Morpho-Syntactic Awareness in a Character-Level Language Model

Yova Kementchedjhieva

Master of Science
School of Informatics
University of Edinburgh
2017

Abstract

This thesis presents the results of an exploratory study into the morpho-syntactic awareness of character-level neural language models. The aim is to identify the source of performance and limitations of these models, such that future research can make theory-informed efforts toward advancement. We analyze an English character-to-character language model in isolation and as embedded into systems for morphological and syntactic analysis. Findings point to an ability of the model to identify units of higher order, such as morphemes and words, when an explicit cue is available. We further run linguistic experiments on the model and observe that its ability to learn morpho-syntactic regularities is conditioned on its access to the relevant units. These findings imply a limitation in the applicability of character-level models to their main target group, morphologically-rich languages.

Acknowledgements

I would like to thank Adam Lopez for being the best supervisor I could have wished for. Adam truly engaged with my work, supplied great ideas for experiments and directions for analysis, and provided me with helpful feedback.

I would also like to thank the following PhD and master's students who helped me with advice on approaches, implementations and writing: Clara Vania, Sameer Bansal, Federico Fancellu, Andreas Grivas and Mark Anderson.

Declaration

Table of Contents

Chapter 1

Introduction

Type the following Spanish sentence in Google Translate: *Queiro pizza vegetariana*. You will not be surprised to see that instead of the literal translation *Want pizza vegetarian*, Google offers the well-formed English sentence *I want vegetarian pizza.* How does the translator know to change the order of words and insert a pronoun?

Google Translate uses a neural machine translation system, that inherently contains a component called a *language model*. The aim of a language model is to capture the formal and functional properties of a natural language. The language model of Google Translate knows that in English, unlike in Spanish, adjectives precede nouns, so it swaps *pizza* and *vegetarian*. It also knows that the subject of an English sentence should always be explicit, so it needs to insert a pronoun in the target sentence. It infers the pronoun from the form of the verb: *quiero* is the first person singular form of *querer* so the correct pronoun here is *I*. The language model needs to not only know what a word means, but what its grammatical properties are and how they interact with other words in the sentence.

Fundamental approaches to language modeling started off processing input on the word level (Mikolov et al., 2013). More recently, a shift has been in progress towards processing on the subword level, to allow access to the aforementioned grammatical properties of words. Luong et al. (2013), Botha and Blunsom (2014) and Qiu et al. (2014) all find empirical gains from morpheme-level processing. Kim et al. (2015) and Ling et al. (2015), find character-level processing beneficial as well. The source of performance in such cases is speculatively attributed to several factors: more appropriate handling of words that were not seen during training; shared statistics between rare words and related more frequent words; and, importantly, linguistic awareness on the subword level. Research into the benefits of subword-level processing has found that the properties of languages do interact with the performance of subword models (Vania and Lopez, 2017). Yet, the exact aspects of this interaction are still to be surveyed in depth.

This work takes on an extensive exploratory study of a character-level language model, as applied to English, with the aim to identify its source of performance and to draw conclusions about theoretical limitations and venues for further improvements. The findings of the study can be summarized in two main takeaways:

- The successful identification of a linguistic unit is a prerequisite for the learning of its properties and the processes in which it participates.
- A character-level language model can learn to identify linguistic units of higher order, such as morphemes and words, but only when an explicit cue is available to delimit these units.

The main implication of these conclusions is that while a character-level language model provides some access to the morphology of English, it should have limited ability to do so for morphologically-rich languages. This is ironic, since the theoretical benefits of character-level processing are much greater for morphologically-rich languages.

The rest of this thesis proceeds as follows: in chapter 2 we outline the background for the language model studied in this work, the linguistic notions it was probed for and the methods that were used for that; in chapter 3 we explore the language model in isolation; in chapters 4 and 5 we look into the properties of the language model as embedded into other natural language processing (NLP) systems; in chapters 6 and 7 we discuss two linguistic experiments on the language model; finally, in chapter 8 we draw conclusions based on the previous five chapters.

Chapter 2

Background

2.1 Morphology and Syntax

The term **morphology** refers to the language domain responsible for word formation. The units which morphology uses to build words are called **morphemes**. The morphologically complex word *reaction*, for example, consists of two morphemes, *react* and *ion*. Morpheme *react* is a *base* morpheme – it carries the semantic content of the word. Morpheme *ion*, on the other hand, is an **affix** – it serves a function, namely to convert the verb *react* into a noun. Affixes that attach to the end of a word, like *ion* are called **suffixes**, while those that attach to the beginning of a word are called **prefixes**. One final remark on morphemes relevant to this study is the distinction between a **free** base, like *react* which is a word in itself, and a **bound** base, like *acquisit* (as in *acquisition*), which cannot occur independently of a suffix.

Affixes can be **inflectional** or **derivational**. Inflectional affixes generate the grammatical forms of a word, e.g. from *bake* they can produce *bake**s**, bak**ed**, bak**ing***. These forms differ in person, tense and aspect, but they all refer to the same notion. Derivational affixes, in contrast, produce new words referring to a related but different notion, e.g. *baker, bakery*, *like, dislike*.

Inflectional affixes that mark a single grammatical category, e.g. person, or number, or tense, etc., are called *agglutinative*. *Fusional* affixes, in contrast, mark multiple grammatical categories. An example from English is the deverbal suffix **s**, as in **likes**, which marks both the properties *3rd person* and *singular*. These terms are also used to refer to languages characterized predominantly by one type of affix or the other. English is a fusional language.

English is also notoriously impoverished with respect to its inflectional morphology, with a verbal paradigm consisting of only four slots (listed above for *bake*). Its derivational morphology, on the other hand, is copious. Most of this work thus focuses on derivational affixes. Hereafter, the terms suffix and prefix will always refer to derivational morphemes, unless stated otherwise.

The term **syntax** refers to the language domain that builds sentences from words. One fundamental syntactic notion is that of the **syntactic category** of a word. Common English syntactic categories are verb, noun, adjective, adverb and proper noun. The term syntactic category and its synonymous term **part of speech** are used interchangeably throughout.

Morphology and syntax often interact on the word level, as illustrated by two main properties of English suffixes and prefixes.

Suffixes have **selectional restrictions** with respect to the syntactic category of the base they would attach to. Suffixes *al, ment* and *ance*, for example, only attach to verbs, e.g. *betrayal, annoyance, containment*, while *hood, ous, ic* only attach to nouns, as in *nationhood, spacious and metallic*. The former are thus known as **deverbal**, and the latter as **denominal**. Certain suffixes are members of more than one class, e.g. *ful* attaches to both verbs and nouns, as in *forgetful* and *peaceful*, respectively. (All examples are from Fabb (1988)).

The majority of English prefixes have a **non-category changing** effect on their bases, i.e. they don't change the syntactic category of the base they attach to. Consider the examples *write* and *rewrite*, which are both verbs, and *kind* and *unkind*, which are both adjectives. A small set of prefixes do have that property but they are infrequent and are considered unproductive, i.e. they cannot be used to create new English words.

These properties of suffixes and prefixes can also be referred to as **linguistic regularities** since they exhibit a rule-like behavior, with a specified trigger for and result of the rule.

2.2 Language Modeling

In its essence, language modeling is just the computation of **joint probabilities** over a string of linguistic units with the aim to either recognize a valid string or generate one. The joint probability of the words in *I like pizza*, for example, is computed as:

$$p(I\ like\ pizza) = p(I) \times p(like|I) \times p(pizza|like)$$

This probability indicates how likely it is for these words to occur in this order.

Recent approaches to language modeling employ recurrent neural networks (Elman, 1990). Recurrent neural networks (RNN) have several properties that make them particularly suitable for the task: they can take input of variable length (e.g. a sentence with five words and another with six words), and they allow, *theoretically*, for patterns of infinite length to be learned. In practice, RNNs do have a limited memory, however, due to the nature in which they learn from their errors. RNNs are thus often augmented with LSTM units (Hochreiter and Schmidhuber, 1997), which process input through a gated mechanism, maintaining a memory flow over longer, yet still not infinite, sequences.

2.2.1 Our Model

The language model (LM) explored in this thesis is a 'wordless' character-to-character RNN with LSTM units. On every timestep, t, it receives a character, c_t, and predicts the following character, c_{t+1}. Input is not split into words and spaces are treated just like any other character. Characters are fed as one-hot vectors and their **embeddings**[1] are trained along with the language model. This architecture closely follows the one informally proposed in Karpathy (2015).

Karpathy trains a language model on the Tiny Shakespeare corpus and experiments with text generation. Consider the following passage generated by Karpathy's model:

> Second Senator:
> They are away this miseries, produced upon my soul,
> Breaking and strongly should be buried, when I perish
> The earth and thoughts of many states.

The sampled text closely resembles the training data on a lexical and structural level, showing that the model has learned much of the formal properties of English.

To my knowledge no scholarly publications employ such a language model in isolation. Yet, 'wordless' character-to-character machine translation systems, which inherently contain a language model, have been shown to outperform other subword-level approaches, which themselves are better than word-level appraoches (Lee et al., 2016).

The reason we chose to work with this particular architecture is that a 'wordless' LM allows for experiments on a subword, i.e. morphological level: we can feed a partial word and ask the model to complete it. As the focus of this project is to study

[1]Embeddings are fixed-dimensional continuous representations of discrete entities.

the morpho-syntactic awareness of character-level neural embeddings, this particular architecture seemed most suitable.

2.2.2 Other Models

A range of word and subword architectures is explored in a comparative study by Vania and Lopez (2017). One distinguishing characteristic of these architectures as compared to the ‘wordless’ character-level RNN is that they process words on a subword level, obtain embeddings for words and then process sentences on a *word* level. The models studied in Vania and Lopez (2017) further differ from one another in terms of the subword units (character, character trigram, morpheme, binary pair encoding) and type of combination function (addition, recurrent neural network, convolutional neural network).

The study finds that subword-level models are generally better than word-level models. The character-trigram biLSTM architecture emerges as optimal on seven out of ten languages considered. The performance of different models, however, appears to correlate with the morphological typology of languages. Fusional languages, for example, categorically behave best under character trigram bi-LSTMs, but that is not the case for agglutinative languages. This is evidence that subword-level word embeddings indeed interact with morphology. The nature of this interaction is a focal point of interest from both a theoretical and a practical perspective.

One important finding of Vania and Lopez (2017) is that providing explicit morphological information during training boosts the performance of even the best morphologically-unaware architectures. This shows that some level of morphological regularity remains outside the scope of the models when trained on raw data alone.

2.3 Morphological Segmentation

Morphological segmentation aims to identify the boundaries in morphologically complex words, i.e. words consisting of multiple morphemes. A morpheme boundary could separate a base from an inflectional morpheme, e.g. *like+s*, a base from a derivational morpheme, e.g. *consider+able*, or two bases, e.g. *air+plane*

Morphological segmentation is a sequence-labeling task, where words are processed one character at a time and every between-character position is assigned a binary label. The word $l_1 i_2 k_3 e_4 s$, for example, which has four between-character positions and a boundary in position 4, would be labeled $\langle 0001 \rangle$. RNN approaches are particularly suitable for sequence-labeling tasks (Sutskever et al., 2014) and recent work on supervised morphological segmentation with RNNs shows promising results (Wang et al., 2016).

