Computational Investigations of Derivational Morphology

Valentin Hofmann

St Catherine's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2023

Acknowledgements

This thesis would not exist without my supervisors, Janet Pierrehumbert and Hinrich Schütze. Janet's view on language was the reason why I moved to Oxford in the first place, and she was the one who encouraged me to delve into computational linguistics. I am deeply grateful for the impact Janet has had on my path as a researcher, but above all I will always take inspiration from Janet's unique combination of scientific curiosity, intellectual verve, and constructive spirit. Hinrich became my supervisor almost by coincidence, and I could never have imagined the profound ways in which working with him would transform me as a researcher. Hinrich taught me so many things, and I will forever be thankful for his guidance and support.

This thesis would also not have been possible without the generous funding from the Arts and Humanities Research Council and the German Academic Scholarship Foundation. Pursuing my doctoral studies at a stimulating place like Oxford was an enormous privilege for me, which I will continue to be grateful for.

I would like to thank the members of the Pierrehumbert Language Modelling Group and the Center for Information and Language Processing, especially Paul Röttger, Nora Kassner, Philipp Dufter, Alexandra Chronopoulou, and Leonie Weissweiler, for sharing ideas, discussing research and life, and above all—having fun. I am looking forward to following all the things they are going to do over the next years and decades, and I sincerely hope that our paths will cross again.

Last but not least, I would like to thank my parents Angelika and Gerald for their unwavering support, and my siblings Korbinian and Velia for moments of joy and fine foods. Hearty thanks also to my friends in Oxford and Munich, especially Jan, Arkadiusz, Otto, Rebecca, Fabian, and Nils, for keeping me sane. However, the heartiest thanks of all must go to my wife Felien, for standing by me throughout all the ups and downs of the DPhil. If there was one thing that cheered me up in busy times, it was the prospect to be with her. This thesis is dedicated to you, darling!

Abstract

The notion that it is difficult to make predictions about derivational morphology has been a recurring theme in morphological research over the last decades. It can be unclear whether a derivative exists at all, what a derivative means exactly, and which affix is used to form a derivative. The central goal of this thesis is to demonstrate that recent progress in natural language processing (NLP) allows for a fresh view on the (un-)predictability of derivational morphology.

Prior research in morphology has recognized semantic and extralinguistic factors as two key challenges for successfully predicting derivational morphology. The first set of papers contained in the thesis leverages novel methods from NLP and applies them to large-scale, socially-stratified datasets. I find that this computational approach results in substantially improved models, demonstrating that derivational morphology is predictable to a larger extent than previously thought.

A side result of the first part of the thesis is that tokenization (i.e., the way in which words are segmented) affects the capability of NLP systems to predict derivational morphology, raising the question whether it deteriorates performance on a larger scale. The second set of papers contained in the thesis shows that this is indeed the case. As a remedy, I devise tokenization strategies that are directly informed by morphology, with beneficial effects on performance.

On a wider scale, the results of this thesis suggest that NLP and deep learning more generally can greatly benefit linguistic research, a view that is still contested by many scholars in linguistics. At the same time, the thesis shows that even, or perhaps especially, in the age of large language models, linguistic insights continue to be relevant for the development of human language technology.

Contents

1 Introduction

Contents

1.1 Overview

Assume you are given the word *barp*, and you are told it is an English verb. What predictions can you make about its morphological status? In terms of inflectional morphology, most speakers of English will agree that *barp* will have four forms, *barp*, *barps*, *barped*, and *barping*. But what about derivational morphology? Will it have a corresponding noun, *barper*? If yes, what will it refer to? To a person who barps, as with *writer*? Or to an instrument that is used to barp, as with *computer*? Now assume you are told *barp* has a corresponding adjective, *barpable*. Does this fact make the existence of *barper* more likely? Also, what will be the nominalization of *barpable*? Perhaps *barpableness*? Or rather *barpability*?

The example of *barp* highlights a notion that has been a recurring theme in morphological research over the last decades (e.g., Bauer, 1988; Feldman, 1994; Stump, 2001; Gaeta, 2007; Haspelmath and Sims, 2010): in many respects, derivation seems less predictable than inflection. As the example of *barp* shows, it can be unclear whether a derivative exists at all, what a derivative means exactly, and which affix is used to form a derivative. These observations have lead some to consider predictability the key differentiating factor between inflection and derivation. For example, in his introduction to morphology, Bauer (2019, p. 109) suggests that inflection and derivation should be seen as "a proxy for highly frequent and predictable versus less frequent and less predictable."

The central goal of this thesis is to show that recent progress in natural language processing (NLP) allows for a fresh view on the (un-)predictability of derivational morphology. Specifically, the introduction of more fine-grained meaning representations and the collection of more diverse datasets have provided models and resources that can be used, I argue, to explicitly address some of the underlying factors causing the variability in derivation. The first half of the studies contained in the thesis demonstrates in various ways that derivational morphology is predictable to a larger extent than previously thought when choosing suitable methods and data.

Although not their original motivation, one finding emerging from the studies on predicting derivational morphology is directly related to current practices in NLP. Specifically, when tokenization (i.e., the way in which words are segmented) is linguistically ungrounded, it drastically affects the capability of NLP systems to predict derivational morphology. Does this deteriorate the performance of NLP systems on a larger scale? Inspired by the fruitful symbiosis of the fields of morphology and deep learning, which goes at least back to Rumelhart and McClelland (1985), I investigate this question and its implications in detail, in the second half of the comprised studies.

In the following, I will first provide an overview of the differences between inflection and derivation, with a particular focus on questions surrounding their predictability (§1.2). The section will highlight *semantic* and *extralinguistic* factors as two key challenges for successfully predicting derivational morphology. Then, I will sketch the recent developments in NLP that promise improvements precisely for handling semantic and

extralinguistic factors (§1.3). Based on these two sections, I will formulate the thesis objectives in greater detail and discuss their relevance for NLP and linguistics (§1.4). Finally, I will list the publications contained in this thesis (§1.5).

1.2 Predicting Inflection and Derivation

According to the traditional textbook definition (e.g., Bauer, 1988; Haspelmath and Sims, 2010; Bauer, 2019), inflectional morphology refers to the different word forms of a lexeme, whereas derivational morphology refers to the different lexemes of a word family. Thus, for the lexeme *build*, inflection would produce word forms such as *builds* and *built*, and derivation would produce related lexemes such as *builder* and *buildable*. Contrary to the dichotomous view implied by this definition, it is easy to find examples that do not fit squarely into either inflection or derivation such as the suffix *-ing* (cf. *they are building the house* vs. *the building is old*), leading many scholar to assume that the distinction between inflection and derivation is a continuum rather than a dichotomy (e.g., Bochner, 1992; Haspelmath and Sims, 2010; ten Hacken, 2014; Bauer, 2019). Still, there are several core axes along which inflection and derivation differ (or at least the ends of the continuum between inflection and derivation), and which are repeatedly discussed in the literature (Bauer, 1988; Feldman, 1994; Stump, 2001; Gaeta, 2007; Haspelmath and Sims, 2010; Bauer, 2019). In the following, I will focus on two axes that are particularly relevant for the predictability of inflection and derivation: productivity and form-meaning correspondence. Then, I will discuss how explicitly modeling semantic and extralinguistic factors might remove some of the unpredictability that derivational morphology displays along these two axes.

First, inflection is generally more productive than derivation. Thus, with regard to inflection, usually all morphosyntactically possible word forms of a lexeme exist (e.g., English verbs usually have a past tense form), a fact that is underscored by the notion of *inflectional paradigms* (Stump, 2015). There are cases of defective paradigms in inflection (e.g., most people are unwilling to produce a past tense form of *forego*), but they are generally rare and require special explanations (Daland et al., 2007; Baerman et al., 2010). In derivation, on the other hand, it is the rule rather than the exception that not all

morphosyntactically possible derivatives of a lexeme exist (e.g., *hard* has a corresponding causative verb, *harden*, but *cold* does not). It therefore does not come as a surprise that the question of whether derivational paradigms exist is highly debated in the literature (e.g., Bonami and Strnadová, 2019). The difference in productivity between inflection and derivation is closely related to the fact that inflection is obligatory and determined by syntactic needs, whereas the existence of derivatives is mainly driven by communicative goals. Put differently, morphosyntax imposes a set of required word forms on inflection, but it does not have such an effect on derivation. This aspect will be brought up again when discussing semantic and extralinguistic factors below.

A second key differentiating factor between inflection and derivation is that the relationship between form and meaning is more varied in derivation than inflection. On the one hand, derivational affixes tend to be highly polysemous, i.e., individual affixes can represent a number of related meanings (Lieber, 2019). For example, the English nominalizing suffix *-er* can express the meanings of agent (*driver*), instrument (*computer*), stimulus (*thriller*), quantity (*fiver*), and patient (*loaner*), *inter alia*. On the other hand, one and the same meaning can also be represented by several affixes, e.g., *-ity* and *-ness*. While such competing affixes are often not completely synonymous as in the case of *hyperactivity* and *hyperactiveness*, there are examples like *purity* and *pureness* or *exclusivity* and *exclusiveness* where a semantic distinction is more difficult to gauge (Bauer et al., 2013; Plag and Balling, 2020). Compared to derivation, the correspondence between form and meaning is in general less varied for inflection.

An observation relevant for the question of (un-)predictability is that lexical-semantic and extralinguistic factors play a particularly big role in derivation. With respect to lexical semantics, inflection does not typically add to the meaning of the base and, as mentioned above, mostly expresses abstract and syntactic meanings—in fact, some scholars regard the link to syntax as the key defining property of inflection (e.g., Anderson, 1992). By contrast, derivation adds lexical-semantic content to the meaning of the base. In English, this is particularly true for derivational prefixes such as *super-* or *under-* (Giraudo and Grainger, 2003; Beyersmann et al., 2015). Of course, the lexical-semantic meaning of a particular derivational affix might not be compatible with the meaning of every base, thus

constraining its productivity in the lexicon. Another semantic effect that is specific to derivation is lexicalization, i.e., the semantic changes that accompany the incorporation of new words into the lexicon, blurring the compositionality of their meaning (Brinton and Traugott, 2005). With respect to extralinguistic factors, there is growing evidence that derivation is characterized by a substantial degree of variation along dimensions such as gender (Needle and Pierrehumbert, 2018), social class (Säily, 2011), and register (Plag et al., 1999). For example, in the case of *-ity* and *-ness*, it has been shown that men tend to use *-ity* more productively than women (Säily, 2014, 2016).

Crucially, when not explicitly modeled, the described effects of semantic and extralinguistic factors might substantially blur findings about the predictability of derivation. For example, interactions between the base and affix meanings might be predictive of some of the gaps in derivation. Similarly, extralinguistic factors might often be predictive of which of a set of competing affixes is used to form a derivative (e.g., whether the nominalization of *exclusive* is *exclusivity* or *exclusiveness* for a particular speaker).

In the past, a key obstacle to testing these hypotheses was the unavailability of suitable methods and data that would allow to include semantic and extralinguistic factors in predictive models of derivation. However, as I will show in the next section, this has drastically changed over the last few years.

1.3 NLP to the Rescue?

Over the last decade, NLP has witnessed unprecedented advances, both in terms of the employed methods and in terms of the available datasets. These advances, I argue, have the potential of significantly improving the ways in which semantic and extralinguistic factors can be harnessed for predictive modeling.

A key ingredient of the recent progress in NLP was the development of deep learning methods for modeling the meaning of words. The theoretical foundation for these methods is distributionalism (Lenci, 2008; Turney and Pantel, 2010; Erk, 2012; Boleda, 2020), i.e., the hypothesis that lexical semantics can be successfully captured by modeling the statistical co-occurrence patterns of words in text. First famously formulated by Firth (1957, p. 11) as "You shall know a word by the company it keeps," the tenets

of distributionalism were the motivating factor behind early *count-based* models of word meaning developed in the 1990s such as Latent Semantic Analysis (Deerwester et al., 1990) and WordSpace (Schütze, 1992a,b). These models initialize meaning representations with co-occurrence counts, which are then typically compressed using methods such as Singular Value Decomposition. Crucially, count-based models are computationally expensive and cannot be scaled to large vocabularies of words, which limits their applicability. This situation changed with the advent of *prediction-based* models of word meaning in the 2010s such as Word2vec (Mikolov et al., 2013a,b,c) and GloVe (Pennington et al., 2014), which are based on neural networks. Prediction-based models provide two important advantages compared to count-based models: they are trained in an online fashion by iteratively predicting co-occurrences for individual words, which does not require the full co-occurrence matrix to be processed as a whole, making them computationally less expensive. Furthermore, prediction-based models empirically result in superior meaning representations (Baroni et al., 2014). While the initial prediction-based models represent each word as a single, static vector, more recent models, specifically language models such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), GPT-3 (Brown et al., 2020), and T5 (Raffel et al., 2020), represent each word as a set of contextualized vectors. This, combined with additional methodological improvements and a considerably larger parameter count, has established language models as the current state-of-the-art method for modeling lexical semantics in NLP (Vulić et al., 2020).

Most prior work on predicting derivational morphology (e.g., Cotterell et al., 2017) was based on word form alone. The fact that adequately modeling semantic effects is precisely one of the challenges discerned above begs the question whether the new NLP methods for representing meaning based on deep learning—specifically prediction-based word embeddings and language models—might allow for a more sound and at the same time (potentially) more successful approach.

A second aspect that has changed in NLP over the last few years, with the potential of enabling improved predictive models of derivation, is the availability of large-scale datasets with text from various extralinguistic contexts. In order to devise models of

derivation that take into account extralinguistic factors, large-scale datasets are indispensable, but they were previously unavailable. This has dramatically changed over the last two decades: the ubiquitous usage of the internet has resulted in massive amounts of (partially high-quality) linguistic data (Kilgarriff and Grefenstette, 2003), the full potential of which still has to be exploited by linguistic research. One resource of particular interest are discussion forums, where users form communities centered around certain topics (e.g., USENET, Reddit). The fact that users are clustered by their interests in these forums induces a natural (albeit latent) social structure (Olson and Neal, 2015; Datta et al., 2017; Kumar et al., 2018; Hofmann et al., 2022), which can be exploited for investigating the effects of social factors on derivation. Discussion forums have been used for studies on the influence of extralinguistic factors on language before (Altmann et al., 2011; Stewart and Eisenstein, 2018), but not with a focus on derivational morphology.

Thus, both recent advances in modeling meaning and the hitherto unprecedented availability of large-scale as well as socially-stratified datasets might have crucial implications for the predictive modeling of derivational morphology. However, this potential has so far not been exploited in existing work.

