Tabula nearly *rasa:* Probing the linguistic knowledge of character-level neural language models trained on unsegmented text

Anonymous TACL submission

Abstract

Recurrent neural networks (RNNs) recently reached striking performance in a variety of natural language processing tasks. This has renewed interest in whether these generic sequence processing devices are effectively capturing linguistic knowledge. Nearly all studies of this sort, however, initialize the RNNs with a vocabulary of known words, and feed them tokenized input during training. We present a multi-lingual study of the linguistic knowledge discovered by RNNs trained at the character level on input data with word boundaries removed. The networks, thus, face a tougher and more cognitively realistic task, having to discover all levels of the linguistic hierarchy from scratch. Results show that our "near *tabula rasa*" RNNs are implicitly encoding phonological, lexical, morphological, syntactic and semantic information, opening the door to intriguing speculations about the degree of prior knowledge that is necessary for successful language learning.

1 Introduction

Recurrent neural networks (RNNs, **?**), in particular in their Long-Short-Term-Memory variant (LSTMs, **?**), are the current workhorse of natural language processing. RNNs, often pre-trained on the simple *language modeling* objective of predicting the next symbol in natural text, are a crucial component of state-of-the-art architectures for machine translation, natural language inference and text categorization (**?**).

RNNs are very general devices for sequence processing, assuming little prior bias. Moreover, the simple prediction task they are trained on in language modeling seems well-aligned with the core role prediction plays in cognition (e.g., **??**). RNNs have thus long attracted researchers interested in language acquisition and processing. Their recent successes in realistic large-scale tasks has strongly rekindled this interest (see, e.g., **?????**, and references there).

The standard pre-processing pipeline of modern RNNs assumes that the input has been tokenized into word units that are pre-stored in the RNN vocabulary. This is a reasonable practical approach, but it makes simulations less interesting from a linguistic point of view. First, discovering words is one of the major challenges a learner faces, and by pre-encoding them in the RNN we are facilitating its task in an unnatural way (not even the staunchest nativists would take specific word dictionaries to be part of our genetic code). Second, assuming a unique tokenization into a finite number of discrete word units is in any case problematic. The very notion of what counts as a word in languages with a rich morphology is far from clear (e.g., **?**), and, universally, mental lexicons are probably organized into a not-necessarily-consistent hierarchy of units at different levels: morphemes, words, compounds, constructions, etc. (e.g., **?**).

Motivated by these considerations, we present here an extensive study of RNNs trained on language modeling at the character level, or *character-level neural language models* (CNLMs, **???**). Moreover, we trained the RNNs on input where whitespace has been removed, so that, like children learning a language, they don't have access to explicit cues to wordhood.[1] This setup is almost as *tabula rasa* as it goes. By using unsegmented orthographic input (and assuming that, in the alphabetic writing systems we work with, there is a reasonable correspondence between letters and phonetic segments), we are only postulating that the learner figured out how to segment the

[1] We do not erase punctuation marks, reasoning that they have a similar function to prosodic cues in spoken language.

continuous speech stream into phonological units, an ability children already possess few months after birth (e.g., **??**).

After training the networks on the unsupervised character-level language modeling task, we probe them with phonological, lexical, morphological, syntactic and semantic tests in English, German and Italian. Our results show that near-*tabula-rasa* CNLMs acquire an impressive spectrum of linguistic knowledge at various levels. This in turn suggests that, given abundant input (large Wikipedia dumps), a learning device whose only prior architectural bias consists in the LSTM memory cell implicitly acquires a variety of linguistic rules that one would intuitively expect to require much more prior knowledge.[2]

2 Related work

Character-based neural language models have received some attention in the last decade because of their greater generality, and because, intuitively, they should be able to use cues, such as morphological information, that word-based models miss by design. Early studies such as **?**, **?** and **?** established that CNLMs (trained with whitespace where relevant) are in general not as good at language modeling as their word-based counterparts, but lag only slightly behind (note that character-level sentence prediction involves a much larger search space than predicting at the word level). **?** and **?** presented informal qualitative analyses showing that CNLMs are learning basic linguistic properties of their input. The latter, who trained LSTM-based models, also showed that they can keep track, to some extent, of hierarchical structure. In particular, they are able to correctly balance parentheses when generating text.

Our aim here is to understand to what extent CNLMs trained on unsegmented input learn various linguistic constructs. This differs from most recent work in the area, that has focused on *character-aware* architectures combining character- and word-level information to develop state-of-the-art language models that are also effective in morphologically rich languages (see, e.g., **???**, and references there). For example, the influential model of Kim and colleagues performs prediction at the word level, but uses a character-based convolutional network to generate word representations. Other work focuses on segmenting words into morphemes with character-level RNNs (e.g., **?**), with emphasis on optimizing segmentation, as opposed to our interest in probing what the network implicitly learned about morphemes and other units through generic language modeling.

