;[1] we further require an account of how to synthesize the meaning from the decomposition. Fortunately, words—just like phrases—to a large extent obey the principle of compositionality: the semantics of the word can be systematically derived from the meaning of its parts.[2] In this work, we propose a novel joint probabilistic model of word formation that captures both *structural decomposition* of a word $w$ into its constituent segments and the *synthesis* of $w$'s *meaning* from the meaning of those segments.

Morphological segmentation is a structured prediction task that seeks to break a word up into its constituent morphemes. The output segmentation has been shown to aid a diverse set of applications, such as automatic speech recognition (Afify et al., 2006), keyword spotting (Narasimhan et al., 2014), machine translation (Clifton and Sarkar, 2011) and parsing (Seeker and Çetinoğlu, 2015). In contrast to much of this prior work, we focus on *supervised* segmentation, i.e., we provide the model with gold segmentations during training time. Instead of *sur-*

[1] There are many different linguistic and computational theories for interpreting the structural decomposition of a word. For example, *un-* often signifies negation and its effect on semantics can then be modeled by theories based on logic. This work addresses the question of structural decomposition and semantic synthesis in the general framework of distributional semantics.

[2] Morphological research in theoretical and computational linguistics often focuses on noncompositional or less compositional phenomena—simply because compositional derivation poses fewer interesting research problems. It is also true that—just as many frequent multiword units are not completely compositional—many frequent derivations (e.g., *refusal*, *fitness*) are not completely compositional. An indication that nonlexicalized derivations are usually compositional is the fact that standard dictionaries like OUP editors (2010) list derivational affixes with their compositional meaning, without a hedge that they can also occur as part of only partially compositional forms. See also Haspelmath and Sims (2013), §5.3.6.

Transactions of the Association for Computational Linguistics, vol. 6, pp. 33–48, 2018. Action Editor: Regina Barzilay.
Submission batch: 5/2016; Revision batch: 10/2016; Published 1/2018.

*face* segmentation, our model performs *canonical* segmentation (Cotterell et al., 2016a; Cotterell et al., 2016b; Kann et al., 2016), i.e., it allows the induction of orthographic changes together with the segmentation, which is not typical. For the example *questionably*, our model can restore the deleted characters *le*, yielding the canonical segments *question*, *able* and *ly*. In this work, our primary contribution lies in the integration of continuous semantic vectors into supervised morphological segmentation—we present a joint model of morphological analysis and semantic synthesis at the word-level.

We experimentally investigate three novel aspects of our model.

- First, we show that jointly modeling continuous representations of the semantics of morphemes and words allows us to improve morphological analysis. On the English portion of CELEX (Baayen et al., 1993), we achieve a 5 point improvement in segmentation accuracy and a 3 point improvement in morpheme $F_1$. On the German DErivBase dataset we achieve a 3 point improvement in segmentation accuracy and a 3 point improvement in morpheme $F_1$.
- Second, we explore improved models of vector composition for synthesizing word meaning. We find a recurrent neural network improves over previously proposed additive models. Moreover, we find that more syntactically oriented vectors (Levy and Goldberg, 2014a) are better suited for morphology than bag-of-word (BOW) models.
- Finally, we explore the productivity of English derivational affixes in the context of distributional semantics.

## 2 Derivational Morphology

Two important goals of morphology, the linguistic study of the internal structure of words, are to describe the relation between different words in the lexicon and to decompose them into *morphemes*, the smallest linguistic unit bearing meaning. Morphology can be divided into two types: *inflectional* and *derivational*. Inflectional morphology is the set of processes through which the word form outwardly displays syntactic information, e.g., verb tense. It follows that an inflectional affix typically neither changes the part-of-speech (POS) nor the semantics of the word. For example, the English verb *to run* takes various forms: *run*, *runs*, *ran* and *running*, all of which convey "moving by foot quickly", but appear in complementary syntactic contexts.

Derivation deals with the formation of new words that have semantic shifts in meaning (often including POS) and is tightly intertwined with lexical semantics (Light, 1996). Consider the example of the English noun *discontentedness*, which is derived from the adjective *discontented*. It is true that both words share a close semantic relationship, but the transformation is clearly more than a simple inflectional marking of syntax. Indeed, we can go one step further and define a chain of words *content* $\mapsto$ *contented* $\mapsto$ *discontented* $\mapsto$ *discontentedness*.

In the computational literature, derivational morphology has received less attention than inflectional. There are, however, two bodies of work on derivation in computational linguistics. First, there is a series of papers that explore the relation between lexical semantics and derivation (Lazaridou et al., 2013; Zeller et al., 2014; Padó et al., 2015; Kisselew et al., 2015). All of these assume a gold morphological analysis and primarily focus on the effect of derivation on distributional semantics. The second body of work, e.g., the unsupervised morphological segmenter MORFESSOR (Creutz and Lagus, 2007), does not deal with semantics and makes *no distinction* between inflectional and derivational morphology.[3] Even though the boundary between inflectional and derivational morphology is a continuum rather than a rigid divide (Haspelmath and Sims, 2013), there is still the clear distinction that derivation changes meaning whereas inflection does not. Our goal in this paper is to develop an account of how the meaning of a word form can be computed jointly, combining these two lines of work.

**Productivity and Semantic Coherence.** We highlight two related issues in derivation that motivated the development of our model: productivity

[3] Narasimhan et al. (2015) also make no distinction between inflectional and derivational morphology, but their model is an exception in that it includes vector similarity as a semantic feature. See §5 for discussion.

and semantic coherence. Roughly, a *productive* affix is one that can still actively be employed to form new words in a language. For example, the English nominalizing affix *ness* (*red*↦*red*+*ness*) can be attached to just about any adjective, including novel forms. In contrast, the archaic English nominalizing affix *th* (*dear*↦*dear*+*th*, *heal*↦*heal*+*th*, *steal*↦*steal*+*th*) does not allow us to form new words such as *cheapth*. This is a crucial issue in derivational morphology since we would not in general want to analyze new words as having been formed from non-productive endings; e.g., we do not want to analyze *hearth* as *hear*+*th* (or *wugth* as *wug*+*th*). Relations such as those between *heal* and *health* are *lexicalized* since they no longer can be derived by productive processes (Bauer, 1983).

Under a generative treatment (Chomsky, 1965) of morphology, productivity becomes a central notion since a grammar needs to account for active word formation processes in the language (Aronoff, 1976). Defining productivity precisely, however, is tricky; Aronoff (1976) writes, *"one of the central mysteries of derivational morphology ... [is that] ... though many things are possible in morphology, some are more possible than others."* Nevertheless, speakers often have clear intuitions about which affixes in the language are productive.[4]

Related to productivity is the notion of *semantic coherence*. The principle of compositionality (Frege, 1892; Heim and Kratzer, 1998) applies to interpretation of words just as it does to phrases. Indeed, compositionality is often taken to be a signal for productivity (Aronoff, 1976). When deciding whether to further decompose a word, asking whether the parts sum up to the whole is often a good indicator. In the case of *questionably* ↦ *question*+*able*+*ly*, the compositional meaning is "in a manner that could be questioned", which corresponds to the meaning of the word. Contrast this with the word *unquiet*, which means "restless", rather than "not quiet" and the compound *blackmail*, which does not refer to a letter written in black ink.