1.4 Thesis Objectives

Based on the insights from the two preceding sections, in the following I will formulate the objectives of this thesis in greater depth and discuss what makes them relevant for the fields of NLP and linguistics.

The first objective of this thesis is to push the envelope of predictive models for derivational morphology by leveraging new methods from NLP and applying them to large-scale, socially-stratified datasets. I hypothesize that such a computational approach allows for a better performance than existing models (i.e., a better fit with the actual derivational patterns in a language). If successful, this would be of high relevance for NLP and linguistics. First, focusing on NLP, the resulting models could be directly applied in scenarios in which derivational morphology is important (e.g., for processing or generating derivationally-rich text). For example, machine translation systems—both statistical (Minkov et al., 2007) and neural (Dalvi et al., 2017)—are known to struggle

with decoding complex morphology in the target language, suggesting that better methods for predicting derivational morphology could prove valuable. Second, with regard to linguistics, a success would suggest that the difference in predictability between inflection and derivation is smaller than commonly assumed, and that part of this assumption is due to the inadequate methodological apparatus employed in prior linguistic scholarship. More generally, if non-symbolic systems such as language models can be shown capable of acquiring derivational morphology sufficiently well, this has important implications for the question of whether derivational patterns should be explained as a result of symbolic rules, or rather as a result of analogical generalizations over stored exemplars—a topic that is still hotly contested in linguistics (e.g., Embick and Marantz, 2008; Arndt-Lappe, 2014) and affects both acquisition and processing. The specific aspects of the predictability of derivation that I address are:

– whether a certain derivative exists (chapter 3);

– whether a certain derivative will be created in the future (chapter 4);

– which affix is used to form a derivative (chapter 5).

Inspired by the historically fruitful relationship between morphology and deep learning, my thesis has a second objective: it aims to take the insights from the first chapters and distill them into concrete recommendations as well as methodological refinements for NLP models. If the insights lead to empirically measurable improvements of NLP systems, this would lend further support to the drawn conclusions and underscore the continued relevance of linguistic scholarship for NLP. The specific aspect that I address is tokenization for language models, i.e., segmenting the input text into a sequence of tokens known to the language model (chapters 5, 6, 7). Given that tokenization is the first step in processing text with all language models, and given that language models are the most widely used language technology today, improving tokenization can be beneficial for various subfields of NLP. At the same time, the very nature of tokenization (i.e., splitting words into smaller pieces) suggests that examining it through the lens of morphology might prove a worthy endeavor.

The two objectives of the thesis lead to two parts: a first part in which I explore how new NLP methods and datasets can improve our understanding of English derivational morphology and enable better predictive models (chapters 3, 4, 5), and a second part in which I examine how the insights from these experiments can in turn inform current practices in NLP (chapters 5, 6, 7). Chapter 5 covers aspects of both objectives and hence constitutes a natural transition between the two parts. Furthermore, I provide a discussion of related work in chapter 2 and a conclusion in chapter 8.

1.5 Publication List

All papers included in this thesis have been published in the proceedings of major NLP conferences (specifically, ACL and EMNLP). I am the first author on each paper, and more detailed authorship statements are provided as an appendix. All papers are left in their original preprint formatting. The publication list is as follows:

– Chapter 3: Valentin Hofmann, Hinrich Schütze, and Janet Pierrehumbert. 2020. A Graph Auto-encoder Model of Derivational Morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1127–1138.

– Chapter 4: Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020. Predicting the Growth of Morphological Families from Social and Linguistic Factors. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7273–7283.

– Chapter 5: Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020. DagoBERT: Generating Derivational Morphology with a Pretrained Language Model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3848–3861.

– Chapter 6: Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre Is Not Superb: Derivational Morphology Improves BERT's Interpretation of Complex Words. In *Proceedings of the 59th Annual Meeting of the Association*

for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP), pages 3594–3608.

– Chapter 7: Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An Embarrassingly Simple Method to Mitigate Undesirable Properties of Pretrained Language Model Tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 385–393.

2
Related Work

Contents

In the following, I will summarize prior work that is relevant for the research presented in this thesis. I will divide the chapter into four parts: I will first discuss different views on derivational morphology and the mental lexicon as proposed in linguistics. I will then sketch existing computational work on modeling derivational morphology. Next, I will describe the current state of research on tokenization for language models. Finally, given its key relevance for the research presented in chapter 4, I will give a brief overview of computational work on modeling lexical change in social media.

2.1 Derivatives and the Mental Lexicon

How is derivational morphology acquired and processed by humans? What is the cognitive status of derivatives? These questions have received a lot of attention in linguistics and are tightly intertwined with different assumptions about the structure of the mental lexicon, which can be thought of as a set of associations between meaning and form

(Pierrehumbert, 2012). In the following, I will center the discussion around the two most prominent positions that have emerged from this debate.

One classical view is that derivational morphology is based on *rules*, a form of generalization that results from language learners scanning the available data and distilling abstract knowledge about the contained linguistic patterns (e.g., Chomsky, 1965; Chomsky and Halle, 1968; Aronoff, 1976; Albright and Hayes, 2003; Embick and Marantz, 2008). As soon as the stage of rule deduction is completed, the data themselves are discarded, except for a stored inventory of morphemes. During processing, the rules operate on the stored morphemes to generate an output (e.g., a derivative). In this conceptualization of derivational morphology, the mental lexicon contains morphemes, but not the derivatives that are generated by the rules. Sublexical theories in psycholinguistics are similar to this account in assuming that all derivatives are decomposed into morphemes during lexical processing, and that the mental lexicon only consists of morphemes, not the derivatives that they build (Taft and Forster, 1975; Taft, 1994; Rastle et al., 2004).

Over the last few decades, evidence has been mounting that encountered derivatives are often stored as full forms in the mental lexicon (Bertram et al., 2000; Sonnenstuhl and Huth, 2002). This finding is in line with another classical view, specifically that derivational morphology is based on *analogy*, a form of generalization that results from language learners storing an extensive inventory of instances of linguistic patterns, without inferring any abstractions over these instances (e.g., Skousen, 1989; Nosofsky, 1990; Skousen, 1992; Daelemans and van den Bosch, 2005; Arndt-Lappe, 2014; Chandler, 2017; Ambridge, 2020). During processing, a similarity-driven algorithm is used to determine the output (e.g., a derivative) based on a comparison of the input with the stored instances. In this conceptualization of derivational morphology, the mental lexicon contains a large number of full-form derivatives.

Derivatives can differ in how complex they are perceived by individuals (Seidenberg and Gonnerman, 2000; Gonnerman and Andersen, 2002; Hay and Baayen, 2005). For example, *business* is reported as less complex than *happiness* (Hay and Baayen, 2005). It has been shown that an important factor influencing the perceived compositionality is the relation between the frequency of the derivative and the frequency of the base

(Hay, 2001): derivatives with a higher frequency than the base such as *business* tend to be rated as less complex than derivatives with a lower frequency than the base such as *happiness*. These findings suggest that the decomposability of derivatives is gradient, a notion that is acknowledged in many recent theories of the mental lexicon. For example, in dual-route models the mental lexicon contains both full-form derivatives alongside their constituent morphemes, with various factors such as frequency relationships having an influence on which representation is accessed first during processing (New et al., 2004; Kuperman et al., 2009; Virpioja et al., 2018). Similarly, there are models that assume multiple levels of representation as well as various forms of interaction between them (Rácz et al., 2015; Needle and Pierrehumbert, 2018; Rácz et al., 2020). In these models, sufficiently frequent complex words are stored in the mental lexicon together with representations that include their internal structure.

In addition to the general difference in productivity between inflection and derivation mentioned in the introduction (§1.2), individual derivational affixes also vary in how productive they are, i.e., how readily they can be used to create new derivatives (Plag, 1999). For example, while the suffix *-ness* can attach to practically all English adjectives, the suffix *-th* is much more limited in its scope of applicability. To capture such differences, various measures of morphological productivity have been proposed in quantitative linguistics (Baayen and Lieber, 1991; Baayen, 1992, 1993; Plag et al., 1999; Bauer, 2001; Baayen, 2005). In this thesis, I put a special emphasis on productive derivational morphology (i.e., I focus on derivational affixes such as *-ness* rather than derivational affixes such as *-th*). Productive derivational morphology is interesting since often the generated derivatives are likely or even guaranteed not to be contained in the mental lexicon but formed on the fly, which makes them particularly informative about the generalization mechanisms underlying derivation. The emphasis on productive derivational morphology has important implications for the datasets that I work with: rather than drawing on derivatives listed in lexical resources, many of which are established lexemes and hence less indicative of productivity (Plag et al., 1999; Bauer, 2001), I focus on derivatives extracted from social media. This makes it possible to mine derivatives from the very low end of the frequency spectrum (cf. §2.2).

2.2 Computational Derivational Morphology

Prior studies in computational linguistics and NLP that specifically address derivational morphology roughly fall into two groups: studies on modeling the meaning of derivatives and studies on predicting the form of derivatives. Here, I will mostly focus on work leveraging neural networks—older studies that are based on techniques from statistical NLP are summarized in Roark and Sproat (2007).

Out of the two groups, modeling the meaning of derivatives has arguably received the greater attention. The theoretical basis for most studies in this camp is the fact that in vector space models of word meaning (Turney and Pantel, 2010), derivational relations often correspond to coherent geometric relations (Lazaridou et al., 2013). Specifically, the geometric relation between the vectors of a base and a derivative can be formalized by means of vector addition or linear vector transformation (Baroni and Zamparelli, 2010), where the meaning of the derivational affix is represented by a vector or a matrix. Even though there are some inconsistencies among the studies, the additive model generally seems to outperform the transformational model (Kisselew et al., 2015; Padó et al., 2016) and is used in almost all studies. In terms of concrete models, the studies differ along two main axes. One distinction relates to whether character n-grams (Schütze, 1992b; dos Santos and Zadrozny, 2014; Kim et al., 2016; Bojanowski et al., 2017; Verwimp et al., 2017; Athiwaratkun et al., 2018) or morphemes (Lazaridou et al., 2013; Luong et al., 2013; Botha and Blunsom, 2014; Qiu et al., 2014; Kisselew et al., 2015; Bhatia et al., 2016; Padó et al., 2016) are used as the main unit of subword information. The models also differ in whether subword vectors are trained directly using count-based (Salle and Villavicencio, 2018) or prediction-based approaches (Luong et al., 2013; Botha and Blunsom, 2014; Qiu et al., 2014; dos Santos and Zadrozny, 2014; Bhatia et al., 2016; Kim et al., 2016; Bojanowski et al., 2017; Verwimp et al., 2017; Athiwaratkun et al., 2018), or whether (in the case of morpheme-based models) vectors are inferred from whole-word vector spaces, e.g., using the vector offset method (Lazaridou et al., 2013; Kisselew et al., 2015; Padó et al., 2016). Furthermore, models that learn subword vectors directly differ in whether the full word is included as a factor (Botha and Blunsom,

2014; Qiu et al., 2014; Bhatia et al., 2016; Bojanowski et al., 2017) or not (Schütze, 1992b; Luong et al., 2013; Kim et al., 2016; Verwimp et al., 2017; Athiwaratkun et al., 2018). Direct training of morpheme vectors makes it necessary to segment the corpus into morphemes using an external morphological analyzer such as Morfessor (Creutz and Lagus, 2002). Outside the taxonomy presented so far, there are also approaches that try to introduce morphological information into word embeddings as a post-processing step (Cotterell et al., 2016; Vulić et al., 2017).

The second group of studies, which address modeling the form of derivatives, is most closely related to the research presented in this thesis. I will therefore discuss the three studies in question—Cotterell et al. (2017), Vylomova et al. (2017), and Deutsch et al. (2018)—in greater detail.

Cotterell et al. (2017) approach the generation of derivatives as a purely form-based task. They work with derivational triples extracted from NomBank (Meyers et al., 2004) that consist of a base, a tag specifying a derivational transformation, and the resulting derivative (e.g., *ameliorate*, RESULT, *amelioration*) and train a system that takes the base as well as the tag as input and generates the derivative as output. More specifically, Cotterell et al. (2017) take inspiration from work on inflection generation and propose a sequence-to-sequence model consisting of an encoder and a decoder network: the encoder network is modeled as a bidirectional gated recurrent neural network that encodes the characters of the base as well as the tag (e.g., *a*, *m*, *e*, *l*, *i*, *o*, *r*, *a*, *t*, *e*, RESULT), and the decoder network is modeled as an attentional recurrent neural network that decodes the characters of the derivative (e.g., *a*, *m*, *e*, *l*, *i*, *o*, *r*, *a*, *t*, *i*, *o*, *n*). They find that the proposed neural sequence-to-sequence model clearly beats a baseline transducer while substantially lagging behind the performance of inflection generation systems.

Vylomova et al. (2017) consider the task of generating a derivative from its base and the sentence context. Thus, instead of a tag specifying the derivational transformation as in the study of Cotterell et al. (2017), systems need to leverage information present in the sentence context to come up with the correct form. To this aim, Vylomova et al. (2017) develop an encoder-decoder model: the encoder network consists of three bidirectional long short-term memory networks that encode the left sentence context, the right sentence

context, and the characters of the base, respectively, and the decoder network is a recurrent neural network with a copying mechanism. In their experiments, in which they examine verb nominalizations extracted from CELEX (Baayen et al., 1995), Vylomova et al. (2017) show that this model performs reasonably well but struggles to differentiate between competing suffixes (e.g., *-ation* vs. *-ment*). They also observe that presenting the part of speech of the derivative to the model improves performance.

Deutsch et al. (2018) follow the setup of Cotterell et al. (2017), i.e., they again approach the generation of derivatives as the task of mapping pairs of a base and a tag to derivatives, using the same set of derivational triples extracted from NomBank (Meyers et al., 2004). They build on the character-based sequence-to-sequence model proposed by Cotterell et al. (2017) but modify it in an important way: they add a component that models the generation of derivatives distributionally, specifically as tag-specific feed-forward neural networks that take as input a semantic representation of the base in the form of a Word2vec embedding (Mikolov et al., 2013a,b,c). Their final model combines the outputs of the character-based sequence-to-sequence model and the distributional model by means of a logistic regression. Deutsch et al. (2018) show that including semantic information in this way boosts the performance compared to the model proposed by Cotterell et al. (2017), especially for derivatives with rare suffixes.