Probing linguistic knowledge of neural language models Extensive work probes the linguistic properties of word-based neural language models, as well as more complex architectures such as sequence-to-sequence systems: see, e.g., **?????????**.

Early work by Jeffrey Elman is close in spirit to ours. In particular, **?** reported phonotactics and word segmentation experiments similar to ours, but using toy inputs. More recently, **?** explored the grammatical properties of character- and subword-unit-level models that are used as components of a machine translation system. He concluded that current character-based decoders generalize better to unseen words, but capture less grammatical knowledge than subword units. Still, his character-based systems lagged only marginally behind the subword architectures on grammatical tasks such as handling agreement and negation. **?** also studied CNLMs with focus on understanding their properties, but only in the domain of sentiment analysis. **?** investigated the rules implicitly used by supervised character-aware neural morphological segmentation methods, finding in particular that the networks discover linguistically sensible patterns. More closely related to our goals, **?** probed the linguistic knowledge induced by a neural network that receives unsegmented acoustic input. They used however a considerably more complex architecture, trained on multimodal data, and they focused on phonology. **?** recently presented a related study probing the linguistic knowledge of plain character-level neural language models. Their results are aligned with ours, as they show that these models have knowledge of lexical and morphological structure, and they capture morphosyntactic categories as well as constraints on possible morpheme combinations. One of their most intriguing results is that the model tracks morpheme boundaries in a localist fashion through a single unit (we could not replicate the result with our model). They do not explore syntactic or semantic knowledge,

[2] Upon publication, we will make our input data, test sets and pre-trained model available.

and they limit their study to English. Moreover, they trained their models on input with whitespace, thus providing the model with a major (and cognitively artificial) cue to word boundaries.

3 Experimental setup

We extracted plain text from full English, German and Italian Wikipedia dumps with WikiExtractor.[3] We randomly extracted testing and validation sections consisting of 50,000 paragraphs each, and used the remainder for training. The training sets contained 16M (German), 9M (Italian), and 41M (English) paragraphs, corresponding to 819M, 463M and 2,333M words, respectively. Order of paragraphs was shuffled for training; we did not attempt to split by sentences. All characters were lower-cased. For word segmentation and word-based language models, we tokenized and tagged the corpora with TreeTagger.[4]

We used as vocabularies the most frequent characters from each corpus, setting thresholds so as to ensure that all characters representing phonemes were included, resulting in vocabularies of 60 (English), 73 (German), and 59 (Italian) characters. We further constructed *word-level neural language models* (WordNLMs); their vocabulary included the most frequent 50,000 words per corpus.

We trained RNN and LSTM CNLMs; we will refer to them simply as *RNN* and *LSTM*, respectively. We used LSTM cells for WordNLMs. For each model/language, we applied random hyperparameter search. We terminated training after 72 hours; none of the models had overfitted, as measured by performance on the validation set, used for model selection.[5]

Language modeling performance on the test partitions is shown in Table **??**. Recall that we removed whitespace, which is both easy to predict, and aids prediction of other characters. Consequently, the fact that our character-level models are below the state of the art is expected.[6] For example, the best model of **?** achieved 1.23 English BPC on a Wikipedia-derived dataset. On EuroParl data, **?** report 0.85 for English, 0.90 for German, and 0.82 for Italian. Still, our English BPC is comparable to that reported by **?** for his static character-level LSTM trained on space-delimited Wikipedia data, suggesting that we are achieving reasonable performance. The perplexity of the word-level model might not be comparable to that of highly-optimized state-of-the-art architectures, but it is at the expected level for a well-tuned vanilla LSTM language model. For example, **?** report 51.9 and 44.9 perplexities respectively in English and Italian for their best LSTMs trained on Wikipedia data with the same vocabulary size as ours.

	LSTM	*RNN*	*WordNLM*
English	1.62	2.08	48.99
German	1.51	1.83	37.96
Italian	1.47	1.97	42.02

Table 1: Performance of language models. For CNLMs, we report bits-per-character (BPC). For WordNLMs, we report perplexity.