2.4 Part-of-Speech Tagging

Part-of-speech (POS) tagging is a sequence labeling task used to determine the syntactic category of words in context. The sequence *I like pizza*, for example, would be labeled $\langle$PRONOUN VERB NOUN$\rangle$. Certain words can be unambiguously tagged in isolation: *desk*, for example would always be a noun. Others, however, can take on different parts of speech in different contexts: *love* is a verb in the context *I love pizza*, but it is a noun in the context *my love for pizza*. In such cases, it is useful to have access to the context as well. Common approached to POS tagging thus employ an RNN, in order to condition a tag prediction not just on the current word, but also on previous words and/or previous tags.

While POS tagging can be performed on the word level, recent studies in the field demonstrate that processing input on a subword-level boosts the performance of such systems (dos Santos and Zadrozny, 2014).

2.5 Mathematical Notation

In mathematical equations throughout the thesis, lowercase bold letter represent column vectors, uppercase bold letter represent matrices. $LSTM(\cdot)$ applies both the weights of the RNN and the gates of the LSTM. $softmax(\cdot)$ performs the softmax operation:

$$softmax(x) = \frac{exp(x)}{\sum_{x' \in X} exp(x')}$$

Chapter 3

Inside the Language Model

This chapter defines the character-to-character language model (hereafter, C2C) more formally, outlines the training methods, and explores the trained model. The exploration takes on a qualitative study of samples of output from C2C, of character and word embeddings, and of individual units from the hidden layer of the network.

3.1 Methods

The structure of C2C is visualized in Figure 3.1.

3.1.1 Model Formulation

At each timestep t, character c_t is projected into high-dimensional space by a character lookup table $\mathbf{E} \in \mathbb{R}^{|V| \times d}$ where $|V|$ is the vocabulary of characters encountered in the training data and d is the dimension of the character embeddings.

$$\mathbf{x}_{c_t} = \mathbf{E}^T \mathbf{v}_i$$

$\mathbf{v}_i \in \mathbb{R}^{|V|}$ is a one-hot vector with i^{th} element set to 1 and all other elements set to zero, where i corresponds to the position of character c_t in the vocabulary, V.

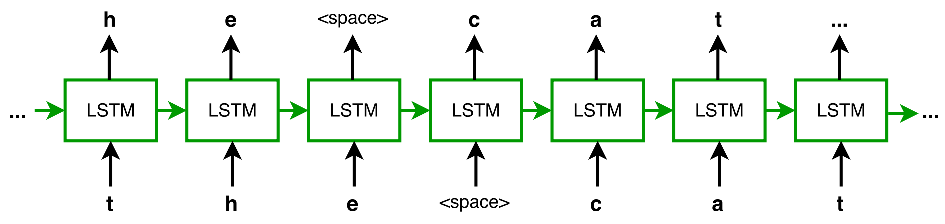


Figure 3.1: C2C Model Architecture.

The hidden state of the neural network is obtained as:

$$\mathbf{h}_t = LSTM(\mathbf{x}_{c_t};\mathbf{h}_{t-1})$$

The hidden state is followed by a linear transformation and a softmax function over all elements of V, which results in a probability distribution.

$$p(c_{t+1} = c|\mathbf{c}) = softmax(\mathbf{W}_o\mathbf{h}_t + \mathbf{b}_o)_i \qquad \forall c \in V$$

where i is the index of c in V.

The embedding matrix, **E**, and all weights and bias terms, including the LSTM ones, are learned during training.

3.1.2 Training Data

The model was trained on a continuous stream of the first 7M tokens from the English Wikipedia corpus. These were all articles on topics starting with the letter A.

3.1.3 Hyperparameters

Hyperparameters were tuned with a search over the grid in Table 5.1. The values in bold emerged as optimal, with a mean log loss on the devset of 1.59.

Parameter	Values			
Layers	**1**	2	3	4
Units	64	128	**256**	
Learning rate	0.0003	**0.003**	0.03	0.3
Minibatches	16	**50**	64	
Backprop	15	**30**	50	
Dropout	**0.2**	0.5		

Table 3.1: Hyperparameters values explored during optimization. The values in bold emerged as optimal.

3.2 C2C's Version of English

Text generated with C2C closely resembles English on the word level and, to some degree, on the level of syntax.

3.2.1 Words

When sampled, C2C generates real English words most of the time, but about 1 in every 20 tokens is a nonce word[1]. Table 3.2 lists some nonce words generated by the model. Regular morphological patterns can be observed within some of these words.

Nonce Words
sinding, fatities, complessed
breaked, indicatement
applie, therapie
knwotator, mindt, ouctromor

Table 3.2: Nonce words generated with C2C through sampling.

The nonce words *sinding, fatities* and *complessed* all seem like well-formed inflected variants of English-looking words. The forms *breaked* and *indicatement* show productive morphological patterns of inflection and derivation being applied to bases of the correct syntactic class, namely verbs. It just happens to be that *break* is an irregular verb and *indicate* forms a noun with suffix *ion* rather than *ment*. These are lexical rules that block the more general structural rules C2C has applied.

C2C not only composes morphologically complex words but also attempts to reverse the process, as can be seen with the forms *therapie* and *applie*. Here the inflectional suffix *s* has been dropped, but the orthographic change associated with it has not been successfully reversed.

Not all nonce words generated by C2C can be explained in terms of morphological productivity, however: *knwotator, mindt*, and *ouctromor* don't resemble any real morphemes and don't follow English phonotactics. These forms may be highly improbable accidents of the sampling process.

3.2.2 Sentences

Consider the sentence in 5.2 generated with C2C through sampling. The sentence could not be considered fluent or grammatical: there is no clear dependencies between verbs, subjects, and objects, it contains the nonce word *eviven*, and the novel and unlikely compound *weaken-little*. Yet, some short-distance syntactic regularities can

[1]As measured by checking for every generated word whether it appeared in the training data or in the *pyenchant* UK or US English dictionaries.

be observed. Articles precede adjectives and nouns but not verbs, prepositions precede nouns, and particle *to* precedes a verb. On an even larger scale, the clause *the novel regarded the modern Laboratory has a weaken-little director* is grammatical in terms of the order between parts of speech[2]. The sentence appears unnatural due to its odd semantics, but consider the following alternative choice of words for the same syntactic structure: *The man thought the modern laboratory has a weaken-little director*. This sentence sounds only marginally anomalous. The predominantly well-formed output of C2C suggests that it is appropriate to further study the linguistic regularities learned by the model.

The novel regarded the modern Laboratory has a weaken-little director and many of them in 2012 to defeat in 1973 - or eviven of Artitagements.

Table 3.3: A sentence generated with C2C through sampling.

3.3 Character embeddings

During training C2C learned high-dimensional continuous representations (embeddings) for the characters in its vocabulary. The character embeddings learned by the model reflect certain linguistically-motivated groupings.

3.3.1 Method

The embeddings were analyzed visually through a dimensionality reduction to a two-dimensional space with the t-SNE tool (Maaten and Hinton, 2008). They were also analyzed numerically using cosine similarity. The cosine similarity scores reported below refer to pairwise similarity, when two characters are compared, and micro-average pairwise similarity for more than two characters.

See Figure 3.2 and Table 3.4 for results. Notice that the similarity across all characters is 0.01, so any other numbers should be interpreted with reference to this baseline.

[2]That is, assuming that *novel* is a noun in this context and *weaken-little* is an adjective, by analogy with its second base.

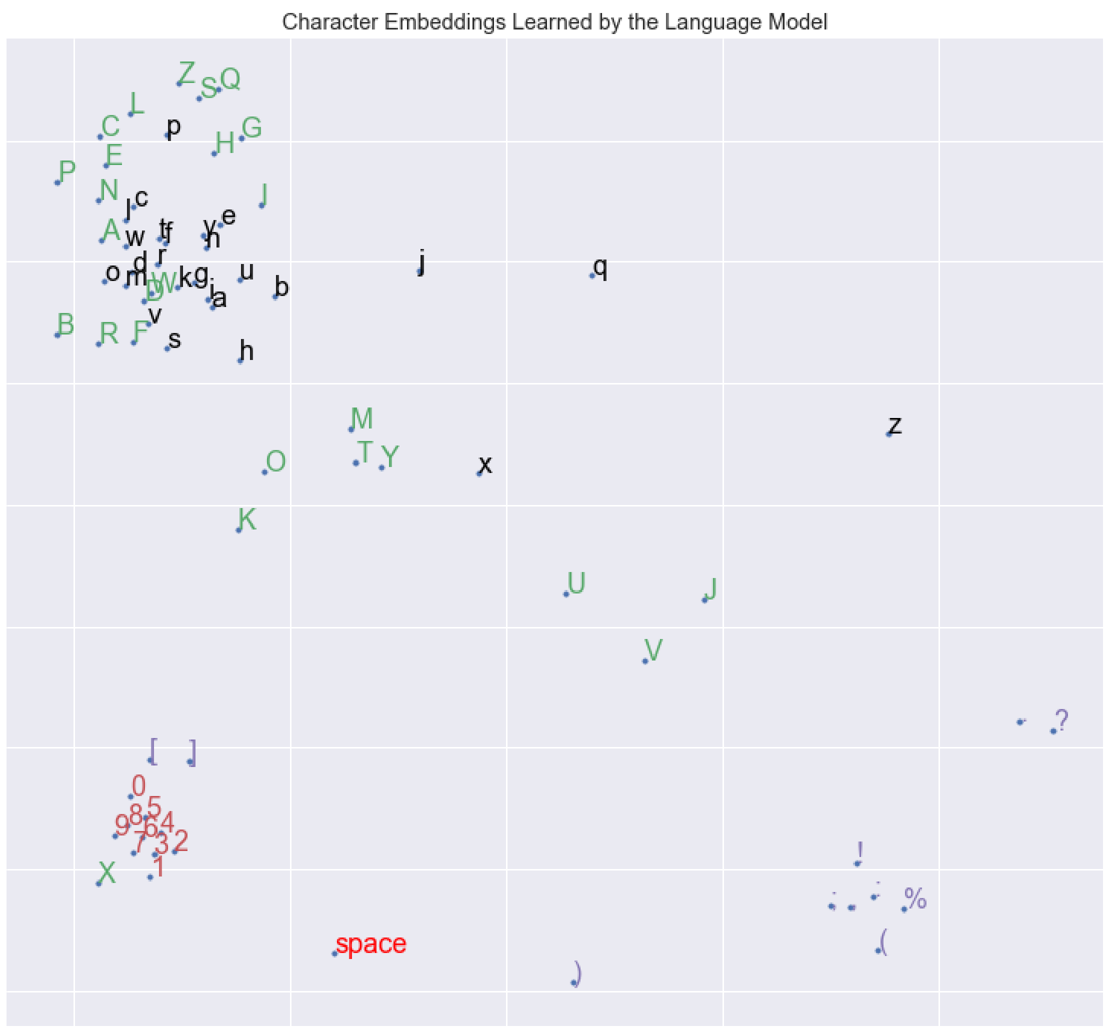


Figure 3.2: Character embeddings in two dimensions. Dimensionality reduction was performed with tSNE (perplexity = 25). Color codes: black – lowercase letters, green – uppercase letters, purple – punctuation, red – digits

Subset	Cosine Similarity
All characters	0.01
Digits	0.7
Punctuation	0.38
Space-Punctuation	-0.02
Space-*A*	0.13
Vowels	-0.01
Lower - Uppercase	0.17

Table 3.4: Cosine Similarity Between Groups of Characters.