One common characteristic of all three studies is that the considered derivatives, which are extracted from lexical resources (Baayen et al., 1995; Meyers et al., 2004), are mostly established lexemes (e.g., *activity*)—Deutsch et al. (2018) even include an explicit dictionary constraint into their model. By contrast, I put an emphasis on productively formed derivatives that show up in the lower end of the frequency spectrum and hence are typically not listed in lexical resources, which is one of the reasons why I focus on datasets extracted from social media. At the same time, the results of Deutsch et al. (2018) suggest that lexical-semantic information is beneficial for modeling derivational morphology—an insight that constitutes a key motivation for this thesis. In general, as underscored by the small number of relevant studies, predicting the form of derivatives has received little attention so far, and more large-scale studies are missing.

2.3 Tokenization for Language Models

The first step in NLP architectures using language models is to map text to a sequence of tokens corresponding to input embeddings. There are three basic algorithms used to accomplish this: byte-pair encoding (BPE; Gage, 1994; Sennrich et al., 2016), WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016), and Unigram (Kudo, 2018). BPE and Unigram are often deployed via SentencePiece, a tokenization library (Kudo and Richardson, 2018). All three tokenizers split text into a mixed sequence of words, subwords, and characters, which sets them apart from both word-level and character-level tokenizers. For example, BPE tokenizes the word *tokenize* into two tokens, *token-ize*. In the following, I will describe the functioning of BPE in greater detail, as it is the most widely used tokenizer today.

BPE (Gage, 1994; Sennrich et al., 2016) works by first creating a vocabulary of tokens based on a training corpus. All characters—or, in the case of byte-level BPE, bytes—occurring in the training corpus are used as the initial vocabulary of tokens. Then, the pair of adjacent tokens with the highest co-occurrence frequency in the training corpus is merged and added to the vocabulary. For example, if *i* and *s* have the highest co-occurrence frequency out of all token pairs in the training corpus, the pair is merged and replaced by the token *is*, which is added to the vocabulary. This procedure, which requires co-occurrence frequencies to be recomputed after each merge, is repeated until a desired vocabulary size (typically in the range of 30,000 to 50,000 tokens) is reached. The resulting merge rules are stored as an ordered list. When tokenizing new text, BPE starts by splitting each word into its characters or bytes and then follows the merge rules in the order in which they were found to generate the final tokenization.

The two other common tokenizers, WordPiece and Unigram, differ from BPE in terms of how they create the vocabulary of tokens and/or how they use the vocabulary of tokens to segment new text. For WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016), the step of vocabulary creation works in the same way as for BPE, with the sole difference that the co-occurrence frequencies of token pairs (e.g., *i* and *s*) are divided by the individual token frequencies to determine the next merge. However, in contrast with

BPE, WordPiece does not follow the merge rules when tokenizing new text but applies the vocabulary in a greedy left-to-right manner. Unigram (Kudo, 2018) deviates more strongly from the functioning of BPE: it starts with a large vocabulary of tokens and works by iteratively *removing* tokens, not *adding* tokens as in the case of BPE. Which token to remove is determined based on a language modeling loss.

It is important to note that while the exact algorithmic details differ, BPE, WordPiece, and Unigram all work by optimizing a segmentation model on a training corpus—a process in which morphology is not explicitly taken into account and only plays a role insofar as it is implicitly encoded in the training corpus statistics (i.e., co-occurrence frequencies). As a result, the segmentations produced by language model tokenizers can be in line with morphology (e.g., WordPiece segments *finalize* as *final-ize*), but they do not need to be (e.g., WordPiece segments *mobilize* as *mob-ili-ze*).

Several studies have examined how language models are affected by their tokenizers. Tan et al. (2020) show that tokenizing inflected words into stems and inflection symbols allows BERT (Devlin et al., 2019) to generalize better on non-standard inflections. Bostrom and Durrett (2020) pretrain RoBERTa (Liu et al., 2019) with different tokenization methods and find tokenizations that align more closely with morphology to perform better on a number of tasks. Ma et al. (2020) show that providing BERT with character-level information also leads to enhanced performance. Relatedly, studies from automatic speech recognition have demonstrated that morphological decomposition improves the perplexity of language models (Fang et al., 2015; Jain et al., 2020). Another strand of research has sought to improve language model tokenizers by training models from scratch (Clark et al., 2021; Si et al., 2021; Xue et al., 2021; Zhang et al., 2021) or modifying the tokenizer during finetuning, mostly by adding tokens and corresponding embeddings (Chau et al., 2020; Tan et al., 2020; Hong et al., 2021; Sachidananda et al., 2021). Furthermore, there has been work on improving tokenization by exploiting the probabilistic nature of tokenizers (Kudo, 2018; Provilkov et al., 2020; Cao and Rimell, 2021).

The research presented here differs from prior work in that it examines tokenization through the lens of derivational morphology. Given that both tokenization and derivational morphology concern the internal structure of words, this is a well-justified perspective,

but it requires access to the derivational segmentation of words. To this aim, I employ a simple affix stripping algorithm throughout the thesis. Since the affix stripping algorithm constitutes a key background methodology of the thesis, and since it provides for an instructive contrast compared to purely data-driven tokenizers such as BPE, I will describe its functioning in greater detail below.

Let A be a set of derivational affixes and S a set of stems. A and S can be chosen in various ways (e.g., by drawing upon linguistic resources for A). To determine the derivational segmentation of a word w, I use an iterative algorithm. Define the set B_1^A of w as the words that remain when one derivational affix from A is removed from w. For example, *unlockable* can be segmented as *un-lockable* and *unlock-able*, so $B_1^A(\textit{unlockable}) = \{\textit{lockable}, \textit{unlock}\}$ (I assume that *un* and *-able* are in A). I then iteratively create $B_{i+1}^A(w) = \bigcup_{b \in B_i^A(w)} B_1^A(b)$, i.e., I iteratively remove affixes from w. The algorithm stops as soon as $B_{i+1}^A(w) \cap S \neq \emptyset$. The element in this intersection, together with the used affixes from A, forms the derivational segmentation of w. If $|B_{i+1}^A(w) \cap S| > 1$ (rarely the case in practice), the element with the lowest number of suffixes is chosen. If there is no i such that $B_{i+1}^A(w) \cap S \neq \emptyset$, w does not have a derivational segmentation. The implementation of this algorithm is sensitive to most morpho-orthographic rules of English (Plag, 2003), e.g., when the suffix *-ize* is removed from *isotopize*, the resulting word is *isotope*, not *isotop*.

Compared to the tokenizers used in the context of language models, the affix stripping algorithm results in many words that cannot be segmented. However, if a word *can* be segmented in a derivational way as defined by A and S, the affix stripping algorithm guarantees to return the derivational segmentation, making it a highly valuable method for the research presented in this thesis.

2.4 Modeling Lexical Change in Social Media

Language change is most visible on the lexical level. New words attract attention and often become the subject of public discourse. Over the last few years, social media, where innovations are taking place at a much faster rate (Crystal, 2004), have become a central resource for studies on lexical change (Altmann et al., 2011; Stewart and Eisenstein,

2018). One central question in this field is what factors determine whether a word will survive in the lexicon of an online community. Usage frequency is a well-known factor that influences the evolution of a word at historical time scales (Pagel et al., 2007). Studies on lexical change in online groups have shown that this is also true for shorter time scales (Altmann et al., 2011; Stewart and Eisenstein, 2018). Another main factor is the dissemination of a word, i.e., how widely a word is spread across different social and linguistic contexts (Stewart and Eisenstein, 2018). Methodologically, most work in this camp, including the study presented in chapter 4, splits the time series of words into non-overlapping windows and tries to predict the dynamics of words in the second window from their behavior in the first window. An alternative approach—one that has been explored less—would be to predict changes in the dynamics of words by leveraging insights from research on anomaly and change point detection (e.g., Aminikhanghahi and Cook, 2017; Truong et al., 2020; Liu et al., 2023).

The studies mentioned so far focus on lexical change as it is reflected by token frequency. A separate line of work has instead looked at lexical change as it manifests by the meaning of words, i.e., lexical semantic change. To capture this phenomenon, various types of diachronic word embeddings have been proposed (Kim et al., 2014; Kulkarni et al., 2015; Hamilton et al., 2016; Bamler and Mandt, 2017; Rosenfeld and Erk, 2018; Rudolph and Blei, 2018; Yao et al., 2018; Gong et al., 2020), and they have already been applied to explore short-term lexical semantic shifts in social media (e.g., del Tredici et al., 2019a). The relevance of diachronic word embeddings is also reflected by the fact that lexical semantic change detection has become an established task in NLP (Kutuzov et al., 2018; Schlechtweg et al., 2018; Tahmasebi et al., 2018; Dubossarsky et al., 2019; Schlechtweg et al., 2019; Asgari et al., 2020; Pömsl and Lyapin, 2020; Pražák et al., 2020; Schlechtweg and Schulte im Walde, 2020; Schlechtweg et al., 2020).

3

A Graph Auto-encoder Model of Derivational Morphology

Contents

The first study approaches the predictability of derivational morphology by trying to predict whether a particular derivative, consisting of a base and a set of affixes, is morphologically well-formed, which is formalized as the task of performing link prediction on a graph representation of the mental lexicon. The study devises a graph auto-encoder model for this task that leverages meaning representations of both bases and affixes in the form of neural morpheme embeddings.

A Graph Auto-encoder Model of Derivational Morphology

Valentin Hofmann[*‡], **Hinrich Schütze**[‡], **Janet B. Pierrehumbert**[†*]

[*]Faculty of Linguistics, University of Oxford
[†]Department of Engineering Science, University of Oxford
[‡]Center for Information and Language Processing, LMU Munich
valentin.hofmann@ling-phil.ox.ac.uk

Abstract

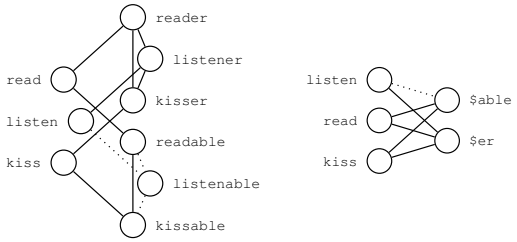
There has been little work on modeling the morphological well-formedness (MWF) of derivatives, a problem judged to be complex and difficult in linguistics (Bauer, 2019). We present a graph auto-encoder that learns embeddings capturing information about the compatibility of affixes and stems in derivation. The auto-encoder models MWF in English surprisingly well by combining syntactic and semantic information with associative information from the mental lexicon.

1 Introduction

A central goal of morphology is, as famously put by Aronoff (1976), "to tell us what sort of new words a speaker can form." This definition is tightly intertwined with the notion of morphological well-formedness (MWF). While non-existing morphologically well-formed words such as pro$computer$ism conform to the morphological patterns of a language and could be formed, non-existing morphologically ill-formed words such as proandism violate the patterns and are deemed impossible (Allen, 1979).

More recent research has shown that MWF is a gradient rather than binary property: non-existing words that conform to the morphological patterns of a language differ in how likely they are to be actually created by speakers (Pierrehumbert, 2012). This is particularly true in the case of derivational morphology, which is not obligatory and often serves communicative needs (Bauer, 2019). As a result, the degree of MWF of a non-existing derivative is influenced by a multitude of factors and judged to be hard to predict (Bauer, 2001).

In NLP, the lack of reliable ways to estimate the MWF of derivatives poses a bottleneck for generative models, particularly in languages exhibiting a rich derivational morphology; e.g., while inflected forms can be translated by generating morphologically corresponding forms in the target language (Minkov et al., 2007), generating derivatives is still a major challenge for machine translation systems (Sreelekha and Bhattacharyya, 2018). Similar problems exist in the area of automatic language generation (Gatt and Krahmer, 2018).



(a) Mental lexicon ($\mathcal{L}$) (b) Two-mode projection ($\mathcal{B}$)

Figure 1: Derivatives in the mental lexicon $\mathcal{L}$ (a) and the derivational graph (DG), their derivational projection $\mathcal{B}$ (b). Predicting whether a word is part of a derivational abstraction corresponds to predicting a single edge in the DG (dotted lines).

This study takes a first step towards computationally modeling the MWF of English derivatives. We present a derivational graph auto-encoder (DGA) that combines semantic and syntactic information with associative information from the mental lexicon, achieving very good results on MWF prediction and performing on par with a character-based LSTM at a fraction of the number of trainable parameters. The model produces embeddings that capture information about the compatibility of affixes and stems in derivation and can be used as pretrained input to other NLP applications.[1]

2 Derivational Morphology

2.1 Inflection and Derivation

Linguistics divides morphology into inflection and derivation. While inflection refers to the different

[1]We make all our code and data publicly available at https://github.com/valentinhofmann/dga.

word forms of a lexeme, e.g., `listen`, `listens`, and `listened`, derivation refers to the different lexemes of a word family, e.g., `listen`, `listener`, and `listenable`. There are several differences between inflection and derivation, some of which are highly relevant for NLP.

Firstly, while inflection is obligatory and determined by syntactic needs, the existence of derivatives is mainly driven by communicative goals, allowing to express a varied spectrum of meanings (Acquaviva, 2016). Secondly, derivation can produce a larger number of new words than inflection since it is iterable (Haspelmath and Sims, 2010); derivational affixes can be combined, in some cases even recursively (e.g., `post$post$modern$ism`). However, morphotactic constraints restrict the ways in which affixes can be attached to stems and other affixes (Hay and Plag, 2004); e.g., the suffix `$less` can be combined with `$ness` (`atom$less$ness`) but not with `$ity` (`atom$less$ity`).

The semantic and formal complexity of derivation makes predicting the MWF of derivatives more challenging than the MWF of inflectional forms (Anshen and Aronoff, 1999; Bauer, 2019). Here, we model the MWF of derivatives as the likelihood of their existence in the mental lexicon.

2.2 Derivatives in the Mental Lexicon

How likely a derivative is to exist is influenced by various factors (Bauer, 2001; Pierrehumbert and Granell, 2018). In this study, we concentrate on the role of the structure of the mental lexicon.

The mental lexicon can be thought of as a set of associations between meaning m and form f, i.e., words, organized in a network, where links correspond to shared semantic and phonological properties (see Pierrehumbert (2012) for a review). Since we base our study on textual data, we will treat the form of words orthographically rather than phonologically. We will refer to the type of information conveyed by the cognitive structure of the mental lexicon as *associative* information.

Sets of words with similar semantic and formal properties form clusters in the mental lexicon (Alegre and Gordon, 1999). The semantic and formal properties reinforced by such clusters create abstractions that can be extended to new words (Bybee, 1995). If the abstraction hinges upon a shared derivational pattern, the effect of such an extension is a new derivative. The extent to which a word conforms to the properties of the cluster influences how likely the abstraction (in our case a derivational pattern) is to be extended to that word. This is what is captured by the notion of MWF.