4 Experiments

4.1 Phonological generalizations

Discovering phonological classes Are CNLMs discovering distributional generalizations about the phonological system of a language (as noisily reflected in the orthography)? We produced agglomerative clusterings of the trained LSTM input and output character embeddings. No clear pattern emerged from the input embeddings, whereas the output embeddings (probably more tuned to capture contextual dependencies) led to meaningful phonological classes in all languages. This is illustrated for German in Figure **??**. We see a basic split between vowels and consonants. Within the vowels, the front vowels *e* and *i* cluster together. Within the consonants, we observe a cluster of alveolar sonorants (*n, l, r*). The labial/labiodental *f, p, b, v, w* plus intruder *g* form a cluster. Symbols often denoting velar or palatal sounds, such as *c*, *k* and *q* cluster together. The *s* and *ß* letters, that, depending on context, denote the same or closely related sounds, cluster together. While the clustering does not reflect a single, consistent phonological dimension, it definitely suggests that the CNLM

[3] https://github.com/attardi/wikiextractor

[4] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

[5] Hyperparameter range and choices are in supplementary material to be made available upon publication. Chosen architectures (layers/embedding size/hidden size): LSTM: En. 3/200/1024, Ge. 2/100/1024, It. 2/200/1024; RNN: En. 2/200/2048, Ge. 2/50/2048, It. same; WordNLM; En. 2/1024/1024, Ge. 2/200/1024, It. same.

[6] Training our models with whitespace, without further hyperparameter tuning, resulted in BPCs of 1.32 (English), 1.28 (German), and 1.24 (Italian).

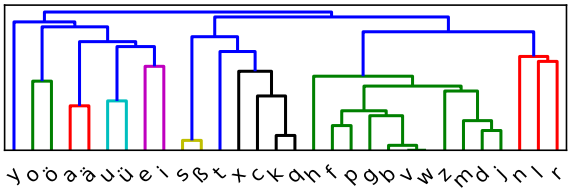

Figure 1: Clustering of German character embeddings (alphabetic characters only)

has discovered a fair deal about the features organizing the phonological system of the language.

Discovering phonotactic constraints Next, we study whether the CNLM is capturing phonotactic constraints more quantitatively. We focus on German and Italian, as they have reasonably transparent orthographies. We construct pairs of letter bigrams (picked to closely approximate phoneme bigrams) with the same first letter, but such that one is phonotactically acceptable in the language and the other isn't. We control letter frequencies such that the independent unigram probability of the unacceptable bigram is higher than that of the acceptable one. For example, "*br*" is an acceptable Italian sequence and "*bt*" isn't, although *t* is more frequent than *r*. For each such pair, we retrain the CNLMs on a version of the training partition from which all words containing either bigram have been removed. We then look at the likelihood the re-trained model assigns to both sequences. If the re-trained models systematically assign a larger probability to correct sequences, this provides evidence that the CNLM implicitly possesses a notion of phonological categories such as stops and sonorants, which allows it to correctly generalize from attested sequences (e.g., "*tr*") to unattested ones (*"br"*). In both languages, we constructed two groups of bigrams: In one, the valid bigram had a vowel following an obstruent; in the other, the obstruent was followed by a liquid. In both cases, in the invalid bigram, the obstruent was followed by a stop or nasal.

Results are shown in Table **??** for all bigram pairs (acceptable: left, impossible: right). The LSTM assigns higher probability to the acceptable bigrams in all but two cases. This confirms that it has learnt about phonological categories such as vowels and consonants and their phonotactic properties. The model makes the correct generalizations entirely on the basis of distributional evidence, with no aid from perceptual or articulatory cues. The RNN systematically prefers the impossible bigrams, presumably because they have higher unigram probability. The dramatic difference is surprising, since the relevant generalizations pertain to adjacent symbols that either model should capture. Possibly, although the tested rules are local, phonological categories are better learnt by a model that can extract generalizations about phoneme classes from wider contexts.

German				*Italian*			
		LSTM	*RNN*			*LSTM*	*RNN*
bu	bt	**4.6**	0.2	bu	bd	$\approx$ **1**	≈ 0
do	dd	**1.9**	0.1	du	dt	**1.3**	≈ 0
fu	ft	**6.5**	≈ 0	fu	ft	**30.5**	≈ 0
po	pt	**6.4**	0.1	pu	pt	**6.8**	≈ 0
tu	tt	**5.4**	≈ 0	tu	td	0.2	≈ 0
zu	zt	**2.4**	0.2	vu	vd	**2.0**	≈ 0
bl	bd	0.8	0.2	zu	zt	**55.7**	≈ 0
fl	fd	**2.1**	0.8	br	bt	$\approx$ **1**	≈ 0
fr	fn	**2.7**	0.1	dr	dt	**2.5**	0.4
kl	kt	**3.8**	0.1	fr	ft	**2.9**	≈ 0
pl	pt	**2.5**	0.9	pr	pt	**5.0**	≈ 0
AM		**3.6**	0.2	AM		**10.7**	≈ 0
GM		**3.0**	0.1	GM		**3.2**	≈ 0

Table 2: Likelihood ratio between acceptable and unacceptable bigrams, with arithmetic (AM) and geometric (GM) means. Values > 1 in bold.