The model we will describe in §3 is a *joint model of both semantic coherence and segmentation*; that is, an analysis is judged not only by character-level features, but also by the degree to which the word is semantically compositional. Implicit in such a treatment is the desire to only segment a word if the segmentation is derived from a productive process. While most prior work on morphological segmentation has not explicitly modeled productivity,[5] we believe, from a computational modeling perspective, segmenting only productive affixes is preferable. This is analogous to the modeling of phrase compositionality in embedding models, where it can be better to not further decompose noncompositional multiword units like named entities and idiomatic expressions; see, e.g., Mikolov et al. (2013b), Wang et al. (2014), Yin and Schütze (2015), Yaghoobzadeh and Schütze (2015), and Hashimoto and Tsuruoka (2016).[6]

In this paper, we refer to the semantic aspect of the model either as *semantic synthesis* or as *coherence*. These are two ways of looking at semantics that are related as follows. If the synthesis (i.e., composition) of the meaning of the derived form from the meaning of its parts is a regular application of the linguistic rules of derivation, then the meaning so constructed is coherent. These are the cases where a joint model is expected to be beneficial for both segmentation and interpretation.

## 3 A Joint Model

From an NLP perspective, canonical segmentation (Naradowsky and Goldwater, 2009; Cotterell et al., 2016b) is the task that seeks to algorithmically decompose a word into its *canonical* sequence of morphemes. It is a version of morphological segmentation that requires the learner to handle orthographic changes that take place during word formation. We believe this is a more natural formulation of morphological analysis—especially for the processing

[4] It is also important to distinguish productivity from *creativity*—a non-rule-governed form of word formation (Lyons, 1977). As an example of creativity, consider the creation of portmanteaux, e.g., *dramedy* and *soundscape*.

[5] Note that segmenters such as MORFESSOR utilize the principle of minimum description length, which implicitly encodes productivity, in order to guide segmentation.

[6] As a reviewer points out, productivity of an affix and semantic coherence of the words formed from it are not perfectly aligned. Nonproductive affixes can produce semantically coherent words, e.g., *warm*↦*warm*+*th*. Productive affixes can produce semantically incoherent words, e.g., *canny*↦*un*+*canny*. Again, this is analogous to multiword units. However, there is a strong correlation and our experiments show that relying on it gives good results.

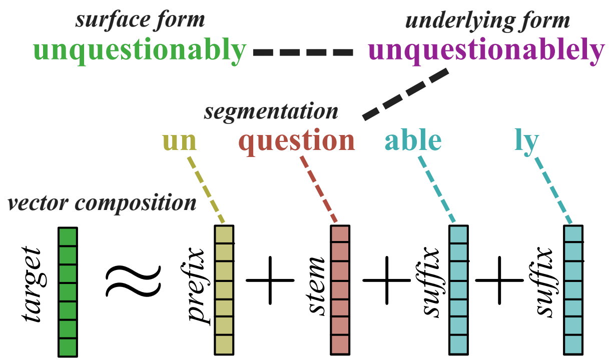

Figure 1: A depiction of the joint model that makes the relation between the three factors and the observed surface form explicit. We show a simple additive model of composition for ease of explication.

of derivational morphology—as it draws heavily on linguistic notions (see §2).

The main innovation we present is the augmentation of canonical segmentation to take into account semantic coherence and productivity. Consider the word *hypercuriosity* and its canonical segmentation *hyper+curious+ity*; this canonical segmentation seeks to decompose the word into its constituent morphemes and account for orthographic changes. This amounts to a *structural* decomposition of the word, i.e., how do we break up the string of characters into chunks? This is similar to the decomposition of a sentence into a parse tree. However, it is also natural to consider the *semantic* compositionality of a word, i.e., how is the meaning of the word synthesized from the meaning of the individual morphemes?

We consider both of these questions together in a single model, where we would like to place high probability on canonical segmentations that are also semantically coherent. Returning to *hypercuriosity*, we could further decompose it into *hyper+cure+ous+ity* in analogy to, say, *vice* ↦ *vicious*. Nothing about the surface form of *curious* alone gives us a strong cue that we should rule out the segmentation *cure+ous*. Turning to distributional semantics, however, it is the case that the contexts in which *curious* occurs are quite different from those in which *cure* occurs. This gives us a strong cue which segmentation is correct.

Formally, given a word string $w \in \Sigma^*$, where $\Sigma$ is a discrete alphabet of characters (in English this could be as simple as the 26 letter lowercase alphabet), and a word vector $v \in V$, where $V$ is a set of low-dimensional word embeddings, we define the model as:

$$p(v, s, l, u \mid w) = \frac{1}{Z_{\boldsymbol{\theta}}(w)} \exp\left(\frac{1}{2\sigma^2}||v - \mathcal{C}_{\boldsymbol{\beta}}(s,l)||_2^2 + \boldsymbol{f}(s,l,u)^\top \boldsymbol{\eta} + \boldsymbol{g}(u,w)^\top \boldsymbol{\omega}\right). \quad (1)$$

This model is composed of three factors: composition factor ($\frac{1}{2\sigma^2}||v - \mathcal{C}_{\boldsymbol{\beta}}(s,l)||_2^2$), segmentation factor $\boldsymbol{f}$ and transduction factor $\boldsymbol{g}$. The parameters of the model are $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\omega}\}$, the function $\mathcal{C}_{\boldsymbol{\beta}}$ composes morpheme vectors together, $s$ is the segmentation, $l$ is the labeling of the segments, $u$ is the underlying representation and $Z_{\boldsymbol{\theta}}(w)$ is the partition function. Note that the conditional distribution $p(v \mid s, l, u, w)$ is Gaussian distributed by construction. A visualization of our model is found in Figure 1. This model is a conditional random field (CRF) that is mixed, i.e., it is defined over both discrete and continuous random variables (Koller and Friedman, 2009). We restrict the range of $u$ to be a subset of $\Sigma^{|w|+k}$, where $k$ is an insertion limit (Dreyer, 2011). In this work, we take $k = 5$. Explicitly, the partition function is defined as

$$Z_{\boldsymbol{\theta}}(w) = \int \sum_{l', s', u'} \exp\left(\frac{1}{2\sigma^2}||v' - \mathcal{C}_{\boldsymbol{\beta}}(s', l')||_2^2 + \boldsymbol{f}(s', l', u')^\top \boldsymbol{\eta} + \boldsymbol{g}(u', w)^\top \boldsymbol{\omega}\right) dv', \quad (2)$$

which is guaranteed to be finite.[7]

A CRF is simply the globally renormalized product of several non-negative factors (Sutton and McCallum, 2006). Our model is composed of three: transduction, segmentation and composition factors—we describe each in turn.

### 3.1 Transduction Factor

The first factor we consider is the transduction factor: $\exp\left(\boldsymbol{g}(u,w)^\top \boldsymbol{\omega}\right)$, which scores a *surface*

[7] Since we have capped the insertion limit, we have a finite number of values that $u$ can take for any $w$. Thus, it follows that we have a finite number of canonical segmentations $s$. Hence we take a finite number of Gaussian integrals. These integrals all converge since we have fixed the covariance matrix as $\sigma^2 I$, which is positive definite.

*representation* (SR) $w$, the character string observed in raw text, and an *underlying representation* (UR), a character string with orthographic processes reversed. The aim of this factor is to place high weight on good pairs, e.g., the pair ($w$=*questionably*,$u$=*questionablely*), so we can accurately restore character-level changes.