3.3.2 Non-Alphabetic Characters

Non-alphanumeric characters are all mapped on one side of the plot, clearly separated from letters, and further clustered into smaller subsets. The ten digits are mapped closely in space, and score a cosine similarity of 0.7. The average pairwise cosine similarity across punctuation marks is 0.38.

The embedding of the space character is mapped close to punctuation marks, as we may expect. Yet, the cosine similarity between a space character and punctuation characters is extremely low: -0.02. The character that a space is most similar to is the letter *A* (cossim = 0.13). This observation seems surprising, but there is an explanation behind it. Assuming that one defining property of characters from a statistical point of view, is what comes after them, a space character and uppercase alphabetic characters share the property that they are both most often followed by a lowercase alphabetic character, rather than a space, a digit or punctuation. Due to the nature of the data (Wikipedia entries starting with *A*), the most frequent uppercase letter seen during training was *A*, so it makes sense that the space character would be matched most closely with this particular uppercase character. Had the language model been trained on the entire Wiki dump, this pattern would have likely repeated with whichever uppercase character had been most frequent across it.

3.3.3 Alphabetic characters

For 28 out of 52 letters their closest character in space was their case counterpart, e.g. the character that is most similar to *f* is *F* and vice-versa. The average pairwise cosine similarity across all pairs of lower- and uppercase versions of the same letter is 0.17. Notice that the t-SNE plot doesn't capture this pattern, which points to the limitations of a visual analysis alone.

A cluster we could expect to see emerging among embeddings is that of the vowel characters. Vowels behave similarly to each other and differently from consonants across a range of interconsonantal contexts, e.g. a vowel character can occur in the context p_r, but a consonant couldn't: *purpose, parsnip, pork, pirate, person*. Although C2C mostly abides by such phonotactic restrictions even in the nonce words it generates, its character embeddings show no particularly high similarity among lowercase vowel character. The average pairwise cosine similarity among lowercase letters for vowels is -0.01, which is in fact lower than the average pairwise cosine similarity across all lowercase characters (0.01). Four out of five lowercase vowel characters, are mapped somewhat closely in 2D space *(a, i, e ,u)*, indicating that C2C may have recognized their similar properties along some dimension.

The most 'similar' pair of lowercase letters is *k* and *d*, which show a cosine similarity of 0.13. From a linguistic point of view there is very little in common between the phonemes corresponding to these letters.

Overall, character embeddings show some expected patterns, based on the distributional properties of characters. Some of these patterns are grounded in linguistics, others are coincidental.

3.4 Word embeddings

Word embeddings are often analyzed in language models that operate on the word level, as a way to measure the semantic awareness of the model. While word embeddings are not intrinsic to C2C, they can stll be obtained and analyzed for semantic similarity.

3.4.1 Method

A common practice in evaluating the performance of word embeddings is to measure the cosine similarity between pairs of semantically related words and to compare it to the similarity between pairs of unrelated words. Four sets of semantically related words were used for evaluation here: sports terms, US states, personal names, and months. A fifth set containing one word from each of the other four sets was used as a reference point for semantic unrelatedness. Only words that occur at least ten times in the training data were included.

C2C was trained on a continuous stream of raw text. The concept of a word is thus not hard-coded in the model, but space characters should make it possible for C2C to implicitly learn to identify units, delimited by a space or punctuation mark on each side. We can thus feed a stream of words into C2C and consider its hidden state after each word-final character as that word's embedding. This method of embedding extraction results in highly contextual embeddings. The word of interest itself is the most recent and thus dominant stimulus, but its preceding context is also taken into consideration. The embedding of *to*, for example, would be different in each of the phrases *I like to*, *he told me to* and *it is difficult to*. More general embeddings can be obtained by taking the element-wise average over the embeddings of all occurrences of a word in some corpus. The corpus of choice for the embeddings discussed below was the Wikipedia data C2C was trained on.

Preliminary experiments showed that formal resemblance dominates over any potential semantic similarity. The words *under* and *October*, for example, which are semantically unrelated but end in the same substring, scored a cosine similarity of 0.535. The highest score obtained by month names that don't bear such formal resemblance was 0.35 between September and July. Pairs of words with extensive formal overlap were thus removed from the semantic sets to abstract away from formal resemblance and evaluate the semantic similarity encoded in the embeddings.

3.4.2 Results

Results are shown in Figure 3.3.

The average pairwise cosine similarity across the 'random' pairs is 0.203, while the average across semantically related pairs is 0.278. The *month* subset is most compactly mapped in the space learned by C2C, likely due to the formulaic contexts in which month names occur: usually preceded by a preposition and in proximity to numbers (*in January 1987*, *on July 4th 1999*). The *people* subset obtained the lowest score out of the four non-random sets, scoring marginally higher than the random subset.

There is no evidence for strong semantic awareness in C2C, but weak patterns of grouping between semantic sets can be seen, especially when the members of a set occur in well-defined syntactic structures.

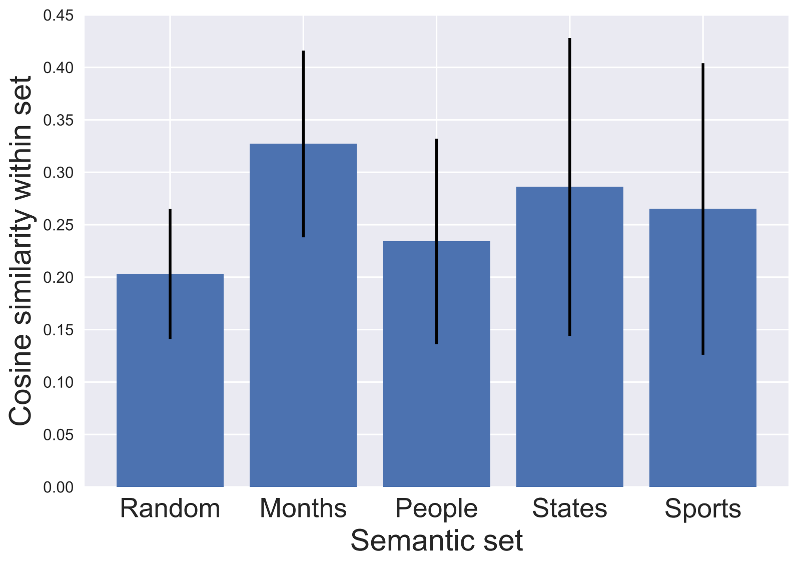


Figure 3.3: Semantic similarity within sets of unrelated (Random) and related (Months, People, States, Sports) words. Similarity within related sets is higher than within the unrelated one, but margins are sometimes very low: compare Random (0.203) and People (0.234).

3.5 Hidden Units

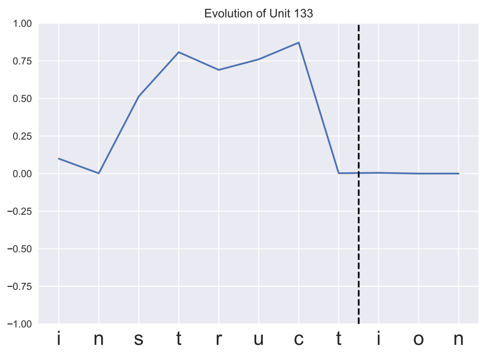
The hidden units of C2C were analyzed by feeding the Wiki data into the system and tracking unit activations on each timestep, i.e. after every character. For each unit, the five inputs which triggered highest activation were recorded. About 40 units exhibited patterns of activation that could be interpreted as meaningful based on just the top five triggers and their contexts. We selected three of the more interesting units to briefly discuss here. Table 3.5 shows a list of the top five triggers for each of these units, together with up to 15 characters that preceded them, to put them in context.

3.5.1 Unit 133

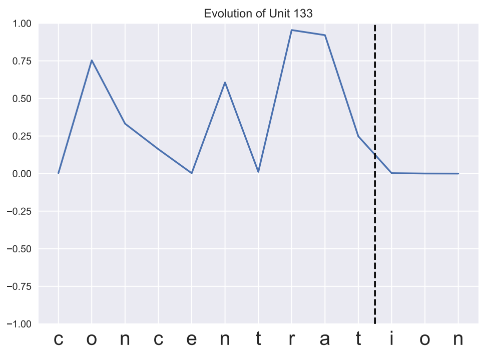
Unit 133 appears to predict the occurrence of suffix -ion. Figure 3.4 shows the unit's behavior in two contexts followed by suffix *ion* or its variants *tion* and *ation*. In both cases activation was highest just before the suffix, i.e. at the end of the base, and dropped to zero for the duration of the suffix.

Unit	Context	Trigger
1	r (1936-1939	)
	School in 1921	.
	ril 13 , 1813	.
	tified in 1901	.
	(1993-1998	)
13	’s prediction	s
	ural relativis	m
	’t contribution	s
	,s were contrac	t
	hat connection	s
133	hered in infor	m
	d to the concentr	a
	cultural recre	a
	was accommod	a
	d Reyes introd	u

Table 3.5: Top 5 Contexts for Three Units in the Network of C2C. The last character in each string marks the peak in activation.



A. Input: *instruction*



B. Input: *concentration*

Figure 3.4: Activation of Unit 133. This unit appears to foresee the occurrence of suffix *ion*.

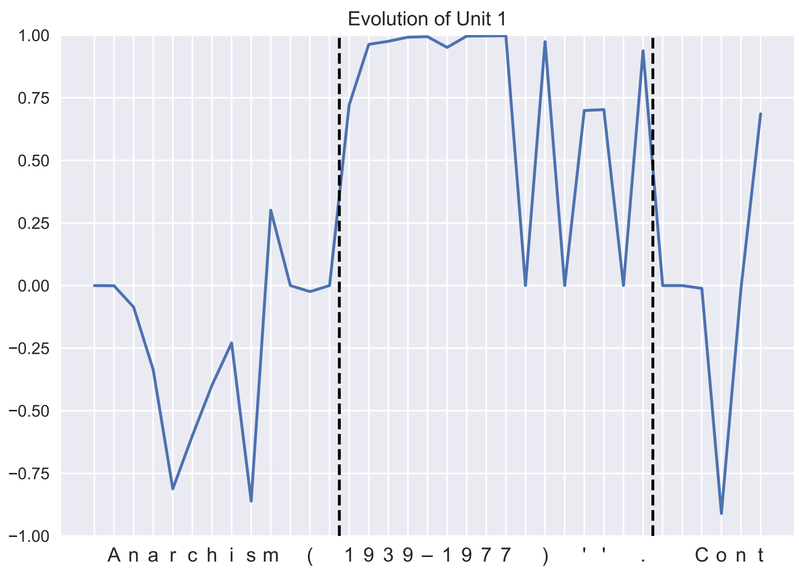


Figure 3.5: Activation of Unit 1. This unit appears to detect sequences of digits and punctuation marks.

3.5.2 Unit 1

Unit 1 fires at a punctuation mark following a digit sequence. The fact that various digit and punctuation sequences trigger the unit, supports the observation made earlier, that C2C indeed recognizes digits and punctuation marks as clusters. Figure 3.5 shows the activation of unit 1 over a sequence that is among the top triggers for this unit.