2.3 Derivational Graphs

The main goal of this paper is to predict the MWF of morphological derivatives (i.e., how likely is a word to be formed as an extension of a lexical cluster) by directly leveraging associative information. Since links in the mental lexicon reflect semantic and formal similarities of various sorts, many of which are not morphological (Tamariz, 2008), we want to create a distilled model of the mental lexicon that only contains derivational information. One way to achieve this is by means of a derivational projection of the mental lexicon, a network that we call the *Derivational Graph* (DG).

Let $\mathcal{L} = (\mathcal{W}, \mathcal{Q})$ be a graph of the mental lexicon consisting of a set of words $\mathcal{W}$ and a set of links between the words $\mathcal{Q}$. Let $\mathcal{W}_a \subset \mathcal{W}$ be a set of words forming a fully interconnected cluster in $\mathcal{L}$ due to a shared derivational pattern a. We define $\mathcal{S}_a$ as the set of stems resulting from stripping off a from the words in $\mathcal{W}_a$ and $\mathcal{R}_a = \{(s, a)\}_{s \in \mathcal{S}_a}$ as the corresponding set of edges between the stems and the shared derivational pattern. We then define the two-mode derivational projection $\mathcal{B}$ of $\mathcal{L}$ as the Derivational Graph (DG) where $\mathcal{B} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \bigcup_a (\mathcal{S}_a \cup \{a\})$ and $\mathcal{E} = \bigcup_a \mathcal{R}_a$. Figure 1 gives an example of $\mathcal{L}$ and DG (= $\mathcal{B}$).

The DG is a bipartite graph whose nodes consist of stems $s \in \mathcal{S}$ with $\mathcal{S} = \bigcup_a \mathcal{S}_a$ and derivational patterns $a \in \mathcal{A}$ with $\mathcal{A} = \bigcup_a \{a\}$. The derivational patterns are sequences of affixes such as `re$$ize$ate$ion` in the case of `revitalization`. The cognitive plausibility of this setup is supported by findings that affix groups can trigger derivational generalizations in the same way as individual affixes (Stump, 2017, 2019).

We define $\mathbf{B} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ to be the adjacency matrix of $\mathcal{B}$. The degree of an individual node n is $d(n)$. We further define $\Gamma^1(n)$ as the set of one-hop neighbors and $\Gamma^2(n)$ as the set of two-hop neighbors of n. Notice that $\Gamma^1(s) \subseteq \mathcal{A}$, $\Gamma^1(a) \subseteq \mathcal{S}$, $\Gamma^2(s) \subseteq \mathcal{S}$, and $\Gamma^2(a) \subseteq \mathcal{A}$ for any s and a since the DG is bipartite.

The advantage of this setup of DGs is that it abstracts away information not relevant to derivational morphology while still allowing to interpret results in the light of the mental lexicon. The cre-

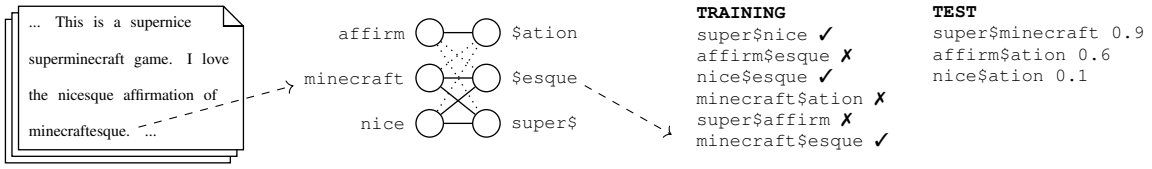


Figure 2: Experimental setup. We extract DGs from Reddit and train link prediction models on them. In the shown toy example, the derivatives `super$minecraft` and `affirm$ation` are held out for the test set.

ation of a derivative corresponds to a new link between a stem and a derivational pattern in the DG, which in turn reflects the inclusion of a new word into a lexical cluster with a shared derivational pattern in the mental lexicon.

3 Experimental Data

3.1 Corpus

We base our study on data from the social media platform Reddit.[2] Reddit is divided into so-called *subreddits* (SRs), smaller communities centered around shared interests. SRs have been shown to exhibit community-specific linguistic properties (del Tredici and Fernández, 2018).

We draw upon the Baumgartner Reddit Corpus, a collection of publicly available comments posted on Reddit since 2005.[3] The preprocessing of the data is described in Appendix A.1. We examine data in the SRs r/cfb (cfb – college football), r/gaming (gam), r/leagueoflegends (lol), r/movies (mov), r/nba (nba), r/nfl (nfl), r/politics (pol), r/science (sci), and r/technology (tec) between 2007 and 2018. These SRs were chosen because they are of comparable size and are among the largest SRs (see Table 1). They reflect three distinct areas of interest, i.e., sports (cfb, nba, nfl), entertainment (gam, lol, mov), and knowledge (pol, sci, tec), thus allowing for a multifaceted view on how topical factors impact MWF: seeing MWF as an emergent property of the mental lexicon entails that communities with different lexica should differ in what derivatives are most likely to be created.

3.2 Morphological Segmentation

Many morphologically complex words are not decomposed into their morphemes during cognitive processing (Sonnenstuhl and Huth, 2002). Based on experimental findings in Hay (2001), we segment a morphologically complex word only if the stem has a higher token frequency than the derivative (in a given SR). Segmentation is performed by means of an iterative affix-stripping algorithm introduced in Hofmann et al. (2020) that is based on a representative list of productive prefixes and suffixes in English (Crystal, 1997). The algorithm is sensitive to most morpho-orthographic rules of English (Plag, 2003): when `$ness` is removed from `happi$ness`, e.g., the result is `happy`, not `happi`. See Appendix A.2. for details.

The segmented texts are then used to create DGs as described in Section 2.3. All processing is done separately for each SR, i.e., we create a total of nine different DGs. Figure 2 illustrates the general experimental setup of our study.

SR	n_w	n_t	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{E}\rvert$
cfb	475,870,562	522,675	10,934	2,261	46,110
nba	898,483,442	801,260	13,576	3,023	64,274
nfl	911,001,246	791,352	13,982	3,016	64,821
gam	1,119,096,999	1,428,149	19,306	4,519	107,126
lol	1,538,655,464	1,444,976	18,375	4,515	104,731
mov	738,365,964	860,263	15,740	3,614	77,925
pol	2,970,509,554	1,576,998	24,175	6,188	143,880
sci	277,568,720	528,223	11,267	3,323	58,290
tec	505,966,695	632,940	11,986	3,280	63,839

Table 1: SR statistics. n_w: number of tokens; n_t: number of types; $|\mathcal{S}|$: number of stem nodes; $|\mathcal{A}|$: number of affix group nodes; $|\mathcal{E}|$: number of edges.

4 Models

Let W be a Bernoulli random variable denoting the property of being morphologically well-formed. We want to model $P(W|d, C_r) = P(W|s, a, C_r)$, i.e., the probability that a derivative d consisting of stem s and affix group a is morphologically well-formed according to SR corpus C_r.

Given the established properties of derivational morphology (see Section 2), a good model of $P(W|d, C_r)$ should include both semantics and formal structure,

$$P(W|d, C_r) = P(W|m_s, f_s, m_a, f_a, C_r), \quad (1)$$

where m_s, f_s, m_a, f_a, are meaning and form (here

[2] `reddit.com`
[3] `files.pushshift.io/reddit/comments`

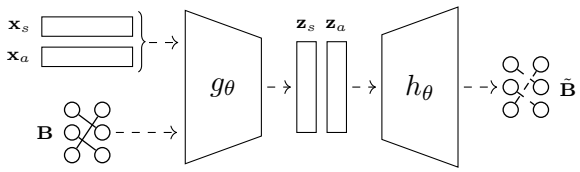


Figure 3: DGA model architecture. The DGA takes as input an adjacency matrix $\mathbf{B}$ and additional feature vectors $\mathbf{x}_s$ and $\mathbf{x}_a$ and learns embeddings $\mathbf{z}_s$ and $\mathbf{z}_a$.

modeled orthographically, see Section 2.2) of the involved stem and affix group, respectively. The models we examine in this study vary in which of these features are used, and how they are used.

4.1 Derivational Graph Auto-encoder

We model $P(W|d, C_r)$ by training a graph auto-encoder (Kipf and Welling, 2016, 2017) on the DG $\mathcal{B}$ of each SR. The graph auto-encoder attempts to reconstruct the adjacency matrix $\mathbf{B}$ (Section 2.3) of the DG by means of an encoder function g_θ and a decoder function h_θ, i.e., its basic structure is

$$\tilde{\mathbf{B}} = h_\theta\left(g_\theta\left(\mathbf{B}\right)\right), \tag{2}$$

where $\tilde{\mathbf{B}}$ is the reconstructed version of $\mathbf{B}$. The specific architecture we use (see Figure 3), which we call a *Derivational Graph Auto-encoder* (DGA), is a variation of the bipartite graph auto-encoder (van den Berg et al., 2018).

Encoder. The encoder g_θ takes as one of its inputs the adjacency matrix $\mathbf{B}$ of the DG $\mathcal{B}$. This means we model f_s and f_a, the stem and affix group forms, by means of the associative relationships they create in the mental lexicon. Since a DG has no information about semantic relationships between nodes within $\mathcal{S}$ and $\mathcal{A}$, we reintroduce meaning as additional feature vectors $\mathbf{x}_s, \mathbf{x}_a \in \mathbb{R}^n$ for m_s and m_a, stem and affix group embeddings that are trained separately on the SR texts. The input to g_θ is thus designed to provide complementary information: associative information ($\mathbf{B}$) and semantic information ($\mathbf{x}_s$ and $\mathbf{x}_a$).

For the encoder to be able to combine the two types of input in a meaningful way, the choice of g_θ is crucial. We model g_θ as a graph convolutional network (Kipf and Welling, 2016, 2017), providing an intuitive way to combine information from the DG with additional information. The graph convolutional network consists of L convolutional layers. Each layer (except for the last one) performs two steps: message passing and activation.

During the message passing step (Dai et al., 2016; Gilmer et al., 2017), transformed versions of the embeddings $\mathbf{x}_s$ and $\mathbf{x}_a$ are sent along the edges of the DG, weighted, and accumulated. We define $\Gamma^1_+(s) = \Gamma^1(s) \cup \{s\}$ as the set of nodes whose transformed embeddings are weighted and accumulated for a particular stem s. $\Gamma^1_+(s)$ is extracted from the adjacency matrix $\mathbf{B}$ and consists of the one-hop neighbors of s and s itself. The message passing propagation rule (Kipf and Welling, 2016, 2017) can then be written as

$$\mathbf{m}_s^{(l)} = \sum_{n \in \Gamma^1_+(s)} \frac{\mathbf{x}_n^{(l-1)} \mathbf{W}^{(l)}}{\sqrt{|\Gamma^1_+(s)||\Gamma^1_+(n)|}}, \tag{3}$$

where $\mathbf{W}^{(l)}$ is the trainable weight matrix of layer l, $\mathbf{x}_n^{(l-1)}$ is the embedding of node n from layer $l-1$ with $\mathbf{x}_n^{(0)} = \mathbf{x}_n$, and $\sqrt{|\Gamma^1_+(s)||\Gamma^1_+(n)|}$ is the weighting factor. The message passing step is performed analogously for affix groups. The matrix form of Equation 3 is given in Appendix A.3.

Intuitively, a message passing step takes embeddings of all neighbors of a node and the embedding of the node itself, transforms them, and accumulates them by a normalized sum. Given that the DG $\mathcal{B}$ is bipartite, this means for a stem s that the normalized sum contains $d(s)$ affix group embeddings and one stem embedding (and analogously for affix groups). The total number of convolutional layers L determines how far the influence of a node can reach. While one convolution allows nodes to receive information from their one-hop neighbors (stems from affix groups they co-occur with and vice versa), two convolutions add information from the two-hop neighbors (stems from stems co-occurring with the same affix group and vice versa), etc. (see Figure 4).

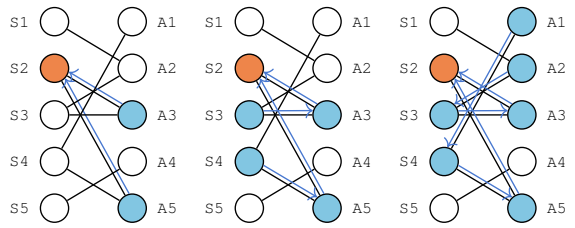
During the activation step, the output of the convolutional layer l for a particular stem s is

$$\mathbf{x}_s^{(l)} = \mathrm{ReLU}\left(\mathbf{m}_s^{(l)}\right), \tag{4}$$

where $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$ is a rectified linear unit (Nair and Hinton, 2010). The final output of the encoder is

$$\mathbf{z}_s = \mathbf{m}_s^{(L)}, \tag{5}$$

i.e., there is no activation in the last layer. The activation step is again performed analogously for affix groups. $\mathbf{z}_s$ and $\mathbf{z}_a$ are representations of s and a enriched with information about the semantics of nodes in their DG neighborhood.



(a) $L = 1$ (b) $L = 2$ (c) $L = 3$

Figure 4: Influence of L, the number of convolutional layers, on message passing. The blue nodes illustrate neighbors whose messages can be received by the orange node under varying L.

Decoder. We model the decoder as a simple bilinear function,

$$h_\theta\left(\mathbf{z}_s, \mathbf{z}_a\right) = \sigma\left(\mathbf{z}_s^\top \mathbf{z}_a\right), \tag{6}$$

where σ is the sigmoid and $\mathbf{z}_s$ and $\mathbf{z}_a$ are the outputs of the encoder.[4] We set $P(W|d, C_r) = h_\theta\left(\mathbf{z}_s, \mathbf{z}_a\right)$ and interpret this as the probability that the corresponding edge in a DG constructed from a corpus drawn from the underlying distribution exists. The resulting matrix $\tilde{\mathbf{B}}$ in Equation 2 is then the reconstructed adjacency matrix of DG.

Notice that the only trainable parameters of the DGA are the weight matrices $\mathbf{W}^{(l)}$. To put the performance of the DGA into perspective, we compare against four baselines, which we present in decreasing order of sophistication.

4.2 Baseline 1: Character-based Model (CM)

We model $P(W|d, C_r)$ as $P(W|f_s, f_a, C_r)$ using a character-based model (CM), i.e., as opposed to the DGA, f_s and f_a are modeled directly by means of their orthographic form. This provides the CM with phonological information, a central predictor of MWF (see Section 2.2). CM might also learn semantic information during training, but it is not directly provided with it. Character-based models show competitive results on derivational tasks (Cotterell et al., 2017; Vylomova et al., 2017; Deutsch et al., 2018), a good reason to test their performance on MWF prediction.