We encode this portion of the model as a weighted finite-state machine for ease of computation. This factor generalizes probabilistic edit distance (Ristad and Yianilos, 1998) by looking at additional input and output context; see Cotterell et al. (2014) for details. As mentioned above and in contrast to Cotterell et al. (2014), we bound the insertion limit in the edit distance model.[8] Computing the score between two strings $u$ and $w$ requires a dynamic program that runs in $\mathcal{O}(|u|\cdot|w|)$. This is a generalization of the forward algorithm for Hidden Markov Models (HMMs) (Rabiner, 1989).

We employ standard feature templates for the task that look at features of edit operations, e.g., substitute $i$ for $y$, in varying context granularities. See Cotterell et al. (2016b) for details. Recent work has also explored weighting of WFST arcs with scores computed by LSTMs (Hochreiter and Schmidhuber, 1997), obviating the need for human selection of feature templates (Rastogi et al., 2016).

### 3.2 Segmentation Factor

The second factor is the segmentation factor: $\exp\left(\boldsymbol{f}(s,l,u)^\top\boldsymbol{\eta}\right)$. The goal of this factor is to score a segmentation $s$ of a UR $u$. In our example, it scores the input-output pair ($u$=*questionablely*, $s$=*question*+*able*+*ly*). It additionally scores a labeling of the segmentation. Our label set in this work is $L = \{\text{stem}, \text{prefix}, \text{suffix}\}$. The proper labeling of the segmentation above is $l$=*question:stem*+*able:suffix*+*ly:suffix*. The labeling is critical for our composition functions $\mathcal{C}_{\boldsymbol{\beta}}$ (Cotterell et al., 2015): which vectors are used depends on the label given to the segment; e.g., the vectors of the prefix "post" and the stem "post" are different.

We can view this factor as an unnormalized first-order semi-CRF (Sarawagi and Cohen, 2005). Computation of the factor again requires dynamic programming. The algorithm is a different generalization of the forward algorithm for HMMs, one that extends it to the semi-Markov case. This algorithm runs in $\mathcal{O}(|u|^2\cdot|L|^2)$.

| model | composition function |
|---|---|
| stem | $c = \sum_{i=1}^{N} \mathbb{1}_{l_i=\text{stem}} m_{s_i}^{l_i}$ |
| mult | $c = \bigodot_{i=1}^{N} m_{s_i}^{l_i}$ |
| add | $c = \sum_{i=1}^{N} m_{s_i}^{l_i}$ |
| wadd | $c = \sum_{i=1}^{N} \alpha_i m_{s_i}^{l_i}$ |
| fulladd | $c = \sum_{i=1}^{N} U_i m_{s_i}^{l_i}$ |
| LDS | $h_i = X h_{i-1} + U m_{s_i}^{l_i}$ |
| RNN | $h_i = \tanh(X h_{i-1} + U m_{s_i}^{l_i})$ |

Table 1: Composition models $\mathcal{C}_{\boldsymbol{\beta}}(s,l)$ used in this and prior work. The representation of the word is $h_N$ for the dynamic and $c$ for the non-dynamic models. Note that for the dynamic models $h_0$ is a learned parameter.

**Features.** We again use standard feature templates for the task. We create atomic indicator features for the individual segments. We then conjoin the atomic features with left and right context features as well as the label to create more complex feature templates. We also include transition features that fire on pairs of sequential labels. See Cotterell et al. (2015) for details. Recent work has also showed that a neural parameterization can remove the need for manual feature design (Kong et al., 2016).

### 3.3 Composition Factor

The composition factor takes the form of an unnormalized multivariate Gaussian density: $\exp\left(\frac{1}{2\sigma^2}||v - \mathcal{C}_{\boldsymbol{\beta}}(s,l)||_2^2\right)$, where the mean is computed by the (potentially non-linear) composition function (See Table 1) and the covariance matrix $\sigma^2 I$ is a diagonal matrix. The goal of the composition function $\mathcal{C}_{\boldsymbol{\beta}}(s,l)$ is to stitch together *morpheme* embeddings to approximate the vector of the entire word.

The simplest form of the composition function $\mathcal{C}_{\boldsymbol{\beta}}(s,l)$ is *add*, an additive model of the morphemes. See Table 1: each vector $m_{s_i}^{l_i}$ refers to a morpheme-

[8] As our transduction model is an unnormalized factor in a CRF, we do not require the local normalization discussed in Cotterell et al. (2014)—a weight on an edge may be any non-negative real number since we will renormalize later. The underlying model, however, remains the same.

specific, label-dependent embedding. If $l_i$ = stem, then $s_i$ represents a stem morpheme. Given that our segmentation is canonical, an $s_i$ that is a stem generally itself is an entry in the lexicon and $v(s_i) \in V$. If $v(s_i) \notin V$, then we set $v(s_i)$ to 0.[9] We optimize over vectors with $l_i \in \{\text{prefix}, \text{suffix}\}$ as they correspond to bound morphemes.

We also consider a more expressive composition model, a recurrent neural network (RNN). Let $N$ be the number of segments. Then $\mathcal{C}_{\boldsymbol{\beta}}(s, l) = h_N$ where $h_i$ is a hidden vector, defined by the recursion:[10] $h_i = \tanh\left(Xh_{i-1} + Um^{l_i}_{s_i}\right)$ (Elman, 1990). Again, we optimize the morpheme embeddings $m^{l_i}_{s_i}$ only when $l_i \neq$ stem along with the other parameters of the RNN, i.e., the matrices $U$ and $X$.

## 4 Inference and Learning

Exact inference is intractable since we allow arbitrary segment-level features on the canonicalized word forms $u$. Since the semi-CRF factor has features that fire on substrings, we would need a dynamic programming state for each substring of each of the exponentially many settings of $u$; this breaks the dynamic program. We thus turn to approximate inference through an importance sampling routine (Rubinstein and Kroese, 2011).

### 4.1 Inference by Importance Sampling

Rather than considering all underlying orthographic forms $u$ and segmentations $s$, we sample from a tractable proposal distribution $q$—a distribution over canonical segmentations. In the following equations we omit the dependence on $w$ for notational brevity and define $\boldsymbol{h}(l, s, u) = \boldsymbol{f}(s, l, u) + \boldsymbol{g}(u, w)$. Crucially, the partition function $Z_{\boldsymbol{\theta}}(w)$ is *not* a function of parameter subvector $\boldsymbol{\beta}$ and its gradient with respect to $\boldsymbol{\beta}$ is 0.[11] Recall that computing the gradient of the log-partition function is equivalent to the problem of marginal inference (Wainwright and Jordan, 2008). We derive our estimator as follows:

$$\nabla_{\boldsymbol{\theta}} \log Z = \mathop{\mathbb{E}}_{(l,s,u)\sim p} [\boldsymbol{h}(l, s, u)] \tag{3}$$

$$= \sum_{l,s,u} p(l, s, u)\boldsymbol{h}(l, s, u) \tag{4}$$

$$= \sum_{l,s,u} \frac{q(l, s, u)}{q(l, s, u)} p(l, s, u)\boldsymbol{h}(l, s, u) \tag{5}$$

$$= \mathop{\mathbb{E}}_{(l,s,u)\sim q} \left[\frac{p(l, s, u)}{q(l, s, u)}\boldsymbol{h}(l, s, u)\right], \tag{6}$$

where we have omitted the dependence on $w$ (which we condition on) and $v$ (which we marginalize out). So long as $q$ has support everywhere $p$ does (i.e., $p(l, s, u) > 0 \Rightarrow q(l, s, u) > 0$), the estimate is unbiased. Unfortunately, we can only efficiently compute $p(l, s, u)$ up to a constant factor, $p(l, s, u) = \bar{p}(l, s, u)/Z'_{\boldsymbol{\theta}}(w)$. Thus, we use the *indirect importance sampling estimator*,

$$\frac{1}{\sum_{i=1}^{M} w^{(i)}} \sum_{i=1}^{M} w^{(i)}\boldsymbol{h}(l^{(i)}, s^{(i)}, u^{(i)}), \tag{7}$$

where $(l^{(1)}, s^{(1)}, u^{(1)}) \ldots (l^{(M)}, s^{(M)}, u^{(M)}) \overset{\text{i.i.d.}}{\sim} q$ and importance weights $w^{(i)}$ are defined as:

$$w^{(i)} = \frac{\bar{p}(l^{(i)}, s^{(i)}, u^{(i)})}{q(l^{(i)}, s^{(i)}, u^{(i)})}. \tag{8}$$