4.2 Word segmentation

We tested whether our CNLM developed an implicit notion of word, despite not being endowed with a hard-coded word dictionary, and being exposed to unsegmented input. Early work on word segmentation has shown that low transition probabilities (**??**), high uncertainty about the next character (**??**) and low mutual information (**?**) serve as statistical cues to word segmentation. Based on these considerations, we tested the model segmentation capabilities as follows. We used the development sets to train logistic classifiers predicting whether a character is first in a word or not, based on the following features, derived from the pretrained CNLMs without further tuning: (1) *surprisal*, the log-probability of the character given prior context, (2) *entropy* of the distribution over the character given prior context, (3) *context PMI*, that is, the total likelihood of the next 20 characters, minus the unconditional likelihood estimated by starting the CNLM at the current position. We collected these quantities for each position and the preceding and following three characters, result-

	LSTM	*RNN*	*8-grams*
English	66/60/63	63/60/61	56/51/53
German	57/52/55	53/49/51	43/36/39
Italian	64/57/60	62/57/60	48/40/44

Table 3: Percentage precision, recall, and F1 on test set word segmentation.

	LSTM	Bayesian
Tokens	75.3/76.6/76.0	74.9/69.8/72.3
Lexical	41.2/61.2/49.2	63.6/60.2/61.9
Boundaries	91.3/90.0/90.5	93.0/86.7/89.8

Table 4: Word segmentation results (percentage precision/recall/F1) on our test partition of the Brent corpus for our CNLM-based model and the Bayesian approach of **?**. Following them, we evaluate at the level of tokens, the lexicon of induced word types, and boundaries.

ing in a 21-feature classifier. We repeated the experiment with features extracted from a character-level 8-gram model estimated on the training set, closer to earlier non-neural work (**??**).

Results are in Table **??**. The CNLM-based classifiers robustly segment more than half of the tokens correctly, and do considerably better than the 8-gram model, with a slight edge for the LSTM.

How does the LSTM compare to *ad-hoc* word segmentation models? We look at the Bayesian bigram model of **?**, an elegant approach using a hierarchical Dirichlet process. The latter, unlike our method, is unsupervised, but it has a specifically designed built-in bias towards a discrete lexicon with a power-law frequency distribution. Note that, while supervised, our model is rather parameter-lean, consisting in a logistic classifier trained on 21 features.

Running Bayesian methods on Wikipedia dumps is computationally unfeasible. We retrained instead the LSTM (with fixed hyperparameters) on the Brent corpus of English child-directed speech (**?**) also used by Goldwater and colleagues. We used 90% to train our language model, 5% to fit the logistic classifier, and 5% for evaluating both the classifier and the Bayesian model on word segmentation. The Bayesian model is trained on the full data-set, as it does not rely on word boundary information during training. Results in Table **??** show that the CNLM performance is comparable to that of the sophisticated Bayesian segmentation method.

We looked at common errors made by the English CNLM-based segmenter. Considering first the 30 most common undersegmentations in the test set (that is, cases in which the model failed to split two or more words): About half (16) are function word sequences that could reasonably be re-analyzed as single words (e.g., *more than*, *as well as*, *such as*). Of the remaining cases, 8 follow the *N of* pattern, where *N* is a (typically relational) noun commonly occurring in this construction (*member of*, *end of*, *part of*...). There are 3 fixed multi-word expressions (*New York*, *United States* and *high school*). The final undersegmentations *based on*, *known as* and *according to* can be seen as lexicalized connectives, especially in the Wikipedia text the model was trained on.

The picture is murkier but still fairly linguistically grounded for the 30 most common oversegmentation errors (that is, character fragments that are wrongly segmented from inside the largest number of distinct words).[7] More than half (17) are common affixes (prefixes such as *re* and *de* or suffixes such as *ing* and *ly*). 3 strings identical to frequent function words were wrongly carved out of longer words (*the*, *to* and *on*). The strings *land* and *man* are not unreasonably segmented out of compounds. It's hard to find a linguistically sound motivation for the 8 remaining top oversegmentations, that are, intriguingly, all CV syllables (*la, le, ma, na, ra, ro, se, ta*).

Interestingly, the CNLM-generated cues we used for word segmentation also cue constituents larger than words. To illustrate this, we created constituency trees for the German validation set using the Berkeley Parser (**?**). For each character in the data, we counted its hierarchical distance from the preceding character, operationalized as the number of intervening closing and opening brackets. This number is zero if both characters belong to the same word, 1 at word boundaries, larger at larger-constituent boundaries. Figure **??** plots CNLM-based PMI by hierarchical distance, for all distances for which at least 1,000 data-points occurred in the data-set. The plot shows that longer hierarchical distance between neighboring characters correspond to lower average PMI, generalizing the finding for word boundaries. This illustrates how it is useful for segmentation knowledge to be implicit, as the model can discover about different kinds of boundaries in a continuous manner.