We use two one-layer bidirectional LSTMs to encode the stem and affix group into a vector $\mathbf{o}$ by concatenating the last hidden states from both LSTM directions $\overrightarrow{\mathbf{h}}_s$, $\overleftarrow{\mathbf{h}}_s$, $\overrightarrow{\mathbf{h}}_a$, and $\overleftarrow{\mathbf{h}}_a$,

$$\mathbf{o} = [\overrightarrow{\mathbf{h}}_s \oplus \overleftarrow{\mathbf{h}}_s \oplus \overrightarrow{\mathbf{h}}_a \oplus \overleftarrow{\mathbf{h}}_a], \tag{7}$$

where $\oplus$ denotes concatentation. $\mathbf{o}$ is then fed into a two layer feed-forward neural network with a ReLU non-linearity after the first layer.[5] The activation function after the second layer is σ.

4.3 Baseline 2: Neural Classifier (NC)

We model $P(W|d, C_r)$ as $P(W|m_s, m_a, C_r)$ using a neural classifier (NC) whose architecture is similar to the auto-encoder setup of the DGA.

Similarly to the DGA, m_s and m_a are modeled by means of stem and affix group embeddings trained separately on the SRs. The first encoder-like part of the NC is a two-layer feed-forward neural network with a ReLU non-linearity after the first layer. The second decoder-like part of the NC is an inner-product layer as in the DGA. Thus, the NC is identical to the DGA except that it does not use associative information from the DG via a graph convolutional network; it only has information about the stem and affix group meanings.

4.4 Baseline 3: Jaccard Similarity (JS)

We model $P(W|d, C_r)$ as $P(W|f_s, f_a, C_r)$. Like in the DGA, we model the stem and affix group forms by means of the associative relationships they create in the mental lexicon. Specifically, we predict links without semantic information.

In feature-based machine learning, link prediction is performed by defining similarity measures on a graph and ranking node pairs according to these features (Liben-Nowell and Kleinberg, 2003). We apply four common measures, most of which have to be modified to accommodate the properties of bipartite DGs. Here, we only cover the best performing measure, Jaccard similarity (JS). JS is one of the simplest graph-based similarity measures, so it is a natural baseline for answering the question: how far does simple graph-based similarity get you at predicting MWF? See Appendix A.4 for the other three measures.

The JS score of an edge (s, a) is traditionally defined as

$$\zeta_{JS}(s, a) = \frac{|\Gamma^1(s) \cap \Gamma^1(a)|}{|\Gamma^1(s) \cup \Gamma^1(a)|}. \tag{8}$$

However, since $\Gamma^1(s) \cap \Gamma^1(a) = \emptyset$ for any s and a (the DG is bipartite), we redefine the set of common neighbors of two nodes n and m, $\Gamma_\cap(n, m)$, as $\Gamma^2(n) \cap \Gamma^1(m)$, i.e., the intersection of the two-hop neighbors of n and the one-hop neighbors of

[4] Besides the simple dot-product decoder, we also implemented a bilinear decoder with $h(\mathbf{z}_s, \mathbf{z}_a) = \sigma(\mathbf{z}_s^\top \mathbf{Q} \mathbf{z}_a)$, where $\mathbf{Q}$ is a trainable weight matrix. However, the model performed significantly worse.

[5] We also experimented with only one layer, but it performed considerably worse.

m, and analogously $\Gamma_{\cup}(n, m)$ as $\Gamma^2(n) \cup \Gamma^1(m)$. Since these are asymmetric definitions, we define

$$\zeta_{JS}(s, a) = \frac{|\Gamma_{\cap}(s, a)|}{|\Gamma_{\cup}(s, a)|} + \frac{|\Gamma_{\cap}(a, s)|}{|\Gamma_{\cup}(a, s)|} \quad (9)$$

JS assumes that a stem that is already similar to a lexical cluster in its derivational patterns is more likely to become even more similar to the cluster than a less similar stem.

4.5 Baseline 4: Bigram Model (BM)

We again model $P(W|d, C_r)$ as $P(W|f_s, f_a, C_r)$, leaving aside semantic information. However, in contrast to JS, this model implements the classic approach of Fabb (1988), according to which pairwise constraints on affix combinations, or combinations of a stem and an affix, determine the allowable sequences. Taking into account more recent results on morphological gradience, we do not model these selection restrictions with binary rules. Instead, we use transition probabilities, beginning with the POS of the stem s and working outwards to each following suffix $a^{(s)}$ or preceding prefix $a^{(p)}$. Using a simple bigram model (BM), we can thus calculate the MWF of a derivative as

$$P(W|d, C_r) = P(a^{(s)}|s) \cdot P(a^{(p)}|s), \quad (10)$$

where $P(a^{(s)}|s) = P(a_1^{(s)}|s) \prod_{i=2}^{n} P(a_i^{(s)}|a_{i-1}^{(s)})$ is the probability of the suffix group conditioned on the POS of the stem. $P(a^{(p)}|s)$ is defined analogously for prefix groups.

5 Experiments

5.1 Setup

We train all models on the nine SRs using the same split of $\mathcal{E}$ into training ($n_{train}^{(p)} = 0.85 \cdot |\mathcal{E}|$), validation ($n_{val}^{(p)} = 0.05 \cdot |\mathcal{E}|$), and test ($n_{test}^{(p)} = 0.1 \cdot |\mathcal{E}|$) edges. For validation and test, we randomly sample $n_{val}^{(n)} = n_{val}^{(p)}$ and $n_{test}^{(n)} = n_{test}^{(p)}$ non-edges $(s, a) \notin \mathcal{E}$ as negative examples such that both sets are balanced (0.5 positive, 0.5 negative).

For training, we sample $n_{train}^{(n)} = n_{train}^{(p)}$ non-edges $(s, a) \notin \mathcal{E}$ in every epoch (i.e., the set of sampled non-edges changes in every epoch). Nodes are sampled according to their degree with $P(n) \propto d(n)$, a common strategy in bipartite link prediction (Chen et al., 2017). We make sure non-edges sampled in training are not in the validation or test sets. During the test phase, we rank all edges according to their predicted scores.

Model	n_p
DGA+	30,200
DGA	20,200
CM	349,301
NC+	30,200
NC	20,200

Table 2: Number of trainable parameters for neural models. n_p: number of trainable parameters.

We evaluate the models using average precision (AP) and area under the ROC curve (AUC), two common evaluation measures in link prediction that do not require a decision threshold. AP emphasizes the correctness of the top-ranked edges (Su et al., 2015) more than AUC.

5.2 Training Details

DGA, DGA+: We use binary cross entropy as loss function. Hyperparameter tuning is performed on the validation set. We train the DGA for 600 epochs using Adam (Kingma and Ba, 2015) with a learning rate of 0.01.[6] We use $L = 2$ hidden layers in the DGA with a dimension of 100. For regularization, we apply dropout of 0.1 after the input layer and 0.7 after the hidden layers. For $\mathbf{x}_s$ and $\mathbf{x}_a$, we use 100-dimensional GloVe embeddings (Pennington et al., 2014) trained on the segmented text of the individual SRs with a window size of 10. These can be seen as GloVe variants of traditional morpheme embeddings as proposed, e.g., by Qiu et al. (2014), with the sole difference that we use affix groups instead of individual affixes. For training the embeddings, derivatives are segmented into prefix group, stem, and suffix group. In the case of both prefix and suffix groups, we add prefix and suffix group embeddings.

Since the window size impacts the information represented by the embeddings, with larger windows tending to capture topical and smaller windows morphosyntactic information (Lison and Kutuzov, 2017), we also train the DGA with 200-dimensional embeddings consisting of concatenated 100-dimensional embeddings trained with window sizes of 10 and 1, respectively (DGA+).[7] Since DGA already receives associative informa-

[6] The number of epochs until convergence lies within the typical range of values for graph convolutional networks.

[7] We experimented with using vectors trained on isolated pairs of stems and affix groups instead of window-1 vectors trained on the full text, but the performance was comparable. We also implemented the DGA using only window-1 vectors (without concatenating them with window-10 vectors), but it performed considerably worse.

	sports						entertainment						knowledge							
	cfb		nba		nfl		gam		lol		mov		pol		sci		tec		$\mu \pm \sigma$	
Model	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC
DGA+	.783	.754	.764	.749	.773	.751	.775	.759	.758	.740	.772	.749	.777	.766	.809	.795	.799	.778	.779±.015	.760±.016
DGA	.760	.730	.754	.731	.762	.740	.762	.743	.752	.738	.765	.747	.764	.750	.783	.770	.781	.761	.765±.010	.746±.012
CM	.745	.745	.751	.759	.764	.766	.768	.776	.766	.773	.769	.780	.776	.786	.793	.804	.768	.775	.767±.013	.774±.016
NC+	.737	.739	.729	.733	.737	.740	.722	.728	.730	.733	.732	.741	.725	.731	.772	.781	.758	.756	.738±.016	.742±.016
NC	.704	.710	.705	.715	.719	.728	.709	.714	.699	.711	.709	.720	.695	.708	.731	.743	.734	.737	.712±.013	.721±.012
JS	.632	.593	.617	.582	.626	.588	.619	.588	.609	.584	.622	.589	.614	.591	.649	.617	.638	.608	.625±.012	.593±.011
BM	.598	.602	.592	.597	.600	.600	.592	.592	.583	.585	.596	.594	.583	.584	.610	.601	.589	.596	.594±.008	.595±.006

Table 3: Performance on MWF prediction. The table shows AP and AUC of the models for the nine SRs as well as averaged scores. Grey highlighting illustrates the best score in a column, light grey the second-best.

tion from the DG and semantic information from the embeddings trained with window size 10, the main advantage of DGA+ should lie in additional syntactic information.

CM: We use binary cross entropy as loss function. We train the CM for 20 epochs using Adam with a learning rate of 0.001. Both input character embeddings and hidden states of the bidirectional LSTMs have 100 dimensions. The output of the first feed-forward layer has 50 dimensions. We apply dropout of 0.2 after the embedding layer as well as the first feed-forward layer.

NC, NC+: All hyperparameters are identical to the DGA and the DGA+, respectively.

JS: Similarity scores are computed on the SR training sets.

BM: Transition probabilities are maximum likelihood estimates from the SR training sets. If a stem is assigned several POS tags by the tagger, we take the most frequent one.

Table 2 summarizes the number of trainable parameters for the neural models. Notice that CM has more than 10 times as many trainable parameters as DGA+, DGA, NC+, and NC.

5.3 Results

The overall best performing models are DGA+ and CM (see Table 3). While DGA+ beats CM on all SRs except for lol in AP, CM beats DGA+ on all SRs except for cfb and tec in AUC. Except for CM, DGA+ beats all other models on all SRs in both AP and AUC, i.e., it is always the best or second-best model. DGA beats all models except for DGA+ and CM on all SRs in AP but has lower AUC than NC+ on three SRs. It also outperforms CM on three SRs in AP. NC+ and NC mostly have scores above 0.7, showing that traditional morpheme embeddings also capture information about the compatibility of affixes and stems (albeit to a lesser degree than models with associative or orthographic information). Among the non-neural methods, JS outperforms BM (and the other non-neural link prediction models, see Appendix A.4) in AP, but is beaten by BM in AUC on six SRs.

The fact that DGA+ performs on par with CM while using less than 10% of CM's parameters demonstrates the power of incorporating associative information from the mental lexicon in modeling the MWF of derivatives. This result is even more striking since DGA+, as opposed to CM, has no direct access to orthographic (i.e., phonological) information. At the same time, CM's high performance indicates that orthographic information is an important predictor of MWF.

6 Derivational Embeddings

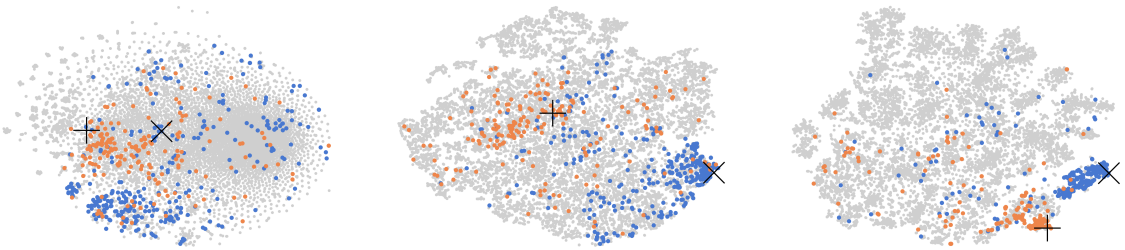
6.1 Comparison with Input Vectors

To understand better how associative information from the DG increases performance, we examine how DGA+ changes the shape of the vector space by comparing input vs. learned embeddings ($\mathbf{X}$ vs. $\mathbf{Z}_{DGA+}$), and contrast that with NC+ ($\mathbf{X}$ vs. $\mathbf{Z}_{NC+}$). A priori, there are two opposing demands the embeddings need to respond to: (i) as holds for bipartite graphs in general (Gao et al., 2018), the two node sets (stems and affix groups) should form two separated clusters in embedding space; (ii) stems associated with the same affix group should form clusters in embedding space that are close to the embedding of the respective affix group.

For this analysis, we define $\delta(\mathcal{N}, \mathbf{v})$ as the mean cosine similarity between the embeddings of a node set $\mathcal{N}$ and an individual embedding $\mathbf{v}$,

$$\delta(\mathcal{N}, \mathbf{v}) = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \cos(\mathbf{u}_n, \mathbf{v}), \qquad (11)$$

where $\mathbf{u}_n$ is the embedding of node n. We calculate δ for the set of stem nodes $\mathcal{S}$ and their centroid $\mathbf{c}_\mathcal{S} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathbf{u}_s$ as well as the set of affix group



(a) $\mathbf{X}$ (b) $\mathbf{Z}_{NC+}$ (c) $\mathbf{Z}_{DGA+}$

Figure 5: Comparison of input embeddings $\mathbf{X}$ with learned representations $\mathbf{Z}_{NC+}$ and $\mathbf{Z}_{DGA+}$. The plots are t-SNE projections (van der Maaten and Hinton, 2008) of the embedding spaces. We highlight two example sets of stems occurring with a common affix: the blue points are stems occurring with `$esque`, the orange points stems occurring with `$ful`. $\times$ marks the embedding of `$esque`, $+$ the embedding of `$ful`.