3.5.3 Unit 13

Unit 13 is possibly the most interesting one, as it appears to recognize complete words even before a space character has been fed to the system. Figure 3.6 shows the activation pattern of unit 13 over a partial sentence from the training data. The dotted lines, which mark the end of tokens, often coincide with the peaks in activation. Even the last dot in the ellipsis causes a high activation in unit 13. The behavior of the unit could be explained either as a signal for the end of a familiar, repeated pattern, or as a predictor of a following space. The fact that we don't see a peak in activation at the right bracket symbol (which should be a cue for a following space) suggests that the former explanation is more plausible. Support for this idea comes from the low correlation coefficient between the activation of unit 13 and the probability the model assigns to a following space: only 0.08. The unit seems to be aware that words can

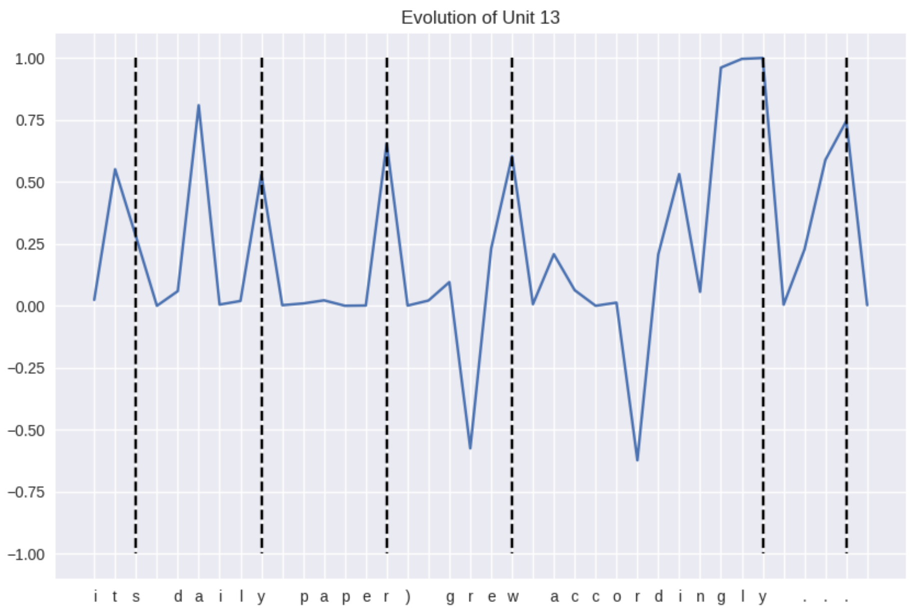


Figure 3.6: Activation of Unit 13. This unit appears to detect complete words.

often be followed by a suffix rather than a space, such as *it* in *its* and *according* in *accordingly*.

3.6 Conclusion

The analysis of the character embeddings, word embeddings and hidden units of C2C indicates that the model has learned surface linguistic regularities, i.e. regularities that can directly be observed in that data, such as the distributional properties of characters, and formulaic similarities in the distribution and shape of semantically related words. The analysis further hints that C2C is capturing more structural dependencies, being able to segment out and decompose individual words.

Chapter 4

Morphological Segmentation is Hard

In this experiment we embed C2C into a system for morphological segmentation to test the extent to which morphological complexity is captured by the language model. The system achieved a rather low F1 score: 53.3. The analysis of correct and incorrect segmentations showed that only certain types of morphemes could be segmented out correctly.

4.1 Method

The architecture of the morphological segmentation system used here can be described as an encoder-decoder: on every timestep a character is encoded with C2C and decoded into a label. The decoder, similarly to the encoder is a RNN with LSTM units (see Figure 4.1). Notice that 4.1 visualizes the testing setup, where a label is not predicted after the last character in a word, since no morpheme boundary could occur there. During training, however, labels for word-final character were included for optimal learning, e.g. to allow the trained model to correctly segment the form *attempts* as *attempt+s*, based on the training instance *attempt* with label sequence 0000001.

4.1.1 Model Formulation

At each timestep t, character c_t is projected into high-dimensional space:

$$\mathbf{x}_{c_t} = \mathbf{E}^T \mathbf{v}_i \qquad \mathbf{E} \in \mathbb{R}^{|V_{char}| \times d_{char}}$$

The hidden state of the encoder is obtained as before:

$$\mathbf{h}_t^{enc} = LSTM^{enc}(\mathbf{x}_{c_t}; \mathbf{h}_{t-1}^{enc})$$

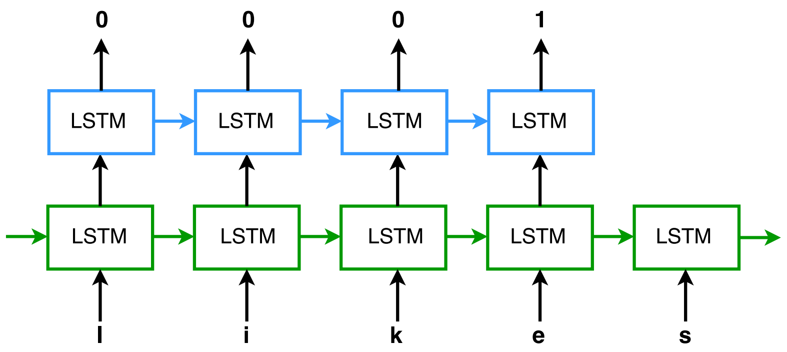


Figure 4.1: C2M Model Architecture. The encoder (in green) is C2C, the decoder (in blue) is trained for the task.

The hidden state of the decoder is then obtained as:

$$\mathbf{h}_t^{dec} = LSTM^{dec}(\mathbf{h}_t^{enc}; \mathbf{h}_{t-1}^{dec})$$

and followed by a linear transformation and a softmax function over all elements in V_{lab}, which results in a probability distribution over labels.

$$p(l_t = l|\mathbf{c}, \mathbf{l}_{word}) = softmax(\mathbf{W}_o^{dec}\mathbf{h}_t^{dec} + \mathbf{b}_o^{dec})_i \qquad \forall \quad l \in V_{lab}$$

where $\mathbf{l}_{word}$ refers to all previous labels for the current word and i refers to the index of l in V_{lab}.

The embedding matrix, $\mathbf{E}$ and $LSTM^{enc}$ weights are taken from C2C. Decoder weights and bias terms are learned during training.

4.1.2 Training Data

The model (hereafter referred to as C2M) was trained on a combined set of gold standard (GS) segmentations from Morpho Challenge 2010 and Hutmegs 1.0 (free data). The training data consisted of 2275 word forms. While this is a fairly limited training set, it should be sufficient to reflect frequent morphological operations.

For the purposes of meaningful C2C embedding, which is highly contextual, a past context is necessary. Yet, GS segmentations are available for words in isolation only. The Wikipedia dump, being one of the largest corpora of raw English text, was used to extract contexts for every word, taking the 15 tokens that preceded the word on up to 15 of its appearances in the dump. The occurrence of a word in each of its contexts was then treated as a separate training instance. Unfortunately, about 20% of the words in the GS dataset never appear in the Wiki dump – these words had to be processed in isolation, i.e. without a preceding context. This method resulted in a dataset of size 8989 on the word level, and 44937 on the character level. Training, development and test set were obtained with a random 80-10-10 split.

4.1.3 Hyperparameters

The hyperparameters of C2M were tuned with a search over the grid in Table 5.1 with the values in bold emerging as optimal. The performance of C2M on a test set of 228 words with 3.5 contexts each on average (6683 word-internal morpheme boundaries) was evaluated in terms of precision and recall of morpheme boundaries.

Parameter	Values		
Layers	1	**2**	3
Units	**64**	128	256
Learning rate	**0.001**	0.01	0.1
Dropout	**0.2**	0.5	0.65

Table 4.1: Hyperparameters values explored during optimization. The values in bold emerged as optimal.

4.2 Performance

Table 4.2 reports the results obtained with C2M and compares them to the results of Ruokolainen et al. (2013), who trained a bidirectional CRF model on the Morpho Challenge data alone, i.e. on a subset of the training data used here. C2M is clearly inferior to the CRF approach, but this is unsurprising given that the CRF makes predictions conditioned on past and future context, while C2M only has access to the past context. The F1 score of 53.3 suggests that C2C is capable of encoding some information about morpheme boundaries despite this limitation. A breakdown of morphemic boundaries by type provides insights into the source of performance and limitations of C2M.

4.2.1 Potential Word Edges as Cues for Morpheme Boundaries

The results labeled *C2M - WE* and *C2M - ¬WE* in Table 4.2 refer to two types of morpheme boundaries: boundaries that could also be a word edge (WE), e.g. *drink+ing, agree+ment*, and boundaries that could not be a word edge (¬ WE), .e.g. *dis+like,*

Model	Precision	Recall	F1
CRF-Ruokolainen et al. (2013)	89.8	83.5	86.5
C2M	52.8	53.9	53.3
C2M - WE	76.6	62.6	68.9
C2M - ¬WE	23.1	34.2	27.6
C2M - EOW	98.5	84.4	90.90
C2M - ¬PREF	53.6	59.2	56.3

Table 4.2: C2M Performance. The main result is above the line. Below the line, WE stand for word edge, EOW for end of word, and PREF for prefix.

intens+ify. It becomes apparent that a large portion of the correct segmentations produced by C2M can be attributed to an ability to recognize word edges. Earlier findings relating to unit 13 of C2C (chapter 3.5.3) support this line of argument: we know that C2C has the means to recognize word edges. The fact that morpheme boundaries often resemble word edges allows for extensive transfer of knowledge.

The sample segmentations in A and C of Table 4.3 can be straightforwardly explained in terms of transfer knowledge on word edges: *act, action* and *ant* are all words C2C has encountered during training. Notice that *ant* has not been observed by C2M, but its status as a word is encoded by C2C.

4.2.2 Actual Word Edges

An interesting result emerges when C2M's performance is tested on word-final characters, which by default should all be labeled as a morpheme boundary (*C2M - EOW* in Table 4.2). Recall that the rest of the results exclude these predictions, since morphological segmentation concerns word-internal morpheme boundaries. C2M performs extremely well at identifying actual word endings. The margin between *C2M - WE* results and *C2M - EOW* results is substantial, even though both look at units of the same type, namely words. The higher accuracy in the EOW setting shows that C2C prefers to ends word where they actually end, rather than at earlier points that would have also allowed it. The reasoning may relate to a syntactic restriction.

Consider example E in Table 4.3. This instance of the word *intensely* occurred in the context of *the Himalayan regions of.* In this context, C2C is reluctant to predict a word edge after *in* (and by extension C2M fails to predict a morpheme boundary), even though *in* is a very frequent word on its own. C2C may be aware that preposition

	Input	True Segmentation	Predicted Segmentation	Correct
A.	actions	act+ion+s	act+ion+s	✓
B.	acquisition	acquisit+ion	acquisit+ion	✓
C.	antenna	antenna	ant+enna	
D.	included	in+clud+ed	in+clude+d	✓
E.	intensely	in+tense+ly	intense+ly	
F.	misundestanding	mis+under+stand+ing	misunder+stand+ing	
G.	woodwork	wood+work	wood+work	✓

Table 4.3: Sample C2M Predictions. Predictions are visualized symbolically for clarity, e.g. the label sequence 00010 for *likes* would be visualized as *like+s*

in would not fit syntactically in the context, i.e. that the sequence *regions of in* is ungrammatical. It thus waits to see a longer sequence that would better fit the context, such as *intense* or *intensely*.