[7] We ignore here single-letter segmentations, that would otherwise account for one third of the most-frequent set.

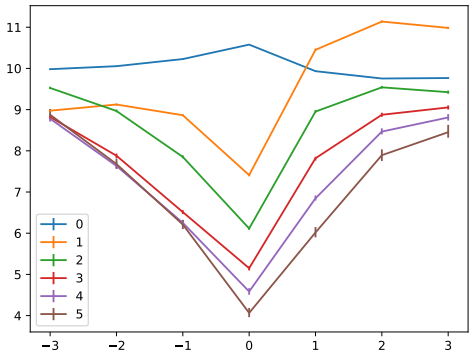

Figure 2: PMI between left and right contexts, as estimated by the LSTM CNLM in German, organized by syntactic hierarchical distance between subsequent characters (with bootstrapped 95 % confidence intervals).

4.3 Discovering morphological categories

Besides being sensitive, to some extent, to word boundaries, does the CNLM also store linguistic properties of words, such as their part of speech and number? These experiments focus on German and Italian, as it's harder to design reliable test sets for morphologically impoverished English.

Word classes (nouns vs. verbs) We sampled 500 verbs and 500 nouns from the Wikipedia training sets, requiring that they end in *-en* (German) or *-re* (Italian) (so that models can't rely on the affix for classification), and that they are unambiguously tagged in the corpus. We randomly selected 20 training examples (10 nouns and 10 verbs), and tested on the remaining items. We repeated the experiment 100 times to control for random train-test split variation. We recorded the final hidden state of a pre-trained CNLM after reading a word, without context, and trained a logistic noun-verb classifier on these representations.

As a baseline, we used a character-level LSTM autoencoder trained to reconstruct words in isolation. The hidden state of the autoencoder should capture relevant orthographic features. We further considered word embeddings from the output layer of the WordNLM, reporting its test accuracy both when OOV words are ignored and when they are randomly classified.

Results are shown in Table **??**. All language models outperform the autoencoders, showing that they learned categories based on broader distributional evidence, not just typical strings cuing nouns and verbs. Moreover, the LSTM CNLM outperforms the RNN, probably because it can track broader contexts. Not surprisingly, the word-based model fares better, but the gap, especially in Italian, is rather narrow, and there is a strong negative impact of OOV words.

	German	*Italian*
LSTM	89.0 (± 0.14)	95.0 (± 0.10)
RNN	82.0 (± 0.64)	91.9 (± 0.24)
Autoencoder	65.1 (± 0.22)	82.8 (± 0.26)
WordNLM$_{subs.}$	97.4 (± 0.05)	96.0 (± 0.06)
WordNLM	53.5 (± 0.18)	62.5 (± 0.26)

Table 5: Word class accuracy, with standard errors. '*subs.*' marks in-vocabulary subset evaluation.

Number We turn next to number, a more granular morphological feature. We study German as it possesses nominal classes that form plural through different morphological processes. We train a number classifier on a subset of these classes, and test on the others. If a model generalizes correctly, it means that it is sensitive to number as an abstract feature, independently of its surface expression.

We extracted plural nouns from the German UD treebank (**??**). We selected nouns with plurals in *-n*, *-s*, or *-e* to train the classifier (e.g., *Geschichte-n* 'stories'), and tested on plurals formed with *-r* or through vowel change (*Umlaut*, e.g., *Töchter* for singular *Tochter* 'daughter').

For the training set, we randomly selected 15 singulars and plurals from each training class. As plural suffixes make words longer, we sampled singulars and plurals from a single distribution over lengths, to ensure that their lengths were approximately matched. For the test set, we selected all plurals in *-r* (127) or Umlaut (38), with their respective singulars. We also used all remaining plurals ending in *-n* (1467), *-s* (98) and *-e* (832) as in-domain test data. To control for the impact of training sample selection, we report accuracies averaged over 200 repetitions. We extract word representations as above, and we compare to an autoencoder and embeddings from the WordNLM. As before, we report results ignoring OOV words, and with random classification for OOV words. Results are summarized in Table **??**.

The classifier based on word embeddings is the most successful, confirming that the latter reliably encode number (**?**). CNLM encodings outperform the autoencoder on plurals formed with suffixes, indicating some capability to detect number beyond orthographic cues. For *-r* plurals, the CNLM LSTM even outperforms the WordNLM. In con-

	train classes	test classes	
	-n/-s/-e	*-r*	*Umlaut*
LSTM	77.9 (± 0.8)	88.2 (± 0.3)	52.8 (± 0.6)
RNN	70.3 (± 0.9)	81.3 (± 0.7)	53.3 (± 0.6)
Autoencoder	64.0 (± 1.0)	73.8 (± 0.6)	59.2 (± 0.5)
WordNLM$_{subs.}$	97.8 (± 0.3)	86.6 (± 0.2)	96.7 (± 0.2)
WordNLM	82.1 (± 0.1)	73.1 (± 0.1)	77.6 (± 0.1)

Table 6: German number classification accuracy, with standard errors computed from 200 runs.

trast, the CNLMs do not generalize to Umlaut plurals, where they are virtually at chance level, and *worse* than the autoencoder. Evidently, CNLM number encoding is not abstract enough to generalize across very different surface morphological processes (adding a suffix vs. changing the root vowel).