Measure	$\mathbf{X}$	$\mathbf{Z}_{NC+}$	$\mathbf{Z}_{DGA+}$
$\delta(\mathcal{S}, \mathbf{c}_{\mathcal{S}})$	.256 ± .026	.500 ± .027	.487 ± .022
$\delta(\mathcal{A}, \mathbf{c}_{\mathcal{A}})$	.377 ± .017	.522 ± .016	.322 ± .030
$\delta(\mathcal{S}_a, \mathbf{c}_{\mathcal{S}_a})$	.281 ± .006	.615 ± .017	.671 ± .024
$\delta(\mathcal{S}_a, \mathbf{u}_a)$	.133 ± .006	.261 ± .022	.278 ± .033

Table 4: Comparison of $\mathbf{X}$ with $\mathbf{Z}_{NC+}$ and $\mathbf{Z}_{DGA+}$. The table shows topological measures highlighting differences between the input and learned embeddings.

nodes $\mathcal{A}$ and their centroid $\mathbf{c}_{\mathcal{A}} = \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \mathbf{u}_a$. Table 4 shows that while NC+ makes the embeddings of both $\mathcal{S}$ and $\mathcal{A}$ more compact (higher similarity in $\mathbf{Z}_{NC+}$ than in $\mathbf{X}$), DGA+ makes $\mathcal{S}$ more compact, too, but decreases the compactness of $\mathcal{A}$ (lower similarity in $\mathbf{Z}_{DGA+}$ than in $\mathbf{X}$). $\mathbf{Z}_{NC+}$ meets (i) to a greater extent than $\mathbf{Z}_{DGA+}$.

We then calculate δ for all sets of stems $\mathcal{S}_a$ occurring with a common affix group a and their centroids $\mathbf{c}_{\mathcal{S}_a} = \frac{1}{|\mathcal{S}_a|}\sum_{s\in\mathcal{S}_a} \mathbf{u}_s$. We also compute δ for all $\mathcal{S}_a$ and the embeddings of the corresponding affix groups $\mathbf{u}_a$. As Table 4 shows, both values are much higher in $\mathbf{Z}_{DGA+}$ than in $\mathbf{X}$, i.e., DGA+ brings stems with a common affix group a (lexical clusters in the mental lexicon) close to each other while at the same time moving a into the direction of the stems. The embeddings $\mathbf{Z}_{NC+}$ exhibit a similar pattern, but more weakly than $\mathbf{Z}_{DGA+}$ (see Table 4 and Figure 5). $\mathbf{Z}_{DGA+}$ meets (ii) to a greater extent than $\mathbf{Z}_{NC+}$.

Thus, DGA+ and NC+ solve the tension between (i) and (ii) differently; the associative information from the mental lexicon allows DGA+ to put a greater emphasis on (ii), leading to higher performance in MWF prediction.

6.2 Comparison between SRs

Another reason for the higher performance of the models with associative information could be that their embeddings capture differences in derivational patterns between the SR communities. To examine this hypothesis, we map the embeddings $\mathbf{Z}_{DGA+}$ of all SRs into a common vector space by means of orthogonal procrustes alignment (Schönemann, 1966), i.e., we optimize

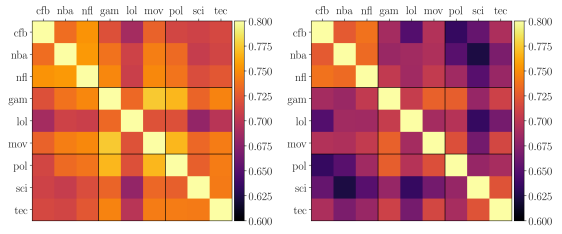
$$\mathbf{R}^{(i)} = \underset{\mathbf{T}^\top\mathbf{T}=\mathbf{I}}{\arg\min} ||\mathbf{Z}^{(i)}_{DGA+}\mathbf{T} - \mathbf{Z}^{(0)}_{DGA+}||_F \quad (12)$$

for every SR, where $\mathbf{Z}^{(i)}_{DGA+}$ is the embedding matrix of the SR i, and $\mathbf{Z}^{(0)}_{DGA+}$ is the embedding matrix of a randomly chosen SR (which is the same for all projections). We then compute the intersection of stem and affix group nodes from all SRs $\mathcal{S}_\cap = \bigcap_i \mathcal{S}^{(i)}$ and $\mathcal{A}_\cap = \bigcap_i \mathcal{A}^{(i)}$, where $\mathcal{S}^{(i)}$ and $\mathcal{A}^{(i)}$ are the stem and affix group sets of SR i, respectively. To probe whether differences between SRs are larger or smaller for affix embeddings as compared to stem embeddings, we define

$$\Delta(\mathcal{S}^{(i)}, \mathcal{S}^{(j)}) = \sum_{s\in\mathcal{S}_\cap} \frac{\cos(\hat{\mathbf{z}}^{(i)}_s, \hat{\mathbf{z}}^{(j)}_s)}{|\mathcal{S}_\cap|}, \quad (13)$$

i.e., the mean cosine similarity between projected embedding pairs $\hat{\mathbf{z}}^{(i)}_s$ and $\hat{\mathbf{z}}^{(j)}_s$ from two SRs i and j representing the same stem s in the intersection set $\mathcal{S}_\cap$, with $\hat{\mathbf{z}}^{(i)}_s = \mathbf{z}^{(i)}_s\mathbf{R}^{(i)}$. $\Delta(\mathcal{A}^{(i)}, \mathcal{A}^{(j)})$ is defined analogously for affix groups.

The mean value for $\Delta(\mathcal{A}^{(i)}, \mathcal{A}^{(j)})$ (0.723 ± 0.102) is lower than that for $\Delta(\mathcal{S}^{(i)}, \mathcal{S}^{(j)})$ (0.760 ± 0.087), i.e., differences between affix group embeddings are more pronounced than between stem embeddings. Topically connected SRs are more similar to each other than SRs of different topic groups,



(a) $\Delta(\mathcal{S}^{(i)}, \mathcal{S}^{(j)})$ (b) $\Delta(\mathcal{A}^{(i)}, \mathcal{A}^{(j)})$

Figure 6: Comparison of embedding spaces across SRs. The plots show color-coded values of $\Delta(\mathcal{S}^{(i)}, \mathcal{S}^{(j)})$ and $\Delta(\mathcal{A}^{(i)}, \mathcal{A}^{(j)})$ for all pairs of SRs, respectively. The block-diagonal structure highlights the impact of topical relatedness on embedding similarities.

with the differences being larger in $\Delta(\mathcal{A}^{(i)}, \mathcal{A}^{(j)})$ than in $\Delta(\mathcal{S}^{(i)}, \mathcal{S}^{(j)})$ (see Figure 6).

These results can be related to Section 6.1: affix groups are very close to the stems they associate with in $\mathbf{Z}_{DGA+}$, i.e., if an affix group is used with stems of meaning p in one SR and stems with meaning q in the other SR, then the affix groups also have embeddings close to p and q in the two SRs. Most technical vocabulary, on the other hand, is specific to a SR and does not make it into $\mathcal{S}_{\cap}$.[8]

A qualitative analysis supports this hypothesis: affix groups with low cosine similarities between SRs associate with highly topical stems; e.g., the affix group `$ocracy` has a low cosine similarity of -0.189 between the SRs nba and pol, and it occurs with stems such as `kobe`, `jock` in nba but `left`, `wealth` in pol.

7 Related Work

Much recent computational research on **derivational morphology in NLP** has focused on two related problems: predicting the meaning of a derivative given its form, and predicting the form of a derivative given its meaning.

The first group of studies models the meaning of derivatives as a function of their morphological structure by training embeddings directly on text segmented into morphemes (Luong et al., 2013; Qiu et al., 2014) or by inferring morpheme embeddings from whole-word vector spaces, e.g., using the vector offset method (Lazaridou et al., 2013; Padó et al., 2016). Formally, given a derived form f_d, this line of research tries to find the meaning m_d that maximizes $P(m_d|f_d)$.

The second group of studies models the form of derivatives as a function of their meaning. The meaning is represented by the base word and a semantic tag (Cotterell et al., 2017; Deutsch et al., 2018) or the sentential context (Vylomova et al., 2017). Formally, given a meaning m_d, these studies try to find the derived form f_d of a word that maximizes $P(f_d|m_d)$.

Our study differs from these two approaches in that we model $P(W|f_d, m_d)$, i.e., we predict the overall likelihood of a derivative to exist. For future research, it would be interesting to apply derivational embeddings in studies of the second type by using them as pretrained input.

Neural link prediction is the task of inferring the existence of unknown connections between nodes in a graph. Advances in deep learning have prompted various neural models for link prediction that learn distributed node representations (Tang et al., 2015; Grover and Leskovec, 2016). Kipf and Welling (2016, 2017) proposed a convolutional graph auto-encoder that allows to include feature vectors for each node. The model was adapted to bipartite graphs by van den Berg et al. (2018).

Previous studies on neural link prediction for bipartite graphs have shown that the embeddings of the two node sets should ideally form separated clusters (Gao et al., 2018). Our work demonstrates that relations transcending the two-mode graph structure can lead to a trade-off between clustering and dispersion in embedding space.

8 Conclusion

We have introduced a derivational graph auto-encoder (DGA) that combines syntactic and semantic information with associative information from the mental lexicon to predict morphological well-formedness (MWF), a task that has not been addressed before. The model achieves good results and performs on par with a character-based LSTM at a fraction of the number of trainable parameters (less than 10%). Furthermore, the model learns embeddings capturing information about the compatibility of affixes and stems in derivation.

Acknowledgements. Valentin Hofmann was funded by the Arts and Humanities Research Council and the German Academic Scholarship Foundation. This research was also supported by the European Research Council (Grant No. 740516). We thank the reviewers for their helpful and very constructive comments.

[8] One SR standing out in Figure 6 is lol, a multiplayer online video game, in which many common stems such as `fame` and `range` have highly idiosyncratic meanings.

References

Paolo Acquaviva. 2016. Morphological semantics. In Andrew Hippisley and Gregory Stump, editors, *The Cambridge handbook of morphology*, pages 117–148. Cambridge University Press, Cambridge.

Lada A. Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social Networks*, 25:211–230.

Maria Alegre and Peter Gordon. 1999. Rule-based versus associative processes in derivational morphology. *Brain and Language*, 68(1-2):347–354.

Margaret R. Allen. 1979. *Morphological investigations*. University of Connecticut, Mansfield, CT.

Frank Anshen and Mark Aronoff. 1999. Using dictionaries to study the mental lexicon. *Brain and Language*, 68:16–26.

Mark Aronoff. 1976. *Word formation in generative grammar*. MIT Press, Cambridge, MA.

Laurie Bauer. 2001. *Morphological productivity*. Cambridge University Press, Cambridge, UK.

Laurie Bauer. 2019. *Rethinking morphology*. Edinburgh University Press, Edinburgh, UK.

Joan Bybee. 1995. Regular morphology and the lexicon. *Language and Cognitive Processes*, 10(425-455).

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *International Conference on Knowledge Discovery and Data Mining (KDD) 23*.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017. Paradigm completion for derivational morphology. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017*.

David Crystal. 1997. *The Cambridge encyclopedia of the English language*. Cambridge University Press, Cambridge, UK.

Hanjun Dai, Bo Dai, and Le Song. 2016. Discriminative embeddings of latent variable models for structured data. In *International Conference on Machine Learning (ICML) 33*.

Marco del Tredici and Raquel Fernández. 2018. The road to success: Assessing the fate of linguistic innovations in online communities. In *International Conference on Computational Linguistics (COLING) 27*.

Daniel Deutsch, John Hewitt, and Dan Roth. 2018. A distributional and orthographic aggregation model for english derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*.

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

Ming Gao, Leihui Chen, Xiangnan He, and Aoying Zhou. 2018. BiNE: Bipartite network embedding. In *International Conference on Research and Development in Information Retrieval (SIGIR) 41*.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In *International Conference on Machine Learning (ICML) 34*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *International Conference on Knowledge Discovery and Data Mining (KDD) 22*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Annual Meeting of the Association for Computational Linguistics (ACL) 49*.

Martin Haspelmath and Andrea D. Sims. 2010. *Understanding morphology*. Routledge, New York, NY.

Jennifer Hay. 2001. Lexical frequency in morphology: Is everything relative? *Linguistics*, 39(6):1041–1070.

Jennifer Hay and Ingo Plag. 2004. What constrains possible suffix combinations? on the interaction of grammatical and processing restrictions in derivational morphology. *Natural Language & Linguistic Theory*, 22(3):565–596.

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2020. Predicting the growth of morphological families from social and linguistic factors. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Diederik P. Kingma and Jimmy L. Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR) 3*.

Thomas N. Kipf and Max Welling. 2016. Variational graph auto-encoders. In *NIPS Bayesian Deep Learning Workshop*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR) 5*.

Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Annual Meeting of the Association for Computational Linguistics (ACL) 51*.

David Liben-Nowell and Jon Kleinberg. 2003. The link prediction problem for social networks. In *ACM Conference on Information and Knowledge Management (CIKM) 12*.

Pierre Lison and Andrey Kutuzov. 2017. Redefining context windows for word embedding models: An experimental study. In *arXiv 1704.05781*.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Conference on Computational Natural Language Learning (CoNLL) 17*.

Einat Minkov, Kristina Toutanova, and Hisami Suzuki. 2007. Generating complex morphology for machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL) 45*.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning (ICML) 27*.

Sebastian Padó, Aurélie Herbelot, Max Kisselew, and Jan Šnajder. 2016. Predictability of distributional semantics in derivational word formation. In *International Conference on Computational Linguistics (COLING) 26*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2014*.

Janet Pierrehumbert. 2012. The dynamic lexicon. In Abigail Cohn, Cécile Fougeron, and Marie Huffman, editors, *The Oxford handbook of laboratory phonology*, pages 173–183. Oxford University Press, Oxford.

Janet Pierrehumbert and Ramon Granell. 2018. On hapax legomena and morphological productivity. In *Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON) 15*.

Ingo Plag. 2003. *Word-formation in English*. Cambridge University Press, Cambridge, UK.

Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Co-learning of word representations and morpheme representations. In *International Conference on Computational Linguistics (COLING) 25*.

Peter H. Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 36(1).

Ingrid Sonnenstuhl and Axel Huth. 2002. Processing and representation of german -n plurals: A dual mechanism approach. *Brain and Language*, 81(1-3):276–290.

S. Sreelekha and Pushpak Bhattacharyya. 2018. Morphology injection for English-Malayalam statistical machine translation. In *International Conference on Language Resources and Evaluation (LREC) 11*.