This indirect estimator is biased, but consistent.[12]

**Proposal Distribution.** The success of importance sampling depends on the choice of a "good" proposal distribution, i.e., one that ideally is close to $p$. Since we are fully supervised at training time, we have the option of training locally normalized distributions for the individual components. Concretely, we train *two* proposal distributions $q_1(u \mid w)$ and $q_2(l, s \mid u)$ that take the form of a WFST and a semi-CRF, respectively, using features identical

[9] This is not changed in training, so all such $v(s_i)$ are 0 in the final model. Clearly, this could be improved in future work as a reviewer points out, e.g., by setting such $v(s_i)$ to an average of a suitable chosen set of known word vectors.

[10] We do not explore more complex RNNs, e.g., LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho et al., 2014a) as words in our data have ≤7 morphemes. These architectures make the learning of long distance dependencies easier, but are no more powerful than an Elman RNN, at least in theory. Note that perhaps if applied to languages with richer derivational morphology than English, considering more complex neural architectures would make sense.

[11] The subvector $\boldsymbol{\beta}$ is responsible for computing only the *mean* of the Gaussian factor and thus has no impact on its normalization coefficient (Murphy, 2012).

[12] Informally, the indirect importance sampling estimate converges to the *true* expectation as $M \to \infty$ (the definition of statistical consistency).

to the joint model. Each of these distributions is tractable—we can compute the marginals with dynamic programming and thus sample efficiently. To draw samples $(l, s, u) \sim q$, we sample sequentially from $q_1$ and then $q_2$, conditioned on the output of $q_1$.

### 4.2 Learning

We optimize the log-likelihood of the model using AdaGrad (Duchi et al., 2011), which is SGD with a special per-parameter learning rate. The full gradient of the objective for one training example is:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \log p(v, s, l, u \mid w) = {} & \boldsymbol{f}(s, l, u)^\top + \boldsymbol{g}(u, w)^\top \\ & - \frac{1}{\sigma^2}(v - \mathcal{C}_{\boldsymbol{\beta}}(s, l))\nabla_{\boldsymbol{\theta}}\mathcal{C}_{\boldsymbol{\beta}}(s, l) \\ & - \nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}(w), \qquad (9)\end{aligned}$$

where we use the importance sampling algorithm described in §4.1 to approximate the gradient of the log-partition function, following Bengio and Senecal (2003). Note that $\nabla_{\boldsymbol{\theta}}\mathcal{C}_{\boldsymbol{\beta}}(s, l)$ depends on the composition function used. In the most complicated case when $\mathcal{C}_{\boldsymbol{\beta}}$ is a RNN, we can compute $\nabla_{\boldsymbol{\beta}}\mathcal{C}_{\boldsymbol{\beta}}(s, l)$ efficiently with backpropagation through time (Werbos, 1990). We take $M = 10$ importance samples; using so few samples can lead to a poor estimate of the gradient, but for our application it suffices. We employ $L_2$ regularization.

### 4.3 Decoding

Decoding the model is also intractable. To approximate the solution, we again employ importance sampling. We take $M =$10,000 importance samples and select the highest weighted sample.

## 5 Related Work

The idea that vector semantics is useful for morphological segmentation is not new. Count vectors (Salton, 1971; Turney and Pantel, 2010) have been shown to be beneficial in the unsupervised induction of morphology (Schone and Jurafsky, 2000; Schone and Jurafsky, 2001). Embeddings were shown to act similarly (Soricut and Och, 2015). Our method differs from this line of research in two key ways. (i) We present a *probabilistic* model of the process of synthesizing the word's meaning from the meaning of its morphemes. Prior work was either not probabilistic or did not explicitly model morphemes. (ii) Our method is supervised and focuses on derivation. Schone and Jurafsky (2000) and Soricut and Och (2015), being fully unsupervised, do not distinguish between inflection and derivation and Schone and Jurafsky (2001) focus on inflection. More recently, Narasimhan et al. (2015) look at the unsupervised induction of "morphological chains" with semantic vectors as a crucial feature. Their goal is to jointly figure out an ordering of word formation and a morphological segmentation, e.g., *play*↦*playful*↦*playfulness*. While it is a rich model like ours, theirs differs in that it is unsupervised and uses vectors as features, rather than explicitly treating vector composition. All of the above work focuses on *surface segmentation* and not *canonical segmentation*, as we do.

A related line of work that has different goals concerns morphological generation. Two recent papers that address this problem using deep learning are Faruqui et al. (2016a) and Faruqui et al. (2016b). In an older line of work, Yarowsky and Wicentowski (2000) and Wicentowski (2002) exploit log frequency ratios of inflectionally related forms to tease apart that, e.g., the past tense of *sing* is not *singed*, but instead *sang*. Related work by Dreyer and Eisner (2011) uses a Dirichlet process to model a corpus as a "mixture of a paradigm", allowing for the semi-supervised incorporation of distributional semantics into a structured model of inflectional paradigm completion.

Our work is also related to recent attempts to integrate morphological knowledge into general embedding models. For example, Botha and Blunsom (2014) train a log-bilinear language model that models the composition of morphological structure. Likewise, Luong et al. (2013) train a recursive neural network (Goller and Küchler, 1996) over a heuristically derived tree structure to learn morphological composition over continuous vectors. Our work is different in that we learn a joint model of segmentation and composition. Moreover, supervised morphological analysis can drastically outperform unsupervised analysis (Ruokolainen et al., 2013).

Early work by Kay (1977) can be interpreted as finite-state canonical segmentation, but it neither addresses nor experimentally evaluates the question of joint modeling of morphological analysis and semantic synthesis. Moreover, we may view canoni-

|  |  | dev |  |  | test |  |  |
|---|---|---|---|---|---|---|---|
|  | Model | Acc | $F_1$ | Edit | Acc | $F_1$ | Edit |
| EN | Semi-CRF (Baseline) | 0.55 (.018) | 0.75 (.014) | 0.80 (.043) | 0.54 (.018) | 0.75 (.014) | 0.78 (.034) |
|  | Joint (Baseline) | 0.77 (.011) | 0.87 (.007) | 0.41 (.029) | 0.77 (.013) | 0.87 (.007) | 0.43 (.029) |
|  | Joint + Vec (This Work) | 0.83 (.014) | 0.91 (.008) | 0.31 (.019) | 0.82 (.020) | 0.90 (.011) | 0.32 (.038) |
|  | Joint + UR (Oracle) | 0.94 (.015) | 0.96 (.009) | 0.07 (.016) | 0.94 (.011) | 0.96 (.007) | 0.07 (.011) |
|  | Joint + UR + Vec (Oracle) | 0.95 (.011) | 0.97 (.007) | 0.05 (.013) | 0.95 (.023) | 0.97 (.006) | 0.05 (.025) |
| DE | Semi-CRF (Baseline) | 0.39 (.062) | 0.68 (.039) | 1.15 (.230) | 0.39 (.058) | 0.68 (.042) | 1.14 (.240) |
|  | Joint (Baseline) | 0.79 (.107) | 0.88 (.069) | 0.40 (.313) | 0.79 (.099) | 0.87 (.063) | 0.41 (.282) |
|  | Joint + Vec (This Work) | 0.82 (.102) | 0.90 (.067) | 0.33 (.312) | 0.82 (.096) | 0.90 (.061) | 0.33 (.282) |
|  | Joint + UR (Oracle) | 0.86 (.108) | 0.90 (.070) | 0.25 (.288) | 0.86 (.100) | 0.90 (.064) | 0.25 (.268) |
|  | Joint + UR + Vec (Oracle) | 0.87 (.106) | 0.92 (.069) | 0.20 (.285) | 0.88 (.096) | 0.93 (.062) | 0.19 (.263) |