Example D shows that C2M is indeed capable of segmenting prefix *in* in other contexts. The word *included* is preceded by the context *the English term propaganda*.This context allows a following preposition: *the English term propaganda in*, so C2C predicted that the word may end after just *in*, which allowed C2M to correctly predict a boundary after the prefix.

4.2.3 Bound Bases

Morpheme *acquisit* (as in *acquisition*, example B), was not observed by C2M during training and it is a bound morpheme that could not have been observed on its own by C2C. Yet, it was segmented out correctly on five out of fifteen occasions. We generated endings with C2C conditioned on *acquisit* in its different contexts and found that C2C always predicted the expected ending *ion*. It appears that C2M managed to generalize from free bases that take *ion*, such as *correct, react, predict* to the bound base *acquisition*. Recall that C2C has a unit that specializes in recognizing bases that take suffix *ion* (section 3.5.1). This explains the model's ability to share knowledge between different bases that all take that suffix.

4.3 Conclusion

The main takeaway from the performance of C2M is that C2C is remarkably better at capturing word edges than morpheme boundaries, likely due to the explicit cues provided by the space character and punctuation. The fact that morpheme boundaries often resemble word edges in English allows C2C to indirectly capture some morphological boundaries such as those following a free base or a suffix. This finding explains C2C's ability to compose novel morphologically complex forms, such as *sinding* and *fatities*, and to decompose familiar ones. Yet, it also makes it theoretically impossible for C2C to segment out most prefixes, since they don't occur as independent words.

C2C seemingly has the potential to generalize knowledge about boundaries from free bases to bound bases when they occur in the same context. Yet, this generalization involves the highly frequent suffix *ion* and no similar generalizations were observed for other suffixes. This suggests that generalizations of this type require large amounts of data.

Chapter 5

Part-of-Speech Tagging is Easy

This chapter explores two POS tagging models based on C2C. They were designed to be used as an exploratory tool for the linguistic awareness encoded by C2C and as an auxiliary tool in the morphological experiments described in subsequent sections. The taggers achieved an accuracy score of 87.06% and 90.75%. It is worth exploring the source of such high performance in spite of certain structural limitations.

5.1 Method

Two model architectures were explored: an RNN encoder-decoder with LSTM units that predicts tags on the character level (hereafter C2T) and a feed-forward neural network that predicts tags on the word level (W2T). In both models, input is encoded with C2C. The architecture of the models is visualized in 5.1 and 5.2.

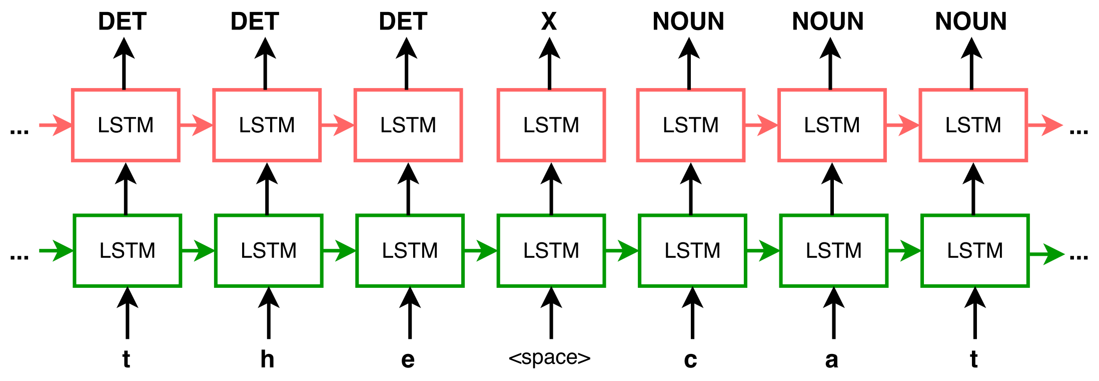


Figure 5.1: C2T Model Architecture. The encoder (in green) is C2C, the decoder (in pink) is trained for the task.

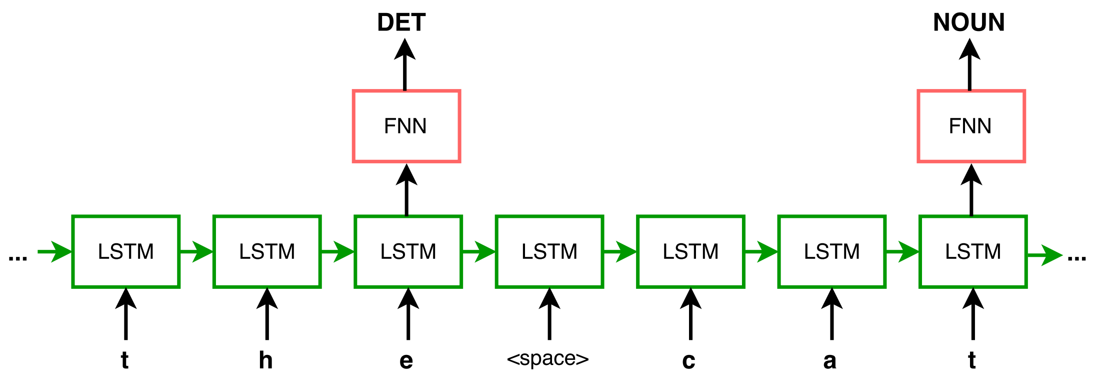


Figure 5.2: W2T Model Architecture. The encoder (in green) is C2C, the decoder (in pink) is trained for the task.

5.1.1 Model Formulation

Encoding is the same for both models. At each timestep t, character c_t is projected into high-dimensional space:

$$\mathbf{x}_{c_t} = \mathbf{E}^T \mathbf{v}_i \qquad \mathbf{E} \in \mathbb{R}^{|V_{char}| \times d_{char}}$$

The hidden state of the encoder is obtained as:

$$\mathbf{h}_t^{enc} = LSTM^{enc}(\mathbf{x}_{c_t}; \mathbf{h}_{t-1}^{enc})$$

5.1.1.1 Character-level Decoding

The hidden state of the character-level decoder is computed on every timestep, as:

$$\mathbf{h}_t^{dec} = LSTM^{dec}(\mathbf{h}_t^{enc}; \mathbf{h}_{t-1}^{dec})$$

and followed by a linear transformation and a softmax function over all elements V_{tag} which results in a probability distribution over tags.

$$p(t_t = t|\mathbf{c}, \mathbf{t}) = softmax(\mathbf{W}_o^{dec}\mathbf{h}_t^{dec} + \mathbf{b}_o^{dec})_i \qquad \forall \quad t \in V_{tag}$$

where i is the index of t in V_{tag}.

5.1.1.2 Word-level Decoding

Word-level decoding happens only at timesteps where a word ends, t':

$$p(t_{t'} = t|\mathbf{c}) = softamx(tanh(\mathbf{W}_o^{dec}\mathbf{h}_t^{enc} + \mathbf{b}_o^{dec}))_i \quad t \in V_{tag}$$

The embedding matrix, **E** and $LSTM^{enc}$ weights are taken from C2C. Decoder weights and bias terms are learned during training.

Similarly to C2M, encodings in both C2T and W2T were obtained for tokens in context, which means that POS tags were conditioned not only on the current token, but also on past tokens. This creates potential for the disambiguation of homonyms and zero-derivation forms of differing syntactic categories, like drink$_{NOUN}$ and drink$_{VERB}$.

5.1.2 Training Data

Both systems were trained on the English UD corpus with a 80-10-10 train-dev-test split with UD POS tags. Training data for character-level prediction was created by pairing each character of a word with the POS tag of that word, e.g. 'like$_{VERB}$' was labeled as as ⟨ VERB VERB VERB VERB ⟩. UD doesn't specify a POS tag for the space character, so the generic X tag was used for it. Since W2T was trained on the word level, the space character was not presented to W2T at any point during training.

5.1.3 Hyperparameters

Hyperparameters were optimized with a search over the grid in Table 5.1 with values in bold and italics emerging as optimal for C2T and W2T, respectively. Notice that W2T was trained without minibatches and dropout.

Parameter	Values			
Layers	1	***2***	3	
Units	64	**128**	*256*	
Minibatches	16	**50**	64	
Learning rate	*0.005*	0.001	0.02	**0.1**
Backprop	15	30	**50**	
Dropout	**0.2**	0.5		

Table 5.1: Hyperparameters values explored during optimization. The values in bold and italics emerged as optimal for C2T and W2T, respectively.

5.2 Results

C2T obtained an accuracy score of 78.85% on the character level and 87.06% on the word level, where word-level accuracy was measured by comparing the tag predicted for the last character of a word to the gold standard tag. The per-character score is naturally lower by a large margin, as predictions early on in the word are based on very little information about the identity of the word. W2T outperformed C2T, scoring 90.75%. Notice that the per-word scores for both C2T and W2T fall short of the state-of-the-art in POS tagging due a structural limitation: both taggers assign tags based on just past and present information. State-of-the-art taggers have access to future information as well, through bidirectional encoding of the input. The high accuracy of C2T in spite of this limitation suggests that the majority of the information concerning the POS tag of a word is contained within that word and its past context, and that C2C is particularly good at encoding this information.

5.2.1 Main Source of Error: Function Words

The confusion matrix in Figure 5.3 provides some insights into the performance of C2T on the test set. Overall, C2T does well at tagging content words, with adjectives and proper nouns posing a slightly higher challenge than verbs, adverbs and nouns, often being mislabeled as nouns. Figure 5.4 shows C2T's performance on words that occurred three times or less in the training data. The NOUN and PNOUN columns in this plot are densest, suggesting that when C2T has little previous knowledge about a word, it defaults to a nominal tag. This is not surprising, however, considering the fact that words at the tail of the Zipfian distribution are almost exclusively nouns and proper names (Fung (1995)). In the UD data the taggers were trained on, 71% of the words that occurred three or less times were nouns and proper nouns. The other 29% were mostly verbs and adjectives.

The confusion matrices for W2T closely resemble those for C2T, so they are not included in the report.

5.3 Evolution of Tag Predictions over Time

Figure 5.5 illustrates the evolution of POS tag predictions over the string *and I have already overheard youngsters* (extract from the UD data) as processed by C2T and W2T. Even though W2T is trained on the word level, it is interesting to observe how its predictions progress on the character level, especially when handling morphologically complex words.