4.4 Capturing syntactic dependencies

We take a further step up the linguistic hierarchy, probing CNLMs for their ability to capture syntactic dependencies between non-adjacent words. We again focus on German and Italian, due to their rich inflectional morphology, which makes it easier to construct controlled evaluation sets.

4.4.1 German

Gender agreement Each German noun belongs to one of three genders (masculine, feminine, neuter), morphologically marked on the article. As the article and the noun can be separated by adjectives and adverbs, we can probe knowledge of nouns' lexical gender together with long-distance agreement. We create stimuli of the form

(1) {der, die, das} sehr rote Baum
the very red tree

where the correct nominative singular article (*der*, in this case) matches the gender of the noun. We then run the CNLM on the three versions of this phrase (removing whitespace) and record the probabilities it assigns to them. If the model assigns the highest probability to the version with the right article, we count it as a hit for the model. To avoid phrase segmentation ambiguities, we present phrases surrounded by full stops.

We select all nominative singular nouns from the German UD treebank. We construct four conditions varying the number of adverbs and adjectives between article and noun. We first consider stimuli where no material intervenes. In the second condition, an adjective with the correct (nominative singular) case ending, randomly selected from the training corpus, is added. Crucially, the ending of the adjective does not reveal the gender of the noun. In the third and fourth conditions, one (*sehr*) or two adverbs (*sehr extrem*) intervene between the article and the adjective. These do not cue gender either. In each condition, we obtained 2290 (m.), 2261 (f.), and 1111 (n.) stimuli.

We constructed an n-gram baseline that picks the article occurring most frequently before the phrase in the training data, choosing randomly in case of ties. Here and below, when running the WordNLM, we excluded OOV nouns, resulting in a slightly easier test for this rival model. However, testing the CNLMs on the reduced set only led to slight improvements, that we do not report here.

Results are presented in Figure **??** (left). WordNLM performs best, followed by the LSTM CNLM. While the n-gram baseline performs similarly to the CNLM when there is no intervening material, accuracy drops to chance level (0.33) in the presence of an adjective. This problem would not be mitigated by interpolation with or backoff to lower-order n-grams, as the relevant gender information is present only on the first and last word of each stimulus. We conclude that, while direct association between articles and nouns can be learnt from simple corpus statistics, the CNLM has some capability to preserve the relevant information across more than a dozen timesteps. The RNN CNLM is much worse than the LSTM counterpart and even the n-gram model for the adjacent context, and its accuracy drops to random as more material intervenes, further confirming the importance of storing long-distance information. Note that, at the character level, even "adjacent" agreement requires carrying information through multiple time steps (the agreement violation will not emerge until enough characters of the noun have been processed to disambiguate its gender with respect to its prefix-sharing cohort).

Case agreement To test the model's knowledge of case agreement between articles and nouns, we selected the two determiners *dem* and *des*, which unambiguously indicate dative and genitive case, respectively, for masculine and neuter nouns:

(2) a. {dem, des} sehr roten Baum
the very red tree *(dative)*
b. {dem, des} sehr roten Baums
the very red tree *(genitive)*

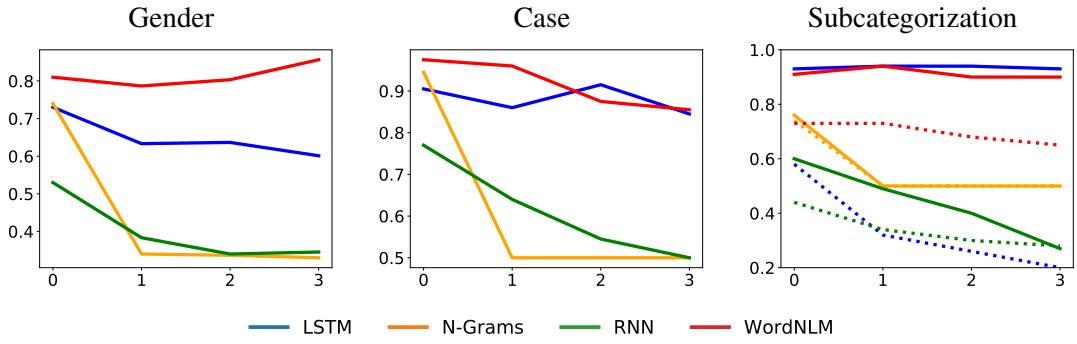

Figure 3: Accuracy on the German syntax tasks, as a function of the number of intervening elements.