Gregory Stump. 2017. Rule conflation in an inferential-realizational theory of morphotactics. *Acta Linguistica Academica*, 64(1):79–124.

Gregory Stump. 2019. Some sources of apparent gaps in derivational paradigms. *Morphology*, 29(2):271–292.

Wanhua Su, Yan Yuan, and Mu Zhu. 2015. A relationship between the average precision and the area under the ROC curve. In *International Conference on the Theory of Information Retrieval (ICTIR) 2015*.

Monica Tamariz. 2008. Exploring systematicity between phonological and context-cooccurrence representations of the mental lexicon. *The Mental Lexicon*, 3(2):259–278.

Chenhao Tan and Lillian Lee. 2015. All who wander: On the prevalence and characteristics of multi-community engagement. In *International Conference on World Wide Web (WWW) 24*.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *International Conference on World Wide Web (WWW) 24*.

Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2018. Graph convolutional matrix completion. In *KDD 2018 Deep Learning Day*.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579 2605.

Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin, and Trevor Cohn. 2017. Context-aware prediction of derivational word-forms. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL) 15*.

A Appendices

A.1 Data Preprocessing

We filter the Reddit posts for known bots and spammers (Tan and Lee, 2015). We remove abbreviations, strings containing numbers, references to users and SRs, and both full and shortened hyperlinks. We convert British English spelling variants to American English and lemmatize all words. We follow Han and Baldwin (2011) in reducing repetitions of more than three letters (`niiiiice`) to three letters. Except for excluding stopwords, we do not employ a frequency threshold.

	sports						entertainment						knowledge						$\mu \pm \sigma$	
	cfb		nba		nfl		gam		lol		mov		pol		sci		tec			
Model	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC
JS	.632	.593	.617	.582	.626	.588	.619	.588	.609	.584	.622	.589	.614	.591	.649	.617	.638	.608	.625±.012	.593±.011
AA	.603	.556	.599	.556	.602	.553	.605	.561	.589	.553	.596	.552	.592	.556	.606	.558	.606	.562	.600±.006	.556±.003
CN	.600	.553	.596	.553	.598	.550	.602	.558	.585	.550	.592	.548	.588	.552	.601	.554	.603	.558	.596±.006	.553±.003
PA	.537	.517	.543	.527	.542	.522	.559	.545	.545	.534	.533	.519	.541	.534	.513	.503	.537	.526	.539±.011	.525±.011

Table 5: Performance on MWF prediction. The table shows AP and AUC of the models for the nine Subreddits as well as averaged scores. Grey highlighting illustrates the best score in a column, light grey the second-best.

A.2 Morphological Segmentation

We start by defining a set of potential stems $O^{(i)}$ for each Subreddit i. A word w is given the status of a potential stem and added to $O^{(i)}$ if it consists of at least 4 characters and has a frequency count of at least 100 in the Subreddit.

Then, to determine the stem of a specific word w, we employ an iterative algorithm. Let $V^{(i)}$ be the vocabulary of the Subreddit, i.e., all words occurring in it. Define the set B_1 of w as the bases in $V^{(i)}$ that remain when one affix is removed, and that have a higher frequency count than w in the Subreddit. For example, `reaction` can be segmented as `re$action` and `react$ion`, so $B_1(\texttt{reaction}) = \{\texttt{action}, \texttt{react}\}$ (assuming `action` and `react` both occur in the Subreddit and are more frequent than `reaction`). We then iteratively create $B_{i+1}(w) = \bigcup_{b \in B_i(w)} B_1(b)$. Let further $B_0(w) = \{w\}$. We define $S(w) = O^{(i)} \cap B_m(w)$ with $m = \max\{k | O^{(i)} \cap B_k(w) \neq \emptyset\}$ as the set of stems of w. If $|S(w)| > 1$ (which is rarely the case in practice), the element with the lowest number of suffixes is chosen.

The algorithm is sensitive to most morpho-orthographic rules of English (Plag, 2003): when `$ness` is removed from `happi$ness`, e.g., the result is `happy`, not `happi`.

A.3 Message Passing Rule

Let $\hat{\mathbf{B}} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ be the adjacency matrix of the DG $\mathcal{B}$ with added self-loops, i.e., $\hat{B}_{ii} = 1$ and $\hat{\mathbf{D}} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ the degree matrix of $\hat{\mathbf{B}}$ with $\hat{D}_{ii} = \sum_j \hat{B}_{ij}$. The matrix form of the message passing step can be expressed as

$$\mathbf{M}^{(l)} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{B}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X}^{(l-1)} \mathbf{W}^{(l)}, \tag{14}$$

where $\mathbf{W}^{(l)}$ is the trainable weight matrix of layer l, and $\mathbf{X}^{(l-1)} \in \mathbb{R}^{|\mathcal{V}| \times |n|}$ is the matrix containing the node feature vectors from layer $l-1$ (Kipf and Welling, 2016, 2017). The activation step then is

$$\mathbf{X}^{(l)} = \mathrm{ReLU}\left(\mathbf{M}^{(l)}\right). \tag{15}$$

A.4 Feature-based Link Prediction

Besides Jaccard similarity, we implement three other feature-based link prediction methods.

Adamic-Adar. The Adamic-Adar (AA) index (Adamic and Adar, 2003) has to take the bipartite structure of DGs into account. Using the modified definition of common neighbors as with ζ_{JS}, we calculate it as

$$\zeta_{AA}(s, a) = \sum_{\substack{n \in \Gamma_\cap(s,a) \\ n \in \Gamma_\cap(a,s)}} \frac{1}{d(n)}. \tag{16}$$

Common Neighbors. The score of an edge (s, a) is calculated as the cardinality of the set of common neighbors (CN) of s and a. Similarly to ζ_{JS} and ζ_{AA}, we calculate the CN score as

$$\zeta_{CN}(s, a) = |\Gamma_\cap(s, a)| + |\Gamma_\cap(a, s)|. \tag{17}$$

Preferential Attachment. For preferential attachment (PA), the score of an edge (s, a) is the product of the two node degrees,

$$\zeta_{PA}(s, a) = d(s) \cdot d(a). \tag{18}$$

The training regime is identical to Jaccard similarity. AA outperforms PA and CN but is consistently beaten by JS (see Table 5).

4

Predicting the Growth of Morphological Families

Contents

Building on the study reported in chapter 3, the second study examines a problem closely related to predicting whether a derivative exists, but with a focus on forecasting lexical dynamics into the future: rather than taking the test derivatives from the same time point as the train derivatives, it evaluates models on the derivational system at a later point in time. To make this feasible, the study specifically examines the task of predicting whether the morphological family of a base will grow or not in the future, i.e., whether a base will become the basis for the creation of new derivatives, and not how exactly these derivatives will look like.

Predicting the Growth of Morphological Families from Social and Linguistic Factors

Valentin Hofmann[*‡], **Janet B. Pierrehumbert**[†*], **Hinrich Schütze**[‡]

[*]Faculty of Linguistics, University of Oxford
[†]Department of Engineering Science, University of Oxford
[‡]Center for Information and Language Processing, LMU Munich
valentin.hofmann@ling-phil.ox.ac.uk

Abstract

We present the first study that examines the evolution of morphological families, i.e., sets of morphologically related words such as "trump", "antitrumpism", and "detrumpify", in social media. We introduce the novel task of Morphological Family Expansion Prediction (MFEP) as predicting the increase in the size of a morphological family. We create a ten-year Reddit corpus as a benchmark for MFEP and evaluate a number of baselines on this benchmark. Our experiments demonstrate very good performance on MFEP.

1 Introduction

Lexical change is a prime indicator of topical dynamics in social media. When people or events attract the attention of a user community, this is reflected by the *token frequency evolution of individual words*. The burst in token frequency of the word "trump" in social media before the 2016 presidential election (see Figure 1), e.g., mirrors the increasing presence of Donald Trump in public discourse during that time.

However, token frequency is only one way of measuring changes in topical prominence. Accompanying the increase in token frequency of "trump", there was a parallel increase in the number of words morphologically related to "trump", i.e., words like "trumpification", "antitrumpism", and "detrumpify" (see Figure 1, Table 1). Most of these words have a very low token frequency and are removed in the first steps of a typical NLP pipeline.

Here, we present the first study of lexical change in social media that takes as its main unit of analysis the *type frequency evolution of morphological families*, i.e., changes in the number of morphologically related words such as "trump", "trumpification", "antitrumpism", and "detrumpify". We show that morphological families allow for a fresh view

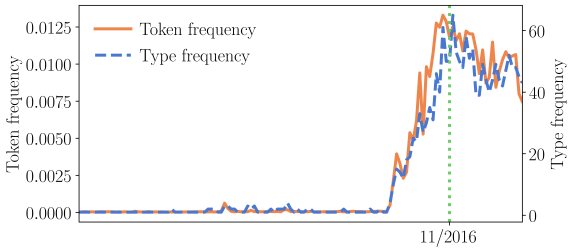


Figure 1: Token frequency of "trump" and type frequency of derivations of "trump" in the r/politics Subreddit between 08/2007 and 07/2018. Vertical line: date of the 2016 presidential election.

Month	Words
04/2015	"trump"
05/2015	"trump", "trumpish", "trumpster", "trumpy"
06/2015	"trump", "trumpening", "trumper", "trumpish", "trumpness", "trumpology", "trumpster", "trumpy"
07/2015	"trump", "trumpening", "trumper", "trumpic", "trumpification", "trumpiness", "trumpish", "trumpism", "trumpistan", "trumpness", "trumpster", "trumpy"

Table 1: Derivations of "trump" in four subsequent months of the r/politics Subreddit.

of lexical change in social media, making them a promising tool for studies in the social sciences that draw on NLP techniques. At the same time, our work adds to the growing body of computational research on derivational morphology (Cotterell et al., 2017; Vylomova et al., 2017; Cotterell and Schütze, 2018; Deutsch et al., 2018; Pierrehumbert and Granell, 2018; Hofmann et al., 2020) by introducing a temporal perspective.

Contributions. We introduce the novel task of Morphological Family Expansion Prediction (MFEP), which aims at predicting whether a morphological family will increase in size or not. We publish a benchmark for MFEP and show that the growth of morphological families can be successfully modeled using social and linguistic factors relating to the morphological parent. Furthermore,

our results add a new perspective to the growing body of research on the link between cultural and linguistic change in social media.[1]

2 Morphological Families

We define a morphological *family* as a set F of words w with a shared free morpheme. Thus, "trump", "trumpification", "antitrumpism", and "detrumpify" are in the same morphological family because they share the free morpheme "trump". By contrast, "antitrumpism" and "antiprogressivism" are in different morphological families: even though both words have two morphemes in common ("anti" and "ism"), they do not belong to the same morphological family according to our definition since "anti" and "ism" are bound morphemes, not free morphemes like in the case of "trump", "trumpification", "antitrumpism", and "detrumpify". In this study, we only consider derivational morphology.[2] Compounds such as "trumpwall" are not split into their parts, but they can become a parent (see below). Thus, each word belongs to exactly one morphological family. The cardinality $|F|$ of a family will be referred to as the morphological family *size*, a term also used in other studies (Schreuder and Baayen, 1997; de Jong et al., 2000; del Prado Martín et al., 2004).

The morphological *parent* w^* is the morphologically most basic word of a family F. The word "trump" is the parent of "antitrumpism", "trumpification", and "detrumpify". We denote the morphological family of w^* as $F(w^*)$.

Except for the parent, all members of a family are morphological *children* and form a subset $\tilde{F}$ of the entire family F. The words "antitrumpism", "trumpification", and "detrumpify" are all morphological children. We further distinguish between *old* children $\tilde{F}_o$, established words in the lexicon, and *new* children $\tilde{F}_n$, innovative forms. While "trumpify" can still be considered a new child of "trump", "trumpster" is on its way to becoming an old child in the family.[3]

As the example of "trumpster" shows, morphological families are in constant flux. Specifically, there are three types of change in morphological families: word *birth* $\emptyset \rightsquigarrow \tilde{F}_n$, word *entrenchment* $\tilde{F}_n \rightsquigarrow \tilde{F}_o$, and word *death* $\tilde{F}_n, \tilde{F}_o \rightsquigarrow \emptyset$. The topic of this paper is word birth, i.e., we ask: given a set $\mathcal{F}$ of morphological families, which will increase in size during a specified time interval? This differentiates our work from previous research on lexical change in social media which has focused on word entrenchment (see Section 7). One question we are particularly interested in is whether endogenous (language-internal) or exogenous (language-external) factors are better predictors of morphological family growth; these factors have been previously compared for changes in word token frequency (Altmann et al., 2011), but not for changes in morphological type frequency.

3 Experimental Data

We develop MFEP using Reddit, a social media platform hosting discussions about a variety of topics. Reddit is divided into smaller communities centered around a shared interest, so-called *subreddits* (SRs), which are highly conducive to linguistic innovation (del Tredici and Fernández, 2018). Concretely, we draw upon the Baumgartner Reddit Corpus, a collection of (almost) all publicly available comments posted on Reddit since 2005.[4] A three-year slice of this corpus was used in a study on lexical change by Stewart and Eisenstein (2018). Gaffney and Matias (2018) show that the corpus's coverage of Reddit is not complete, but we do not expect this to affect our analysis.

Our study examines data from 2007 to 2018 in the four SRs `r/gaming`, `r/movies`, `r/nba`, and `r/politics`. These SRs were chosen because they are of comparable size, belong to the largest SRs of Reddit, and at the same time all reflect distinct areas of interest (Table 2). For each month, we also draw a random sample of comments from all SRs that will be used for computing word topicality (Section 4). The size of the sample equals the average size of the four selected SRs within the respective month.

Preprocessing. As in previous work (Tan and Lee, 2015), we filter posts for known bots and spammers. We remove abbreviations ("viz."), strings containing numbers ("b4"), references to users ("u/user") and SRs ("r/subreddit"), and both

[1] We make all our data and code publicly available at https://github.com/valentinhofmann/mfep.

[2] The distinction between inflection and derivation is gradual, not binary (Haspelmath and Sims, 2010). The suffix "ly", e.g., is variously defined as inflectional or derivational (Bauer, 2019). We try to exclude inflectional morphology as far as possible (e.g., by lemmatizing), but we are aware that a clear separation does not exist in linguistic reality.

[3] "Trumpster" is already listed in the English Wiktionary at https://en.wiktionary.org/wiki/Trumpster.

[4] The corpus is available at https://files.pushshift.io/reddit/comments/.