The first thing to notice is that C2T predictions are a lot less scattered than W2T

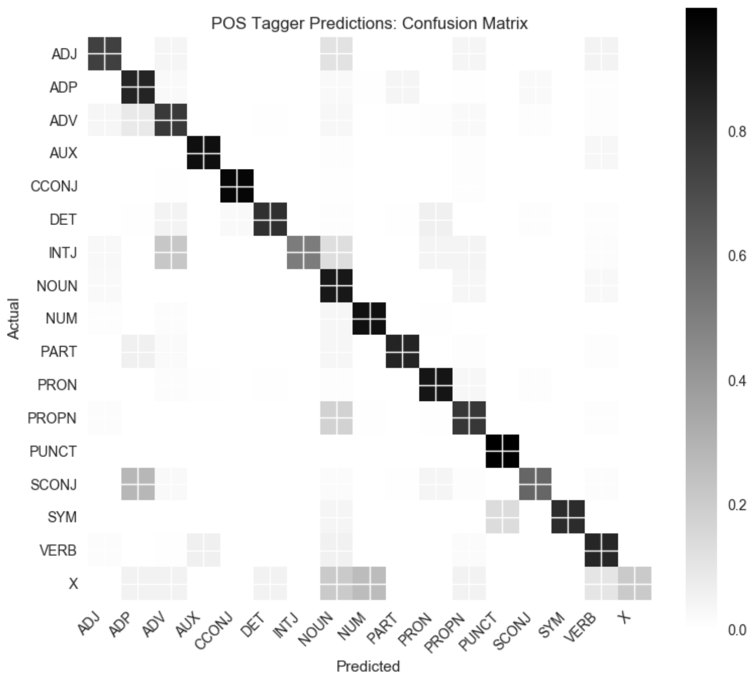


Figure 5.3: C2T Predictions. Confusion matrix of C2T predictions over test set.

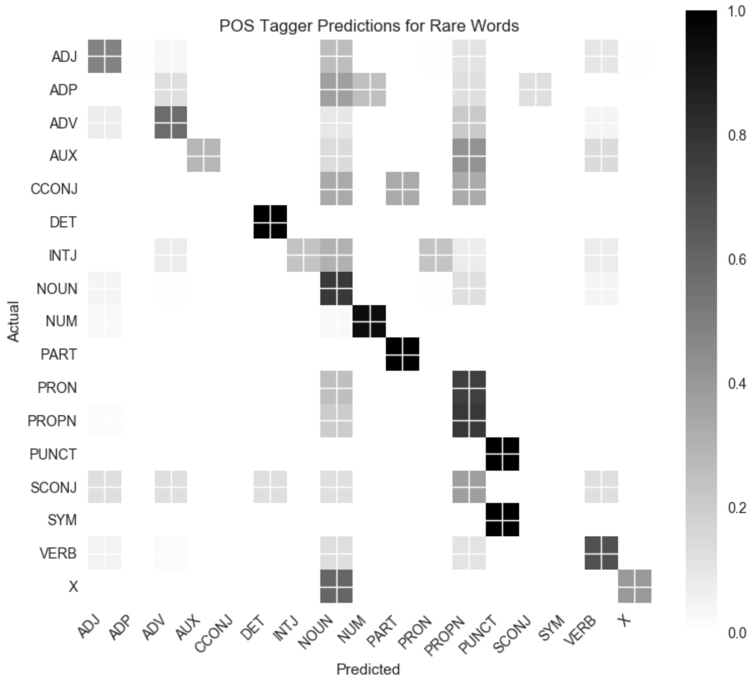


Figure 5.4: C2T Predictions for Rare Words. Confusion matrix of C2T predictions over items in the test set that occurred less than 4 times in the training set.

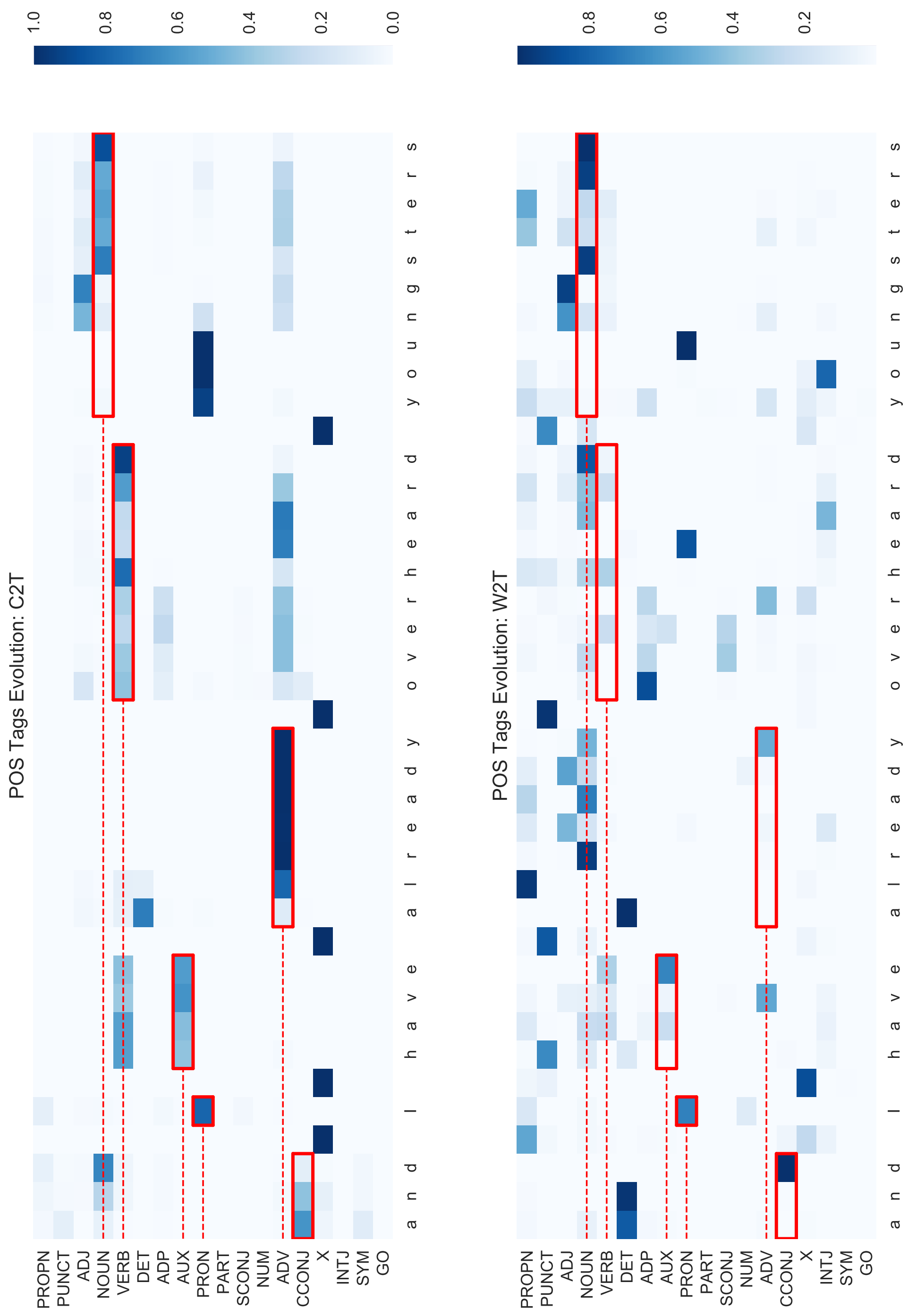


Figure 5.5: C2T (left) and W2T (right) Evolution of POS Tag Predictions. Red rectangles point to the correct tags of words.

predictions. This could be expected, given that C2T was trained to process and predict from partial words and W2T was trained on the word level only. Early into the word *already*, for example, C2T identifies the word, recognizes it as an adverb and maintains this hypothesis throughout. W2T, on the other hand, swings between ADJ and NOUN for most of the word and only recognizes it and switches to the correct POS tag upon seeing the very last character. With respect to the morphologically complex word *youngsters* we see both models making reasonable predictions, predicting PRON for *you-*, ADJ for the next two characters and NOUN for *youngster-* and *youngsters*. The only point at which W2T falls behind C2T is at the intermediate state between *young-* and *youngster-*.

5.3.1 C2T Keeps Two Hypotheses

An interesting feature of the evolution of C2T are the two competing hypotheses for the syntactically ambiguous word *have*, which could be either a verb or an auxiliary, as in this case. C2T appears to have learned this property of *have* and to process the word accordingly.

The weight allocated to the ADV tag during the processing of *overheard* and *youngsters* could also be interpreted as a competing hypothesis. A wider context window showed that C2T often holds multiple hypotheses for content words, with the ADV tag being a common second hypothesis. This tendency might relate to the fact that adverbs are often derived from adjectives through *-ly* suffixation, e.g. *excited+ly, confusing+ly*, but adjectives are also often derived from other categories, like verbs *excite+ed, confuse+ing*. This introduces uncertainty when processing such morphologically complex words on the character level.

5.3.2 W2T Learns About Spaces

Figure 5.5 shows that W2T handles spaces in an intuitive way, even though it did not see this character during training. W2T predicts X for a space character on one occasion and PUNCT on three other occasions. From the perspective of W2T what comes after a word is either another word or a punctuation mark. In this sense, the fact that C2T assigned a PUNCT tag to some spaces means that on these occasions it recognized that the previous word had ended and that what it had seen since then (i.e. the space character) didn't constitute a word in itself, therefore it must be a punctuation mark. This observation provides further support for the idea that C2C sees words as units with an identifiable end.

It isn't immediately obvious, based on any differences between the two models discussed above, why C2T should obtain lower accuracy than W2T on the word-level. Yet, the difference of 3.69% is a sizable one. In the rest of this work, W2T is thus used for classification purposes, which are all on the word level, and C2T is used for illustrative purposes in analyzing subword-level phenomena.

5.4 Conclusion

The performance of the two POS taggers based on C2C encodings demonstrated a high level of syntactic awareness in the model on a word-to-word level. This finding is in line with earlier observations from text generated by C2C (Chapter 3.2; example reiterated in Table 5.2.

The novel regarded the modern Laboratory has a weaken-little director and many of them in 2012 to defeat in 1973 - or eviven of Artitagements.

Table 5.2: A sentence generated with C2C through sampling.

The model learns co-occurrences not just between words but between syntactic categories. This allows it to generate phrases that may be odd in meaning, like the one above, but mostly abide by the combinatorial properties of syntax within some temporal window.

Chapter 6

Suffixes Have Selectional Restrictions

C2C may be able to learn the selectional restrictions of derivational suffixes with respect to syntax. The experiment described below tests this hypothesis. The outcome of the experiment indicates that this morphological regularity is indeed captured by C2C.

6.1 Method

The main method used here was to measure and compare the probability of suffixes with different selectional restrictions across subsets of nominal, verbal and adjectival bases, as processed by C2C. In order to abstract away from familiar base-suffix combinations, seen during training, nonce bases were used.

6.1.1 Probability

The probability of a suffix in the context of a particular base was computed as the joint probability of its characters. Consider the following example of suffix *ion* attaching to base *edit*

$$p(ion|reun) = p(i|edit) \times p(o|editi) \times p(n|editio)$$

Since C2C is a wordless language model, $p(\cdot|base)$ is approximated from $p(\cdot|\mathbf{c})$ where $\mathbf{c}$ is the entire past.

The probability of a suffix in the context of a particular syntactic category was computed as the micro-average over all bases belonging to that category.