Table 2: Results for the canonical morphological segmentation task on English and German. Standard deviation is given in parentheses. We compare against two baselines that do not make use of semantic vectors: (i) "Semi-CRF (baseline)", a semi-CRF that *cannot* account for orthographic changes and (ii) "Joint (Baseline)", a version of our joint model without vectors. We also compare against an oracle version with access to gold URs ("Joint + UR (Oracle)", "Joint + UR + Vec (Oracle)"), revealing that the toughest part of the canonical segmentation task is reversing the orthographic changes.

calization as an orthographic analogue to phonology. On this interpretation, the finite-state systems of Kaplan and Kay (1994), which computationally apply SPE-style phonological rules (Chomsky and Halle, 1968), may be run backwards to get canonical underlying forms.

## 6 Experiments and Results

We conduct experiments on English and German derivational morphology. We analyze our joint model's ability to segment words into their canonical morphemes as well as its ability to compositionally derive vectors for new words. Finally, we explore the relationship between distributional semantics and morphological productivity.

For English, we use the pretrained vectors of Levy and Goldberg (2014a) for all experiments. For German, we train word2vec skip-gram vectors on the German Wikipedia. We first describe our English dataset, the subset of the English portion of the CELEX lexical database (Baayen et al., 1993) that was selected by Lazaridou et al. (2013); the dataset contains 10,000 forms. This allows for comparison with previously proposed methods. We make two modifications. (i) Lazaridou et al. (2013) make the *two-morpheme assumption*: every word is composed of exactly two morphemes. In general, this is not true, so we further segment all complex words in the corpus. For example, *friendless+ness* is further segmented into *friend+less+ness*. To nevertheless allow for fair comparison, we provide versions of our experiments with and without the two-morpheme assumption where appropriate. (ii) Lazaridou et al. (2013) only provide a single train/test split. As we require a held-out development set for hyperparameter tuning, we randomly allocate a portion of the training data to select the hyperparameters and then retrain the model using these parameters on the original train split. We also report 10-fold cross validation results in addition to Lazaridou et al.'s train/test split.

Our German dataset is taken from Zeller et al. (2013) and is described in Cotterell et al. (2016b). It, again, consists of 10,000 derivational forms. We report results on 10-fold cross validation.

### 6.1 Experiment 1: Canonical Segmentation

For our first experiment, we test whether jointly modeling the continuous representations allows us to segment words more accurately. We assume that we are given an embedding for the target word. We estimate the model $p(v, s, l, u \mid w)$ as described in §4 with $L_2$ regularization $\lambda||\boldsymbol{\theta}||_2^2$. To evaluate, we decode the distribution $p(s, l, u \mid v, w)$. We perform approximate MAP inference with importance sampling—taking the sample with the highest score.

| | | EN | | | | | | DE | |
|---|---|---|---|---|---|---|---|---|---|
| | | BOW2 | | BOW5 | | DEPs | | SG | |
| | | dev | test | dev | test | dev | test | dev | test |
| oracle | stem | .403 | .402 | .374 | .376 | .422 | .422 | .400 | .405 |
| | add | .635 | .635 | .541 | .542 | .787 | .785 | .712 | .711 |
| | LDS | .660 | .660 | .566 | .568 | .806 | .804 | **.717** | **.718** |
| | RNN | .660 | .660 | .565 | .567 | **.807** | **.806** | .707 | .712 |
| joint | stem | .399 | .400 | .371 | .372 | .411 | .412 | .394 | .398 |
| | add | .625 | .625 | .524 | .525 | .782 | .781 | .705 | .704 |
| | LDS | .648 | .648 | .547 | .547 | .799 | .797 | **.712** | **.711** |
| | RNN | .649 | .647 | .547 | .546 | **.801** | **.799** | .706 | .708 |
| char | GRU | .586 | .585 | .452 | .452 | .769 | .768 | .675 | .667 |
| | LSTM | .586 | .586 | .455 | .455 | .768 | .767 | .677 | .666 |

Table 3: Vector approximation (measured by mean cosine similarity) both with ("oracle") and without ("joint", "char") gold morphology. Surprisingly, joint models are close in performance to models with gold morphology.

In these experiments, we use the RNN with the dependency vectors, the combination of which performs best on vector approximation in §6.2.

We follow the experimental design of Cotterell et al. (2016b). We compare against two baselines (marked "Baseline" in Table 2): (i) a "Semi-CRF" segmenter that cannot account for orthographic changes and (ii) the full "Joint" model of Cotterell et al. (2016b).[13] We additionally consider an "Oracle" setting, where we give the model the gold underlying orthographic form ("UR") at both training and test time. This gives us insight into the performance of the transduction factor of our model, i.e., how much could we benefit from a richer model.

Our hyperparameters are (i) the regularization coefficient $\lambda$ and (ii) $\sigma^2$, the variance of the Gaussian factor. We use grid search to tune them: $\lambda \in \{0.0, 10^1, 10^2, 10^3, 10^4, 10^5\}$, $\sigma^2 \in \{0.25, 0.5, 0.75, 1.0\}$.

**Metrics.** We use three metrics to evaluate segmentation accuracy. Note that the evaluation of canonical segmentation is hard since a system may return a sequence of morphemes whose concatenation is not the same length as the concatenation of the gold morphemes. This rules out metrics for surface segmentation like border $F_1$ (Kurimo et al., 2010), which require the strings to be of the same length.

We now define the metrics. (i) *Segmentation accuracy* measures whether every single canonical morpheme in the returned sequence is correct. It is inflexible: closer answers are penalized the same as more distant answers. (ii) *Morpheme* $F_1$ (van den Bosch and Daelemans, 1999) takes the predicted sequence of canonical morphemes, turns it into a set, computes precision and recall in the standard way and based on that then computes $F_1$. This metric gives credit if some of the canonical morphemes were correct. (iii) *Levenshtein distance* joins the canonical segments with a special symbol # into a single string and computes the Levenshtein distance between predicted and gold strings.

**Discussion.** Results in Table 2 show that jointly modeling semantic coherence improves our ability to analyze words. For test, our proposed joint model ("This Work") outperforms the baseline supervised canonical segmenter, which is state-of-the-art for the task, by .05 (resp. .03) on accuracy and .03 (resp. .03) on $F_1$ for English (resp. German). We also find that when we give the joint model an oracle UR the vectors generally help less: .01 (resp. .02) on accuracy and .01 (resp. .03) on $F_1$ for English (resp. German). This indicates that the chief boon the vector composition factor provides lies in selection of an appropriate UR. Moreover, the up to .15 difference in English between systems with and without the oracle UR suggests that reversing orthographic changes is a particularly difficult part of the task, at least for English.