We selected all noun lemmas of the appropriate genders from the Universal Dependencies, and extracted morphological paradigms from Wiktionary to obtain case-marked forms, retaining only nouns unambiguously marking the two cases. We created four conditions, varying the amount of intervening material, as in the gender agreement experiment (4,509 stimuli per condition).

Results are in Figure **??** (center). Again, WordNLM has the best performance, but the LSTM CNLM is competitive as more elements intervene. Accuracy stays well above 80% even as three words intervene. The n-gram model performs well if there is no intervening material, and at chance otherwise. The RNN CNLM accuracy remains above chance for one or two intervening elements, but drops considerably.

Case subcategorization German verbs and prepositions lexically specify their object's case. We study the preposition *mit* 'with', which selects a dative object. We use objects whose head noun is a nominalized adjective, with regular, overtly marked case inflection. We take all adjectives that occur at least 100 times in the training data, excluding those that end in *-r*, as these often reflected lemmatization problems. We then select all sentences containing a *mit* prepositional phrase in Universal Dependencies, subject to the constraints that (1) the object is not a pronoun (replacing such items with a nominal object often results in ungrammaticality), and (2) the object is a continuous phrase, i.e., it is not interrupted by words that do not belong to it. We obtained 1,629 such sentences. For each sentence, we remove the prepositional phrase and replace it by a phrase of the form

(3) mit der sehr {rote, roten}
with the very red one

where only the *-en* (dative) version of the adjective is compatible with the case requirement of the preposition (and the intervening material does not disambiguate case). Note that the correct form is longer than the wrong one. This ensures that the probabilistic bias for shorter sequences works against the model. We construct three conditions by varying the presence and number of adverbs (*sehr* 'very', *sehr extrem* 'very extremely', *sehr extrem unglaublich* 'very extremely incredibly'). As a control for baseline probabilities of the two adjectival forms, we also created stimuli where all words up to the preposition were removed, and computed accuracy on these stimuli. If accuracy is lower on these stimuli than on the full ones, we can conclude that the baseline probabilities of the two adjective forms cannot explain success on the task. For the n-gram count model, we only counted the occurrences of the prepositional phrase, omitting the sentence environment.

Results are shown in Figure **??** (right). Only the n-gram model fails to outperform the control accuracy for stimuli not including the preposition. Surprisingly, the LSTM CNLM slightly outperforms the WordNLM, even though the CNLM is exposed to a harder test set without OOV removal. Neither model shows accuracy decay as the number of adverbs increases. As before, the n-gram model drops to chance as adverbs intervene, while the RNN CNLM starts with low accuracy that progressively decays below chance.

4.4.2 Italian

We focus on paradigms where gender and number are explicitly and systematically encoded and it is possible to compare same-length strings. We

are able to extract enough stimuli that never occur in the training corpus, so that an n-gram control would be at chance level. Moreover, by experiment construction, baselines relying on unigram frequency are also at chance level.

Article-noun gender agreement Similar to German, Italian articles agree with the noun in gender; however, Italian has a relatively extended paradigm of masculine and feminine nouns differing only in the final vowel (*-o* and *-a*, respectively). We construct pairs of the form:

(4) a. {il, la} congeniale candidato
the congenial candidate (m.)
b. {il, la} congeniale candidata
the congenial candidate (f.)

The intervening adjective, ending in *-e*, does not reveal noun gender, increasing the distance across which gender information has to be transported. We constructed the stimuli with words appearing at least 100 times in the training corpus. We required moreover that the *-a* and *-o* forms of a noun are reasonably balanced in frequency (neither form is twice more frequent than the other), or both rather frequent (appear at least 500 times). As the prenominal adjectives are somewhat marked, we only considered *-e* adjectives that occur prenominally with at least 10 distinct nouns in the training corpus. We obtained 15,005 pairs of stimuli. Here and below, stimuli were checked for strong semantic anomalies.

Results are shown in the first line of Table **??**. WordNLM shows the strongest performance, closely followed by the LSTM CNLM. The RNN CNLM performs strongly above chance (50%), but again lags behind the LSTM.

Article-adjective gender agreement We next consider agreement between articles and adjectives with an intervening adverb:

(5) a. il meno {alieno, aliena}
the (m.) less alien one
b. la meno {alieno, aliena}
the (f.) less alien one

where we used the adverbs *più* 'more', *meno* 'less', *tanto* 'so much'. We considered only adjectives that occurred 1K times in the training corpus (as *-a*/*-o* adjectives are very common). We excluded all cases in which the adverb-adjective combination occurred in the training corpus, obtaining 88 pairs of stimuli. Results are shown in the second line of Table **??**; all three models perform almost perfectly.