Base	Suffixes
Noun	crystale, algoritum, cosmony, landlough
Verb	underspire, restruct, actrace
Adjective	nucleent, transplet, orthouble

Table 6.1: Sample Nonce Bases

Base	Suffixes
Noun	ous, an, ic, ate, ary, hood, less, ish
Verb	ance, ment, ant, ory, ive, ion, able, ably
Adjective	ness, ity, en

Table 6.2: Syntactically Unambiguous Derivational Suffixes

6.1.2 Nonce Bases

Nonce bases were obtained by sampling from C2C and for every complete word (sequence delimited by a space or punctuation on both sides), checking whether it belonged in an English dictionary. Several restrictions were imposed on potential candidates: their probability had to be at most one standard deviation below the mean for real words (to ensure they weren't highly unlikely accidents of the sampling procedure), the probability of a following space character had to be at most one standard deviation below the mean for real words (to avoid prematurely finished words, such as *measu* and *experimen*), and the confidence of their POS tags had to be at most one standard deviation below the mean confidence with which tags of real words were assigned. In addition, nonce words had to be composed entirely of lowercase characters and couldn't end in a suffix (as certain suffixes included in the experiment only attach to base stems).

Some examples of nonce words from the final selection are shown in Table 6.1. An embedding was recorded for every nonce word that met these conditions by taking the hidden state of the language model at the end of the word.

6.1.3 Suffixes

The suffixes included in this experiment (listed in Table 6.2) were taken from Fabb (1988), one of the most extensive studies of the selectional restrictions of English derivational suffixes. Fabb discussed 43 suffixes, many of which attach to a base of two out of three available syntactic categories, e.g. *ize* attaches to both nouns, as in

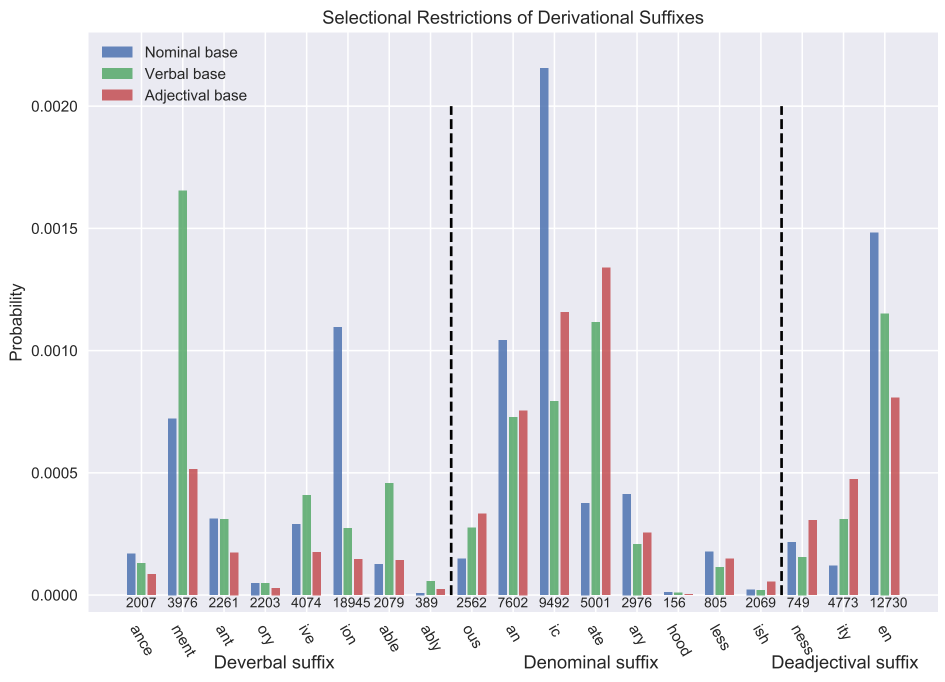


Figure 6.1: Suffix Probability Following Nominal, Verbal and Adjectival Bases. Suffixes are grouped according to their selectional restrictions, with dashed lines delimiting the three groups. Naive suffix frequency counts are reported at the bottom of the plot. Eleven out of nineteen suffixes obtained highest probability following the syntactic category that matched their selectional restrictions.

symbolize, and adjectives, as in *specialize*. The analysis of such syntactically ambiguous suffixes is complex since the frequency at which they attach to each base type should be taken into consideration, but such statistics are not readily available and require morphological parsing. For the purposes of the present study ambiguous suffixes were thus excluded and only the remaining nineteen suffixes were used.

6.2 Results

Figure 6.1 shows the results from the experiment. Eleven out of nineteen suffixes exhibit the expected behavior: suffixes *ment, ive, able, ably, an, ic, ary, hood, less, ness* and *ity* are more probable in the context of their corresponding syntactic base than in other contexts. Suffix *ment*, for instance, is more than twice as probable in the context of a verbal base than in the context of a nominal or an adjectival base. The fact that almost 70% of suffixes 'select' their correct bases, points to a linguistic awareness within C2C with respect to the selectional restrictions of suffixes. Had there been no such awareness, the ratio of success should have been close to 33%.

Despite the overall success of C2C in this respect, some suffixes show a definitive preference for the wrong base.

A further analysis of some of these cases shows that they don't necessarily counter the evidence for syntactic awareness within C2C. The discussion below is a speculation based on a qualitative analysis. An exhaustive quantitative analysis of the issue is outside the focus of this project, as it concerns linguistics beyond natural language processing.

6.2.1 Suffix *ion*

Denominal suffix *ion* should be most probable following a noun, but it shows a much stronger preference for nominal bases. Notice that *ion* is a noun-forming suffix: *communicate*$_{VERB}$ – *communication*$_{NOUN}$, *regress*$_{VERB}$ – *regression*$_{NOUN}$. It appears that from the perspective of C2C the syntactic category of such morphologically complex forms extends to their bases, e.g. C2C perceives the *populat* substring in *population* as nominal. This observation can be explained once again with reference to the high frequency of suffix *ion*.

Suffix *ion* occurred in 18945 words in the dataset. The frequency of its various bases in isolation, e.g. of *populate, regress*, etc., was estimated to be only 9755[1]. This shows that bases that can take suffix *ion* were seen more often with it than without it. As a consequence, C2C is biased to expect a noun when seeing one of these bases and may thus perceive the base itself as a nominal one.

The tag prediction evolution over *population* and *renewable* in Figures 6.2 and 6.3, show that this is indeed the case, by comparing a base suffixed with *ion* to a base suffixed with *able* (whose selectional restrictions, we know, were learned correctly). For both words C2T starts of predicting NOUN. For *renew* it switches to VERB, which is the correct tag for this base, and only upon seeing the suffix, progresses to the conclusive ADJ tag for *renewable*. For *population*, in contrast the prediction remains constant at NOUN, which is indeed the category of the word, as determined by the suffix.

[1]This count was obtained by striping the suffix off words and looking for an occurrence of the remaining material in isolation, followed by *e* and followed by *s/ es*, e.g. for *population* we counted the occurrences of *populat, populate, populates* and *populated*.

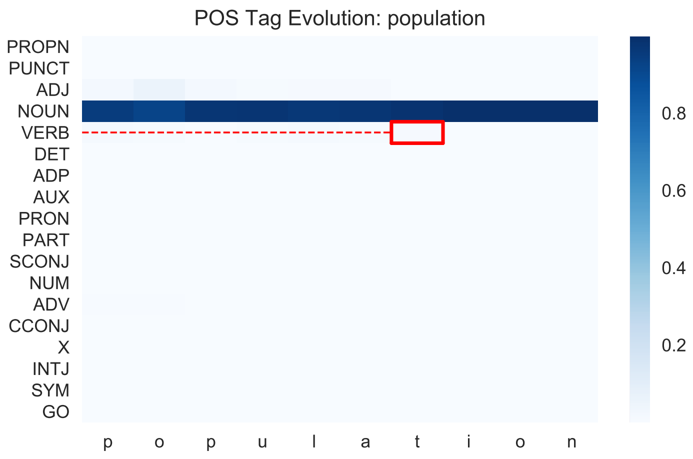


Figure 6.2: C2T Evolution: *population*. The red square points to the syntactic category of the base.

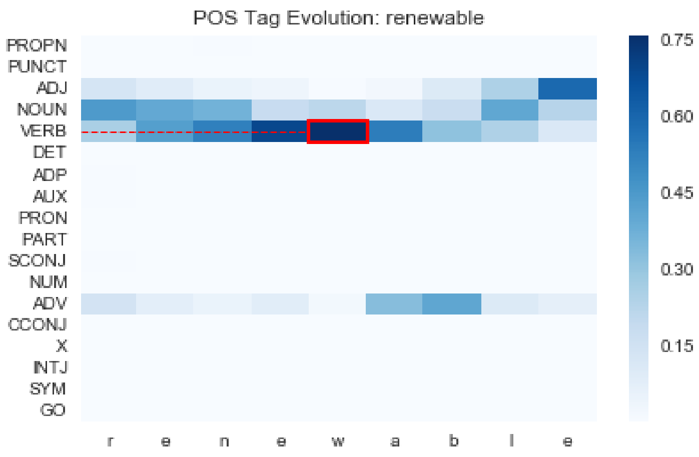


Figure 6.3: C2T Evolution: *renewable*. The red square points to the syntactic category of the base.

6.3 Conclusion

The experiment demonstrated that C2C is capable of capturing the selectional restrictions of derivational suffixes with respect to the syntactic category of bases, when there are no strong interfering factors. This outcome is in line with the findings of earlier chapters: a language model that is capable of encoding information concerning the syntactic and morphological structure of its input (here, suffixed words), is also capable of learning morpho-syntactic dependencies in the input.

Chapter 7

Prefixes Don't Change the POS of Bases

The non-category-changing property of Enlglish prefixes could potentially be captured by a character-level language model, either as a lexical rule relating to specific prefixes that have been observed often enough during training, or as a more general rule relating to any material prepended to a word. The following experiment tests this hypothesis. The results provide no support for the hypothesis.

7.1 Method

The hypothesis was tested by creating novel combinations of prefixes and bases and comparing the POS tags assigned to them with the POS of the bases alone.

7.1.1 Prefixes

The prefixes included in this experiment (listed in Table 7.1) were taken from Lehrer (1995)'s discussion of prefixation in English.

7.1.2 Bases and Their Modification

300 verbal, 300 adjectival and 300 nominal bases were selected from the UD English data, with the following criteria: candidate word shouldn't end in a suffix (as suffixes are a strong cue for POS tags and may dominate syntactic information encoded in the rest of the word); the tag assigned to a candidate word by W2T should match the gold label tag provided in the corpus.

Base	Prefixes
Noun	micro, macro, vice, arch, mini, maxi, counter, pre, sub, super, hyper, hypo, ultra, post, ante, pseudo, ex, multi, meta
Verb	re, mis, out, over, under, up, down, counter, pre, un, in
Adjective	anti, pro, extra, in, un, pre, sub, super, hyper, hypo, ultra, post, ante, pseudo, ex, multi, meta

Table 7.1: Derivational Prefixes

Category	Matching Tags
All	62.3
Noun	78.6
Verb	67.9
Adjective	40.4

Table 7.2: Rate of Matching Tags Across the Entire Experimental Set and Within Syntactic Categories.

Embeddings for the prefixed variants of a base were obtained by replacing the base with its modified versions in the original UD context, e.g. this was his **last**, this was his **unlast**, this was his **extralast**, etc., and recording the hidden state of C2C after processing the prefixed form. Words that existed in the training data (e.g. *outlast*) were excluded in order to focus on C2C's ability to generalize beyond forms seen during training. Based on these embeddings, modified forms were tagged for POS with W2T.