## 6.2 Experiment 2: Vector Approximation

We adopt the experimental design of Lazaridou et al. (2013). Its aim is to approximate a vector of a derivationally complex word using a learned model of composition. As Lazaridou et al. (2013) assume a gold morphological analysis, we compare two settings: (i) oracle morphological analysis and (ii) inferred morphological analysis. To the best of our knowledge, (ii) is a novel experimental condition that no previous work has addressed.

We consider four composition models (See Table 1). (i) *stem*, using just the stem vector. This baseline tells us what happens if we make the incorrect assumption that derivation behaves like inflection and is not meaning-changing. (ii) *add*, a purely additive model. This is arguably the simplest way of combining the vectors of the morphemes. (iii) *LDS*, a linear dynamical system. This is arguably the simplest sequence model. (iv) A (simple) *RNN*. Recur-

[13] i.e., a model *without* the Gaussian factor that scores vectors.

| | | all | HR | LR | -less | in- | un- |
|---|---|---|---|---|---|---|---|
| Lazaridou | stem | .47 | .52 | .32 | .22 | .39 | .33 |
| | mult | .39 | .43 | .28 | .23 | .34 | .33 |
| | dil. | .48 | .53 | .33 | .30 | .45 | .41 |
| | wadd | .50 | .55 | .38 | .24 | .40 | .34 |
| | fulladd | .56 | .61 | .41 | .38 | .47 | .44 |
| BOW2 | lexfunc | .54 | .58 | .42 | .44 | .45 | .46 |
| | stem | .43 | .44 | .38 | .32 | .43 | .51 |
| | add | .65 | .67 | .61 | .60 | .64 | .67 |
| | LDS | .67 | .69 | .62 | .61 | .66 | .67 |
| | RNN | .67 | .69 | .60 | .60 | .65 | .66 |
| | c-GRU | .59 | .60 | .55 | .59 | .55 | .57 |
| | c-LSTM | .52 | .53 | .50 | .55 | .50 | .50 |
| BOW5 | stem | .40 | .43 | .33 | .27 | .37 | .46 |
| | add | .56 | .59 | .51 | .46 | .55 | .59 |
| | LDS | .58 | .61 | .51 | .48 | .57 | .60 |
| | RNN | .58 | .61 | .50 | .48 | .56 | .58 |
| | c-GRU | .45 | .47 | .42 | .42 | .43 | .45 |
| | c-LSTM | .46 | .47 | .43 | .43 | .45 | .46 |
| DEPs | stem | .46 | .45 | .49 | .38 | .57 | .67 |
| | add | .79 | .79 | .77 | .78 | .80 | .80 |
| | LDS | .80 | .81 | **.77** | **.79** | **.81** | **.81** |
| | RNN | **.81** | **.82** | **.77** | **.79** | .80 | **.81** |
| | c-GRU | .75 | .76 | .72 | .78 | .74 | .75 |
| | c-LSTM | .75 | .76 | .71 | .77 | .72 | .73 |

Table 4: Vector approximation (measured by mean cosine similarity) with gold morphology on the train/test split of Lazaridou et al. (2013). HR/LR = high/low-relatedness words. See Lazaridou et al. (2013) for details.

rent neural networks are currently the most widely used nonlinear sequence model and simple RNNs are the simplest such models.

Part of the motivation for considering a richer class of models lies in our removal of the two-morpheme assumption. Indeed, it is unclear that the *wadd* and *fulladd* models (Mitchell and Lapata, 2008) are useful models in the general case of multi-morphemic words—the weights are tied by *position*, i.e., the first morpheme's vector (be it a prefix or stem) is always multiplied by the same matrix.

**Comparison with Lazaridou et al.** To compare with Lazaridou et al. (2013), we use their exact train/test split. Those results are reported in Table 4. This dataset enforces that all words are composed of exactly two morphemes. Thus, a word like *unquestionably* is segmented as *un+questionably*, without further decomposition. The vectors employed by Lazaridou et al. (2013) are high-dimensional count vectors derived from lemmatized and POS tagged text with a before-and-after window of size 2. They then apply pointwise mutual information (PMI) weighting and dimensionality reduction by non-negative matrix factorization. In contrast, we employ WORD2VEC (Mikolov et al., 2013a), a model that is also interpretable as the factorization of a PMI matrix (Levy and Goldberg, 2014b). We consider three WORD2VEC models: two bag-of-word (BOW) models with before-and-after windows of size 2 and 5 and DEPs (Levy and Goldberg, 2014a), a dependency-based model whose context is derived from dependency parses rather than BOW.

In general, the results indicate that the key to better vector approximation is not a richer model of composition, but rather lies in the vectors themselves. We find that our best model, the RNN, only marginally edges out the LDS. Additionally, looking at the "all" column and the DEPs vectors, the simple additive model is only $\leq$.02 lower than LDS. In comparison, we observe large differences between the vectors. The RNN+DEPs model is .23 better than the BOW5 models (.81 vs. .58), .14 better than the BOW2 models (.81 vs. .67) and .25 better than Lazaridou et al.'s best model (.81 vs. .56). A wider context for BOW (5 instead of 2) yields worse results. This suggests that syntactic information or at least positional information is necessary for improved models of morpheme composition. The test vectors are annotated for relatedness, which is a proxy for semantic coherence. HR (high-relatedness) words were judged to be more compositional than LR (low-relatedness) words.

**Character-Level Neural Retrofitting.** As a further strong baseline, we consider a retrofitting (Faruqui et al., 2015) approach based on character-level recurrent neural networks. Recently, running a recurrent net over the character stream has become a popular way of incorporating subword information into a model—empirical gains have been observed in a diverse set of NLP tasks: POS tagging (dos Santos and Zadrozny, 2014; Ling et al., 2015), parsing (Ballesteros et al., 2015) and language modeling

(Kim et al., 2016). To the best of our knowledge, character-level retrofitting is a novel approach.

Given a vector $v$ for a word form $w$, we seek a function to minimize the following objective

$$\frac{1}{2}||v - h_N||_2^2, \tag{10}$$

where $h_N$ is the final hidden state of a recurrent neural architecture, i.e.,

$$h_i = \sigma(Ah_{i-1} + Bw_i), \tag{11}$$

where $\sigma$ is a non-linearity and $w_i$ is the $i^{\text{th}}$ character in $w$, $h_{i-1}$ is the previous hidden state and $A$ and $B$ are matrices. While we have defined the architecture for a vanilla RNN, we experiment with two more advanced recurrent architectures: GRUs (Cho et al., 2014b) and LSTMs (Hochreiter and Schmidhuber, 1997) as well as deep variants (Sutskever et al., 2014; Gillick et al., 2016; Firat et al., 2016). Importantly, this model has *no knowledge* of morphology—it can only rely on representations it extracts from the characters. This gives us a clear ablation on the benefit of adding structured morphological knowledge. We optimize the depth and the size of the hidden units on development data using a coarse-grained grid search. We found a depth of 2 and hidden units of size 100 (in both LSTM and GRU) performed best. We trained all models for 100 iterations of Adam (Kingma and Ba, 2015) with $L_2$ regularization with regularization coefficient 0.01.