	CNLM		WordNLM
	LSTM	*RNN*	
Noun Gender	93.1	79.2	97.4
Adj. Gender	99.5	98.9	99.5
Adj. Number	99.0	84.5	100.0

Table 7: Italian agreement results.

Article-adjective number agreement Finally, we constructed a version of the last test that probed number agreement. For feminine forms (illustrated below) it's possible to compare same-length phrases:

(6) a. la meno {aliena, aliene}
the (s.) less alien one(s)
b. le meno {aliena, aliene}
the (p.) less alien one(s)

Selection of stimuli was as above, but we used a 500-occurrences threshold, as feminine plurals are less common, obtaining 99 pairs of stimuli. Results are shown in the third line of Table **??**; the LSTMs perform almost perfectly, and the RNN still is strongly above chance.

4.5 Semantics

Finally, we probe CNLM knowledge of semantics. We turn to English, as for this language we can use the Microsoft Research Sentence Completion task (**?**). The challenge consists of sentences with a gap, and a 5-word multiple choice to fill the gap. Picking the right word requires a mixture of syntax, lexical semantics, world knowledge and pragmatics. For example, in *"Was she his [client|musings|discomfiture|choice|opportunity], his friend, or his mistress?*, the model should realize that the missing word is coordinated with *friend* and *mistress*, and that the latter are human beings. We chose this challenge because language models can be easily applied by calculating the likelihood of all possible completions and selecting the one with the highest likelihood. The domain of the task (Sherlock Holmes novels) is very different from the Wikipedia data-set we are using; thus we additionally trained our models on the training set provided for the task, consisting of 19th century English novels.

Results are shown in Figure **??**. The models

	LSTM	34.1/59.0	
	RNN	24.3/24.0	
	WordNLM	37.1/63.3	
KN5	40.0	Skipgram	48.0
Word RNN	45.0	Skipgram + RNNs	58.9
Word LSTM	56.0	**?**	61.4
LdTreeLSTM	60.7	**?**	65.1

Table 8: Results on MSR Sentence Completion. For our models (top), we show accuracies for Wikipedia/in-domain training. We compare with language models from prior work (left): Kneser-Ney 5-gram model (**?**), Word RNN (**?**), Word LSTM and LdTreeLSTM (**?**). We further report models incorporating distributional encodings of semantics (right): Skipgram(+RNNs) from **?**, the PMI-based model of **?**, and the context-embedding based approach by **?**.

trained on Wikipedia perform poorly but above chance (20%). When trained on in-domain data, the LSTM CNLM outperforms many earlier word-level neural models, and is only slightly below WordNLM. The vanilla RNN is not successful even when trained in domain, contrasting with *word*-based vanilla RNNs, whose performance, while below that of LSTMs, is much stronger.

5 Discussion

We probed the linguistic information induced by a character-level LSTM language model trained on unsegmented text. The model was found to possess implicit knowledge about phonotactics, word units, morphological features, syntactic agreement phenomena and basic semantics. A more standard model pre-initialized with a word vocabulary and reading tokenized input was in general superior on the higher-level tasks, but the performance of our agnostic model did not generally lag much behind, suggesting that the word prior is helpful but not fundamental. The performance of a character-level RNN was less consistent than that of the LSTM, suggesting that the ability of the latter to track information across longer time stretches is important to extract the correct linguistic generalizations. N-gram baselines relying on adjacent-string statistics failed almost everywhere, showing that the neural models are tapping into somewhat deeper linguistic templates.

Our results are preliminary in many ways. The tests we used are generally quite simple (we did not attempt, for example, to model long-distance agreement in presence of distractors, a challenging task even for word-based models and humans: **?**). Still, they suggest that a large corpus, combined with the weak priors encoded in an LSTM, might suffice to support genuine linguistic generalizations. In future work, we will check if stronger priors are needed when learning from smaller amounts of training data.

Unlike standard word-level models, CNLMs lack a word-based lexicon. Any information they might acquire about units larger than characters must be stored in their recurrent weights. Given that nearly all contemporary linguistics recognizes a central role to the lexicon (see, e.g., **?????**), in future work we would like to explore how lexical knowledge is implicitly encoded in the distributed memory of the CNLMs.

One of our original motivations for not assuming word primitives is that a rigid word notion is problematic both cross-linguistically (cf. polysynthetic and agglutinative languages) and within a single linguistic system (cf. the common view that the lexicon hosts units at different levels of the linguistic hierarchy, from morphemes to large syntactic constructions). Our brief analysis of the CNLM over- and undersegmentations suggested that it is indeed capable to flexibly store information about units at different levels. However, this topic remained largely unexplored, and we plan to tackle it in future work.