7.2 Results

Table 7.2 reports the percentage of matching syntactic category between base and prefixed form across the entire experimental set and within individual syntactic categories. While on average prefixation maintained the syntactic category of the base the majority of the time, a breakdown by syntactic category shows that this is only true for nouns and verbs, but not for adjectives. It is thus more reasonable to attribute the results to a property of bases rather than of prefixes. In this sense, the outcomes of the experiment point to a lack of awareness in C2C with respect to the non-category-changing property of prefixes.

Table 7.3 provides an insight into the property of bases that may account for the

Category	Bare	Prefixed -Matching	Prefixed - Mismatching
Noun	0.89	0.817	0.614
Verb	0.904	0.859	0.673
Adjective	0.821	0.743	0.724

Table 7.3: Confidence of POS Tags. The table reports the probability W2T assigned to bare bases and prefixed forms that match or don't match the syntactic category of their base.

observed results. Nominal and verbal bases were tagged more confidently than adjectival bases (column 1). This suggests that the former provide a stronger cue for their part of speech than the latter. In this sense, it is not surprising that prefixation to nouns and verbs caused a category change at a lower rate: the base is a more recent cue to begin with and when it is a strong one (as in the case of verbs and nouns), it suppresses most of the syntactic signal coming from the prefix.

Columns 2 and 3 in Table 7.3 further support this analysis. Margins between matching and mismatching tags for deverbal and denominal forms are magnitudes higher than for deadjectival ones (0.23 and 0.186 v. 0.019). W2T foresaw its potential mistakes more accurately when assigning a mismatching POS tag to a deverbal or denominal forms, because it was better aware of the syntactic category of the bases of these forms.

The tags assigned to denominal, deverbal and deadjectival bases are shown in Figure 7.1. Less than 1% of modified forms were assigned a tag outside of the five lexical classes, NOUN, PROPN, ADJ, ADV, VERB, so only these classes are included in the plot. Deadjectival and deverbal forms were often misidentified as nouns. Recall that the POS taggers were found to mostly default to a nominal category for rare words (see 5.2.1). The frequent choice of the NOUN tag here indicates that some prefixed form may be processed as rare words.

7.3 Conclusion

The outcome of this experiment indicates that the non-category changing property of prefixation was not learned by C2C. This finding is in line with the conclusion made earlier about prefixes not being segmented out as separate morphemes by C2C. It appears that the correct identification of a unit is a prerequisite for the learning of regularities concerning that unit.

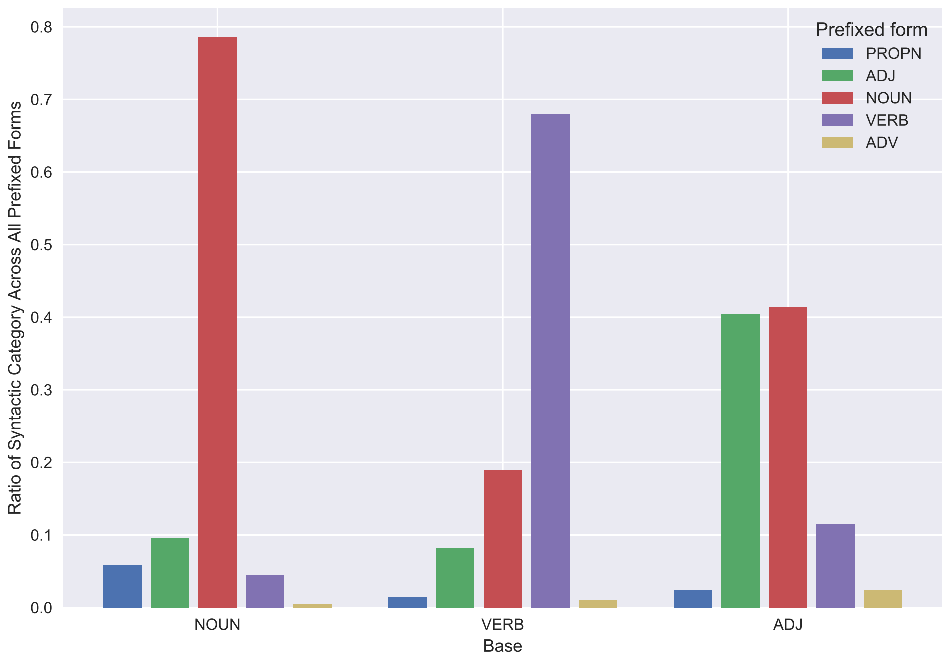


Figure 7.1: Syntactic Category of Novel Prefixed Forms. When prefixed, nouns and verbs mostly retain their status as such, while adjectives often switch to nouns.

Chapter 8

El Fin

This thesis presented an exploratory analysis of a ‘wordless’ character-to-character language model, aiming to identify the morpho-syntactic regularities captured by the model.

8.1 Summary of Results

A study of the internal structure of the language model showed that it learned surface regularities concerning characters and words. Evidence for semantic awareness was limited.

A specific unit was identified that demonstrated the model could recognize words as separate units. More evidence in this respect was found in an experiment embedding the language model into a system for morphological segmentation. The experiment also showed that the model did not inherently identify morpheme boundaries, but it could transfer knowledge on word edges to the task of morphological segmentation, whenever conditions allowed it. Consequently, it encoded boundaries following a base or a suffix better than those following a prefix.

An experiment embedding C2C into two different models for part-of-speech tagging provided further evidence for awareness on the word level and showed that the model was capable of encoding high levels of syntactic information.

Finally, two linguistic experiments tested the ability of the language model to learn regularities about individual morphemes, namely suffixes and prefixes. The model performed well on the experiment concerning the selectional restrictions of suffixes and rather poorly on the experiment concerning the non-category changing property of prefixes.

8.2 Conclusion

The first conclusion of this work is that morpheme boundaries are mainly learned by analogy with syntactic boundaries. Findings relating to the extremely frequent suffix *ion* illustrate that C2C was able to learn to identify purely morphological boundaries through generalization. Yet, a prerequisite for this generalization is that a morpho-syntactic boundary was also seen in the relevant position during training.

The second conclusion is that recognizing boundaries and by extension, the unit that lie between them, is essential to learning the regularities that concern these units. The nature of C2C is such that it could better identify suffixes than prefixes. The selectional restrictions of most derivational suffixes included in the study could thus be captured accurately, while generalizations about prefixes could not be learned.

8.3 Implications

These conclusions have strong implications with respect to the use of character-level LMs for languages other than English. English is the perfect candidate for language modeling, due to its fairly poor inflectional morphology. One consequence of this property of English is that a large portion of its words consist of a single morpheme, e.g. *eat, drink, dance*. Being trained on instances of these bare bases, C2C can learn to identify the boundaries in their suffixed forms, e.g. *eating, drinks, danced*. In Spanish, in contrast, every verb form is explicitly inflected and thus consists of at least two morphemes, a base and an inflectional suffix. A verbal base could thus never be followed by a space character in a training corpus. This means that C2C should not be able to learn the segmentation of any verbal or deverbal word in Spanish. Without access to the units of verbal morphology, the model would also be unable to learn any regularities from this domain of linguistics. This shortcoming should hold not just for C2C but for any character-level language model that processes input as a stream of characters without segmentation on the subword level.

Notice that this finding is in line with the results of Vania and Lopez (2017) showing an improvement in performance upon providing explicit morphological annotations during training. The authors conclude that the gap in performance signifies a limitation in the learning capabilities of a subword-level model with respect to morphology. The findings of this study can be seen as an explanation for this limitation.

One aspect in which a different character-level architecture would likely outperform C2C is the segmentation of prefixes. C2C cannot segment out prefixes due to the unidirectional processing of input. A bidirectional encoding of input, segmented into words in advance, such as the one used in Vania and Lopez (2017), should handle prefixes more appropriately. Notice, that the segmentation abilities and limitations of C2C highlight the importance of having the word-edge cue *even* in a model that explicitly segments input into words. By that we mean, encoding the word *drink* as *<BOW> d r i n k <EOW>* rather than as just *d r i n k*. Word-edge cues serve as place holders and allow the model to learn that an affix can be replaced with a word-edge and the other way around. This would allow the language model to learn to segment out prefixes and suffixes, to generate novel forms with them, and to learn generalizations about them. The issue remains, however, that this would likely only be possible for languages that, like English, have a poor inflectional morphology.

8.4 Future Research

The above implications could be empirically studied by performing analysis similar to the one presented here on a wider range of languages of richer morphology. If future research finds that it is indeed the case that character-level language models gain little access to the morphology of morphologically-rich languages, more elaborate means of encoding morphological information will need to be programmed into language models.

Subword-level models that operate on the level of morphemes could be seen as a step in this direction. Yet, morphological segmentation itself is not advanced enough to ensure that no additional noise is added to a system with model-predicted segmentations. The accuracy of *Morphessor*, the most widely used tool for morphological segmentation, ranges from an F1 score of .71 in an unsupervised English setting to 0.81 in a supervised setting Grönroos et al. (2014). This explains why morpheme-level models don't show a definitive edge over other approaches (Vania and Lopez, 2017). In this sense, it appears that future efforts should focus on advances in the field of morphological segmentation, as a mean of increasing the performance of morpheme-level language models.

Bibliography

Botha, J. A. and Blunsom, P. (2014). Compositional Morphology for Word Representations and Language Modelling. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1899–1907.

dos Santos, C. N. and Zadrozny, B. (2014). Learning Character-level Representations for Part-of-Speech Tagging. *Proceedings of the 31st International Conference on Machine Learning*, ICML-14(2011):1818–1826.

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

Fabb, N. (1988). English suffixation is constrained only by selectional restrictions. *Natural Language and Linguistic Theory*, 6(4):527–539.

Fung, P. (1995). A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 236–243. Association for Computational Linguistics.

Grönroos, S.-A., Virpioja, S., Smit, P., Kurimo, M., et al. (2014). Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology. In *COLING*, pages 1177–1185.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. (2015). Character-Aware Neural Language Models. In *CoRR abs/1508.06615*.

Lee, J., Cho, K., and Hofmann, T. (2016). Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Acl-2016*, pages 1693–1703.

Lehrer, A. (1995). Prefixes in english word formation. *Folia linguistica*, 29(1-2):133–148.

Ling, W., Luis, T., Marujo, L. L., Astudillo, R. F., Amir, S., Dyer, C., Black, A. W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L. L., and Luís, T. (2015). Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September):1520–1530.

Luong, M.-T., Socher, R., and Manning, C. D. (2013). Better Word Representations with Recursive Neural Networks for Morphology. *CoNLL-2013*, pages 104–113.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Mikolov, T., Yih, W.-T., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. *Proceedings of NAACL-HLT*, (June):746–751.

Qiu, S., Cui, Q., Bian, J., Gao, B., and Liu, T.-Y. (2014). Co-learning of Word Representations and Morpheme Representations. *Coling-2014*, pages 141–150.

Ruokolainen, T., Kohonen, O., Virpioja, S., and Kurimo, M. (2013). Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *CoNLL*, pages 29–37.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *NIPS*, pages 3104–3112.

Vania, C. and Lopez, A. (2017). From Characters to Words to in Between : Do We Capture Morphology ? *To appear in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.*

Wang, L., Cao, Z., Xia, Y., and de Melo, G. (2016). Morphological segmentation with window lstm neural networks. In *AAAI*, pages 2842–2848.