Table 4 shows that the two character-level models ("c-GRU" and "c-LSTM") perform much worse than our models. This indicates that supervised morphological analysis produces higher-quality vector representations than "knowledge-poor" character-level models. However, we note that these character-level models have fewer parameters than our morpheme-level models—there are many more morphemes in a languages than characters.

**Oracle Morphology.** In general, the two-morpheme assumption is incorrect. We consider an expanded setting of Lazaridou et al. (2013)'s task, in which we fully decompose the word, e.g., *unquestionably*$\mapsto$*un*+*question*+*able*+*ly*. These results are reported in Table 3 (top block, "oracle"). We report mean cosine similarity. Standard deviations $s$ for 10-fold cross-validation (not shown) are small ($\leq$ .012) with two exceptions: $s = .044$ for the DEPs-joint-stem results (.411 and .412).

The multi-morphemic results mirror those of the bi-morphemic setting of Lazaridou et al. (2013). (i) RNN+DEPs attains an average cosine similarity of around .80 for English. Numbers for German are lower, around .70. (ii) The RNN only marginally edges out LDS for English and is slightly worse for German. Again, this is not surprising as we are modeling short sequences. (iii) Certain embeddings lend themselves more naturally to derivational compositionality: BOW2 is better than BOW5, DEPs is the clear winner.

**Inferred Morphology.** The final setting we consider is the vector approximation task without gold morphology. In this case, we rely on the full joint model $p(v, s, l, u \mid w)$. At evaluation, we are interested in the marginal distribution $p(v \mid w) = \sum_{s,l,u} p(v, s, l, u \mid w)$. We then use importance sampling to approximate the mean of this marginal distribution as the predicted embedding, i.e.,

$$\hat{v} = \int v p(v \mid w) dv \tag{12}$$

$$\approx \frac{1}{\sum_{i=1}^{M} w^{(i)}} \sum_{i=1}^{M} w^{(i)} \mathcal{C}_{\boldsymbol{\beta}}(l^{(i)}, s^{(i)}), \tag{13}$$

where $w^{(i)}$ are the importance weights defined in Equation 8 and $l^{(i)}$ and $s^{(i)}$ are the $i^{\text{th}}$ sampled labeling and segmentation, respectively.

**Discussion.** Surprisingly, Table 3 (joint) shows that relying on the inferred morphology does not drastically affect the results. Indeed, we are often within .01 of the result with gold morphology. Our method can be viewed as a retrofitting procedure (Faruqui et al., 2015), so this result is useful: it indicates that joint semantic synthesis and morphological analysis produces high-quality vectors.

### 6.3 Experiment 3: Derivational Productivity

We now delve into the relation between distributional semantics and morphological productivity. The extent to which jointly modeling semantics aids morphological analysis will be determined by the inherent compositionality of the words within the vector space. We break down our results on the vector approximation task with gold morphology using the

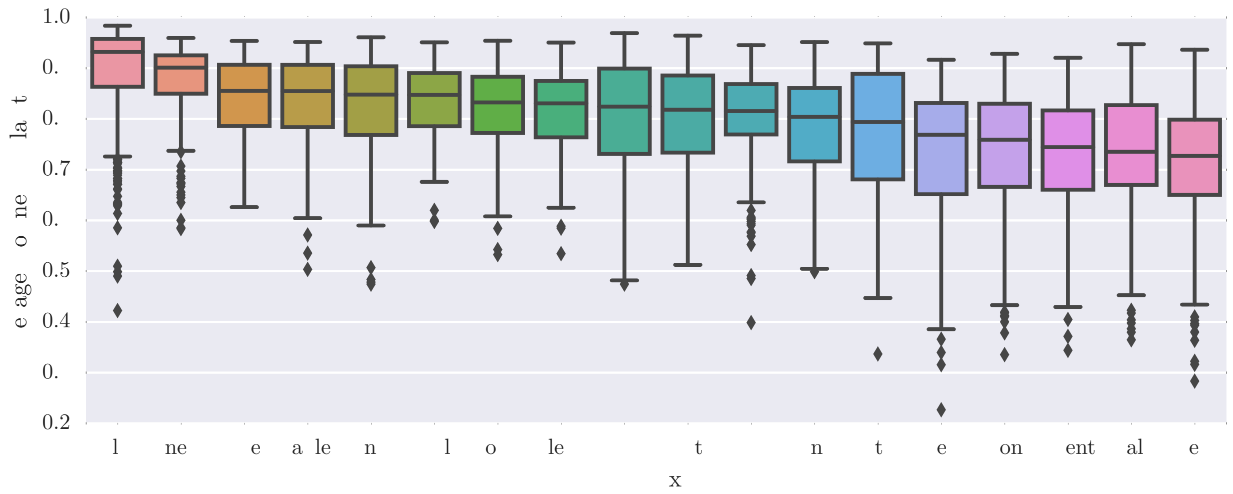

Figure 2: The boxplot breaks down the cosine similarity between the approximated vector and the target vector by affix (using gold morphology). We have ordered the affixes such that the better approximated vectors are on the left.

dependency vectors and the RNN composer in Figure 2 by selected affixes. We observe a wide range of scores: the most compositional ending *ly* gives rise to cosine similarities that are 20 points higher than those of the least compositional *er*.

On the left end of Figure 2 we see extremely productive suffixes. The affix *ize* is used productively with relatively obscure words in the sciences, e.g., *Rao-Blackwellize*. Likewise, the affix *ness* can be applied to almost any adjective without restriction, e.g., *Poissonness* 'degree to which data have a Poisson distribution'. On the right end, we find *-ment*, *-er* and *re-*. The affix *-ment* is borderline productive (Bauer, 1983)—modern English tends to form novel nominalizations with *ness* or *ity*. More interesting are *re-* and *er*, both of which are very productive in English. For *er*, many of the words bringing down the average are simply non-compositional. For example, *homer* 'homerun in baseball' is not derived from *home*+*er*—this is an error in data. We also see examples like *cutter*. It has a compositional reading (e.g., "box cutter"), but also frequently occurs in the non-compositional meaning 'type of boat'. Finally, proper nouns like *Homer* and *Turner* end in *er* and in our experiments we computed vectors for lowercased words. The affix *re-* similarly has a large number of non-compositional cases, e.g., *remove*, *relocate*, *remark*. Indeed, to get the compositional reading of *remove*, the first syllable (rather than the second) is typically stressed to emphasize the prefix.

We finally note several limitations of this experiment. (i) The ability of our models—even the recurrent neural network—to model transformations between vectors is limited. (ii) Our vectors are far from perfect; e.g., sparseness in the training data affects quality and some of the words in our corpus are rare. (iii) Semantic coherence is not the only criterion for productivity. An example is *-th* in English. As noted earlier, it is compositional in a word like *warmth*, but it cannot be used to form new words.

## 7 Conclusion

We have presented a model of the semantics and structure of derivationally complex words. To the best of our knowledge, this is the first attempt to jointly consider, within a single model, (i) the morphological decomposition of the word form and (ii) the semantic coherence of the resulting analysis. We found that directly modeling coherence increases segmentation accuracy, improving over a strong baseline. Also, our models show state-of-the-art performance on the derivational vector approximation task introduced by Lazaridou et al. (2013).

Future work will focus on the extension of the method to more complex instances of derivational morphology, e.g., compounding and reduplication, and on the extension to additional languages. We also plan to explore the relation between derivation and distributional semantics in greater detail.

**Acknowledgments.** The first author was supported by a DAAD Long-Term Research Grant and an NDSEG fellowship and the second by a Volkswagenstiftung Opus Magnum grant. We would also like to thank action editor Regina Barzilay for suggesting several changes we incorporated into the work and the three anonymous reviewers.

## References

Mohamed Afify, Ruhi Sarikaya, Hong-Kwang Jeff Kuo, Laurent Besacier, and Yuqing Gao. 2006. On the use of morphological analysis for dialectal Arabic speech recognition. In *Ninth International Conference on Spoken Language Processing*.

Mark Aronoff. 1976. *Word Formation in Generative Grammar*. MIT Press.

Harald Baayen, Richard Piepenbrock, and Hedderik van Rijn. 1993. The CELEX lexical data base on CD-ROM.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal, September. Association for Computational Linguistics.

Laurie Bauer. 1983. *English Word-Formation*. Cambridge University Press.

Yoshua Bengio and Jean-Sébastien Senecal. 2003. Quick training of probabilistic neural nets by importance sampling. In *Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics*.

Jan A. Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *International Conference on Machine Learning*, pages 1899–1907.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. *Workshop On Syntax, Semantics and Structure in Statistical Translation*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper & Row.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press.

Ann Clifton and Anoop Sarkar. 2011. Combining morpheme-based machine translation with postprocessing morpheme prediction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 32–42, Portland, Oregon, USA, June. Association for Computational Linguistics.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2014. Stochastic contextual edit distance and probabilistic FSTs. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 625–630, Baltimore, Maryland, June. Association for Computational Linguistics.

Ryan Cotterell, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-Markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 164–174, Beijing, China, July. Association for Computational Linguistics.

Ryan Cotterell, Arun Kumar, and Hinrich Schütze. 2016a. Morphological segmentation inside-out. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2325–2330, Austin, Texas, November. Association for Computational Linguistics.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016b. A joint model of orthography and morphological segmentation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 664–669, San Diego, California, June. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):3.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *International Conference on Machine Learning*, pages 1818–1826.

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 616–627. Association for Computational Linguistics, July.

Markus Dreyer. 2011. *A Non-parametric Model for the Discovery of Inflectional Paradigms from Plain Text using Graphical Models over Strings*. Ph.D. thesis, Johns Hopkins University.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, May–June. Association for Computational Linguistics.

Manaal Faruqui, Ryan McDonald, and Radu Soricut. 2016a. Morpho-syntactic lexicon generation using graph-based semi-supervised learning. *Transactions of the Association for Computational Linguistics*, 4:1–16.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016b. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, California, June. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California, June. Association for Computational Linguistics.

Gottlob Frege. 1892. Über Begriff und Gegenstand. *Vierteljahresschrift für wissenschaftliche Philosophie*, 16:192–205.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1306, San Diego, California, June. Association for Computational Linguistics.

Christoph Goller and Andreas Küchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *IEEE International Conference on Neural Networks*.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2016. Adaptive joint learning of compositional and non-compositional phrase embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 205–215, Berlin, Germany, August. Association for Computational Linguistics.

Martin Haspelmath and Andrea Sims. 2013. *Understanding morphology*. Routledge.

Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 961–967, Austin, Texas, November. Association for Computational Linguistics.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Martin Kay. 1977. Morphological and syntactic analysis. *Linguistic Structures Processing*, 5:131.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2741–2749.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Max Kisselew, Sebastian Padó, Alexis Palmer, and Jan Šnajder. 2015. Obtaining a better understanding of distributional models of German derivational morphology. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 58–63.

Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Lingpeng Kong, Chris Dyer, and Noah A Smith. 2016. Segmental recurrent neural networks. In *4th International Conference on Learning Representations*.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho Challenge competition 2005–2010: Evaluations and results. In *Special Interest Group on Computational Morphology and Phonology*.

Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1517–1526, Sofia, Bulgaria, August. Association for Computational Linguistics.

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

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

Marc Light. 1996. Morphological cues for lexical semantics. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 25–31, Santa Cruz, California, USA, June. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September. Association for Computational Linguistics.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria, August. Association for Computational Linguistics.

John Lyons. 1977. *Semantics*. Cambridge University Press.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *2th International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of Association of Computational Linguistics*, pages 236–244, Columbus, Ohio, June. Association for Computational Linguistics.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press.

Jason Naradowsky and Sharon Goldwater. 2009. Improving morphology induction by learning spelling rules. In *Twenty-first International Joint Conference on Artificial Intelligence*, pages 1531–1536.

Karthik Narasimhan, Damianos Karakos, Richard Schwartz, Stavros Tsakalidis, and Regina Barzilay. 2014. Morphological segmentation for keyword spotting. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 880–885, Doha, Qatar, October. Association for Computational Linguistics.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

OUP editors. 2010. *New Oxford American Dictionary*. Oxford University Press.

Sebastian Padó, Alexis Palmer, Max Kisselew, and Jan Šnajder. 2015. Measuring semantic content to assess asymmetry in derivation. In *Proceedings of the 11th International Conference on Computational Semantics*.

Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the Institute of Electrical and Electronics Engineers*, 77(2):257–286.

Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. 2016. Weighting finite-state transductions with neural context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 623–633, San Diego, California, June. Association for Computational Linguistics.

Eric S. Ristad and Peter N. Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Reuven Y. Rubinstein and Dirk P. Kroese. 2011. *Simulation and the Monte Carlo method*. John Wiley & Sons.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29–37, Sofia, Bulgaria, August. Association for Computational Linguistics.

Gerard Salton, editor. 1971. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice Hall.

Sunita Sarawagi and William W. Cohen. 2005. Semi-Markov conditional random fields for information extraction. In *Advances in neural information processing systems*, pages 1185–1192.

Patrick Schone and Daniel Jurafsky. 2000. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings the 4th Conference on Computational Natural Language Learning*, pages 67–72. Association for Computational Linguistics.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of the Second Meeting of the North American*

*Chapter of the Association for Computational Linguistics on Language Technologies*, pages 1–9. Association for Computational Linguistics.

Wolfgang Seeker and Özlem Çetinoğlu. 2015. A graph-based lattice dependency parser for joint morphological segmentation and syntactic analysis. *Transactions of the Association for Computational Linguistics*, 3:359–373.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, Denver, Colorado, May–June. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Charles Sutton and Andrew McCallum. 2006. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*, pages 93–128. MIT Press.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Antal van den Bosch and Walter Daelemans. 1999. Memory-based morphological analysis. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 285–292, College Park, Maryland, USA, June. Association for Computational Linguistics.

Martin J. Wainwright and Michael I. Jordan. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar, October. Association for Computational Linguistics.

Paul J. Werbos. 1990. Backpropagation through time: What it does and how to do it. *Proceedings of the Institute of Electrical and Electronics Engineers*, 78(10):1550–1560.

Richard Wicentowski. 2002. *Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework*. Ph.D. thesis, Johns Hopkins University.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, Lisbon, Portugal, September. Association for Computational Linguistics.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *The 38th Annual Meeting of the Association for Computational Linguistics*.

Wenpeng Yin and Hinrich Schütze. 2015. Convolutional neural network for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 901–911, Denver, Colorado, May–June. Association for Computational Linguistics.

Britta D. Zeller, Jan Šnajder, and Sebastian Padó. 2013. DErivBase: Inducing and evaluating a derivational morphology resource for German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1201–1211, Sofia, Bulgaria, August. Association for Computational Linguistics.

Britta D. Zeller, Sebastian Padó, and Jan Šnajder. 2014. Towards semantic validation of a derivational lexicon. In *Proceedings the 25th International Conference on Computational Linguistics*, pages 1728–1739, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.