SR	Start	End	N_I	N_W	N_U	N_T	$\lvert\mathcal{F}\rvert$	$\mu_{\lvert F\rvert}$	$\sigma_{\lvert F\rvert}$
r/politics	08/2007	07/2018	19	2,970,509,554	1,519,039	2,020,843	19,000	4.64	4.05
r/gaming	09/2007	08/2018	19	1,119,096,999	2,875,931	1,859,228	19,000	4.00	3.48
r/movies	06/2008	05/2018	17	738,365,964	1,772,085	613,158	17,000	3.44	3.09
r/nba	10/2010	09/2018	13	898,483,442	486,746	721,641	13,000	3.20	2.93

Table 2: SR summary statistics. Start: first month included; end: last month included; N_I: number of intervals; N_W: number of word tokens; N_U: number of users; N_T: number of threads; $|\mathcal{F}|$: number of family examples ($1,000 \times N_I$); $\mu_{|F|}$: mean family size per month; $\sigma_{|F|}$: standard deviation of family size per month.

full and shortened hyperlinks. We convert British English spelling variants to American English and lemmatize all words to remove inflectional morphology. We follow Han and Baldwin (2011) in reducing repetitions of more than three letters ("niiiiice") to three letters. Except for stopwords, we do not employ a frequency threshold; in particular, we include words that occur only once.

Computing morphological families. Given a collection of texts S, we define the morphological families as follows. Let V_S be the vocabulary of S, i.e., all words occurring in it. We define the set of parents $O_S \subset V_S$ as the 1,000 most frequent words in S, regardless of whether the word is decomposable or not. This means that parents are not necessarily morphological roots (Haspelmath and Sims, 2010).[5] We attempt to segment all other words w using affixes from a representative list of productive prefixes and suffixes in English (Crystal, 1997). We define the set C of candidate parents of w as follows. If $w \in O_S$, then $C(w) = \{w\}$. Otherwise, $C(w) = \bigcup_{b \in B(w)} C(b)$, where $B(w)$ is the set of bases that remain when one of w's affixes is removed. For $w^* \in O_S$, we then define its morphological family as

$$F(w^*) = \{w \in V_S | w^* \in C(w) \wedge |C(w)| = 1\}.$$

Procedurally, families can be identified by a recursive bottom-up algorithm. The algorithm is sensitive to morpho-orthographic rules of English (Plag, 2003); e.g., when "ness" is removed from "trumpiness", the result is "trumpy", not "trumpi".

[5] In a situation where both "sense" and "sensation", e.g., fall above the frequency threshold, we get two separate morphological families of the parent "sense" (a root) with "nonsense", "sensitive", etc. and the parent "sensation" (not a root) with "sensational", "sensationalism", etc. (the children of "sensation" are not added to the family of "sense"). However, most morphological parents are in fact roots.

4 Predictors for MFEP

In a setup similar to Altmann et al. (2011), we formalize MFEP as a binary classification task. Given a *context interval* $i^{(c)}$, a following temporally adjacent *probe interval* $i^{(p)}$, and a morphological parent w^*, we ask: what properties of w^* can predict whether $|F(w^*)|$ increases in $i^{(p)}$ or not?

Here, we set the length of $i^{(c)}$ to 18 months and the length of $i^{(p)}$ to 6 months. Morphological families are computed separately for each pair of $i^{(c)}$ and $i^{(p)}$. The lowest frequency count of a parent in $i^{(c)}$ is 244. Table 2 summarizes statistics of the morphological families for each SR.

We define a number of predictors for family expansion that are measurements of properties of w^*. All predictors are motivated by work in psycholinguistics and NLP. They fall into three natural classes: (i) a *type frequency-based* predictor ($\overline{|F|}$), (ii) *token frequency-based* predictors (f_r, $\bar{z}^{(p)}$), and (iii) *dissemination-based* predictors (D_w^U, D_w^T, Q_w). All predictors except for $\bar{z}^{(p)}$(which is measured in $i^{(c)}$ and $i^{(p)}$) are measured in $i^{(c)}$.

Family size $\overline{|F|}$. The family size is a prime example of an endogenous (language-internal) factor, i.e., one that depends on the linguistic system. A morpheme with a large family might combine more readily with new affixes than a morpheme that occurs only with a small number of affixes. This idea bears a theoretical connection to smoothing techniques such as Witten-Bell and Kneser-Ney smoothing, which model the probability of previously unseen n-grams containing a given word ($\approx |\tilde{F}_n|$) by assuming a rich-get-richer process (Manning and Schütze, 1999; Teh, 2006). It is also in line with lexical growth models based on preferential attachment (Steyvers and Tenenbaum, 2005). Intuitively, the fact that morphological children themselves can become the basis for new derivations also suggests a rich-get-richer process.

Notice that $\overline{|F|}$ is equivalent to the type fre-

quency of w^*. In linguistics, type frequency is known to be a good predictor of the productivity of inflectional patterns (Bybee, 1995). Furthermore, it has been shown that the morphological family size facilitates lexical processing (Schreuder and Baayen, 1997). To probe whether type frequency also influences the likelihood of a family to grow, we include the predictor $\overline{|F|}$, morphological family size averaged over the 18 months of $i^{(c)}$.

Relative token frequency of parent f_r. Frequent words are known to be more accessible in lexical processing than rare words (Jescheniak and Levelt, 1994). Therefore, they might be more available for use in novel derivations, causing an increase in morphological family size.

Trending behavior $\bar{z}^{(p)}$. Changes in the relative frequency of a morphological parent might be indicative of concomitant changes in the morphological family size. If a word gains in popularity and becomes more frequent, this could increase the chances of new morphologically related words being created. The trending behavior is a prime example of an exogenous (language-external) factor, i.e., one that depends on non-linguistic events (e.g., a presidential election). Therefore, we measure whether the parent increases in frequency.

This is done in the following way: we calculate the z-score of the frequency distribution of the parent in the probe interval $i^{(p)}$ relative to the frequency distribution in the context interval $i^{(c)}$. The mean of these z-scores is then used as a continuous variable in the model,

$$\bar{z}^{(p)} = \frac{1}{|i^{(p)}|}\sum_{j=1}^{|i^{(p)}|} z_j^{(p)} = \frac{1}{|i^{(p)}|}\sum_{j=1}^{|i^{(p)}|} \frac{x_j^{(p)} - \mu^{(c)}}{\sigma^{(c)}},$$

where $|i^{(p)}| = 6$ is the length in months of the probe interval and $x_j^{(p)}$ is the relative frequency of the parent in month j ($1 \leq j \leq |i^{(p)}|$) of $i^{(p)}$; $\mu^{(c)}$ and $\sigma^{(c)}$ are mean and standard deviation of relative frequency of the parent in the 18 months of the context interval $i^{(c)}$. The measure detects increases in frequency relative to the intrinsic variation in usage frequency of a particular word. This is necessary since some words naturally exhibit stronger short-term fluctuations, which we do not want to count as frequency bursts. Similar methods for peak detection in time series are frequently used, e.g., in Baskozos et al. (2019). We use both $i^{(c)}$ and $i^{(p)}$ for calculating $\bar{z}^{(p)}$ because this captures the idea of exogenous forcing without any additional assumptions; notice that the metric is calculated on the parent only and does not include any information about what is being predicted in MFEP, namely changes in the morphological family size.

User dissemination D_w^U. Following findings by Church and Gale (1995) and Altmann et al. (2011), we define user dissemination D_w^U as the extent to which the number of users of a specific word w deviates from a Poisson process,

$$D_w^U = \frac{U_w}{\tilde{U}(f_w)},$$

where U_w is the number of users who posted at least one comment including w in $i^{(c)}$, f_w is the relative frequency of w in $i^{(c)}$, and $\tilde{U}(f_w)$ is the expected number of users under a Poisson model given the relative frequency f_w. $\tilde{U}(f_w)$ can be calculated as

$$\tilde{U}(f_w) = \sum_{j=1}^{N_U} \tilde{U}_j \approx \sum_{j=1}^{N_U} \left(1 - e^{-f_w m_j^U}\right),$$

where N_U is the number of users, $\tilde{U}_j$ is the probability that the posts of user j contain w at least once, and m_j^U is the total number of words posted by user j in $i^{(c)}$. The approximation is valid for $f_w \ll 1$ and $m_j^U / \sum_j^{N_U} m_j^U \ll 1$. Our data satisfy both requirements.

User dissemination and the following dissemination measures have a cognitive justification: it has been shown that items that are used in more diverse situations and contexts are stored in human memory in a way that makes them more retrievable (Anderson and Milson, 1989; Brysbaert et al., 2016). Thus, words with a higher dissemination are more accessible to speakers and could figure more prominently among bases for new formations. The dissemination measures fall into a gray area between exogenous and endogenous factors since they reflect the cognitive representation of language-external properties (Altmann et al., 2011).

Thread dissemination D_w^T. Similar to user dissemination, thread dissemination D_w^T is defined as the extent to which the number of threads containing a specific word w deviates from a Poisson process (Altmann et al., 2011),

$$D_w^T = \frac{T_w}{\tilde{T}(f_w)},$$

where T_w is the number of threads that include at least one instance of w, and $\tilde{T}(f_w)$ is the expected

number of threads under a Poisson model. $\tilde{T}(f_w)$ can again be calculated as

$$\tilde{T}(f_w) = \sum_{j=1}^{N_T} \tilde{T}_j \approx \sum_{j=1}^{N_T} \left(1 - e^{-f_w m_j^T}\right),$$

where N_T, $\tilde{T}_j$, and m_j^T are defined analogously to N_U, $\tilde{U}_j$, and m_j^U. The approximation is again valid since the data satisfy $m_j^T / \sum_j^{N_T} m_j^T \ll 1$.

Topicality Q_w. Because SRs are communities centered around interests, words that are characteristic of a SR's topic are more frequent in that SR than in the others. Topicality has been shown to have an impact on lexical dynamics at long time scales (Church, 2000; Montemurro and Zanette, 2016). It could also influence the productivity of morphological families: higher topical dissemination, i.e., lower topicality, could facilitate growth. To capture this effect, we introduce a metric of topical distinctiveness, Q_w, which we define as

$$Q_w = \frac{f_w}{\tilde{f}_w},$$

where f_w is the relative frequency of the word w in a SR in $i^{(c)}$, and $\tilde{f}_w$ is the expected relative frequency of w based on a random sample of posts from all SRs in $i^{(c)}$.

The polarity of Q_w is reversed to D_w^U and D_w^T, i.e., a word that is very clumped in SR space will have a *high* value of Q_w, but a word that is very clumped in user or thread space will have a *low* value of D_w^U or D_w^T, respectively.

5 Experimental Setup

Finding growing families. We use two different notions of growth for MFEP: *absolute growth* and *relative growth*. We define absolute growth as

$$\delta_a(F) = \mu_{|F|}^{(p)} - \mu_{|F|}^{(c)},$$

where $\mu_{|F|}^{(p)}$ and $\mu_{|F|}^{(c)}$ are the mean morphological family size in $i^{(p)}$ and $i^{(c)}$, respectively. Relative growth is defined similarly as

$$\delta_r(F) = \frac{\mu_{|F|}^{(p)}}{\mu_{|F|}^{(c)}}.$$

For both δ_a and δ_r, we define binary features based on thresholds l_a and l_r, i.e., we define a morphological family F to be a positive example if $\delta_a(F) > l_a$ for a pair of $i^{(p)}$ and $i^{(c)}$ (in the case of absolute growth). We thus train two models: one for predicting whether $\delta_a(F) > l_a$, one for predicting whether $\delta_r(F) > l_r$.

Model. We use Random Forests (RF) to perform the classification (Breiman, 2001). RF offers two main advantages in comparison with other models. Firstly, as opposed to other tree-based models, RFs decorrelate trees, which is important if the features are correlated (as is the case here). Secondly, the feature importance scores of a RF provide a transparent way to compare the predictive power of features. We do not use more complex albeit potentially better performing methods such as deep architectures since our primary goal is to compare various features and show that MFEP is a feasible computational task.

Since the data contain considerably more negative than positive examples, we randomly sample one negative example for every positive example for the final data. The interval pairs from all SRs were merged into one dataset, which was then split into 0.8 and 0.2 for train/dev and test sets. The train/dev set was split again into 0.8 and 0.2 for train and dev sets. Thus, all sets contain a balanced sample of interval pairs from all SRs.[6]

We use a total of 68,000 pairs of intervals $(i^{(c)}, i^{(p)})$ where $i^{(c)}$ is the context interval and $i^{(p)}$ the probe interval (see also Table 2). Recall that $i^{(c)}$ has length 18 months and $i^{(p)}$ 6 months. Temporally adjacent interval pairs are overlapping by $|i^{(p)}|$ months, i.e., every month in the original data is used exactly once in a probe interval and three times in a context interval.

We do not perform hyperparameter tuning and instead choose typical values for the hyperparameters of RF: 80 for the number of trees, and 20 for tree depth. For our initial MFEP models, we set the thresholds $l_a = 2.4$ and $l_r = 1.6$, two values in the mid-range of existing values for δ_a and δ_r. We will later analyze the sensitivity to these hyperparameters in greater detail.

6 Results

Overall performance. As shown in Table 3, the RF models exhibit a good performance with an overall prediction accuracy of 80.9% for $l_a = 2.4$ and 70.8% for $l_r = 1.6$ (random baselines: 50.0%).

[6] Since the chosen SRs cover diverse topics, the model should have a high transferability to other SRs, but we have not tested this.

	l_a (δ_a)								l_r (δ_r)							
	0.0	0.8	1.6	2.4	3.2	4.0	4.8	$\mu \pm \sigma$	1.0	1.2	1.4	1.6	1.8	2.0	2.2	$\mu \pm \sigma$
All	.618	.696	.745	.809	.769	.844	.846	.761 ± .077	.618	.631	.651	.708	.701	.724	.744	.683 ± .045
$\overline{\lvert F \rvert}$	.602	.674	.724	.742	.705	.754	.831	.719 ± .066	.602	.607	.629	.645	.628	.681	.744	.648 ± .046
f_r	.522	.542	.524	.551	.534	.574	.492	.534 ± .024	.522	.507	.509	.465	.558	.483	.605	.521 ± .044
$\bar{z}^{(p)}$	.520	.540	.523	.528	.506	.475	.631	.532 ± .045	.520	.521	.521	.555	.544	.595	.488	.535 ± .031
D_w^U	.506	.513	.506	.509	.466	.484	.569	.508 ± .030	.506	.499	.494	.505	.485	.569	.558	.517 ± .031
D_w^T	.521	.546	.508	.518	.542	.590	.431	.522 ± .045	.521	.525	.492	.490	.478	.578	.535	.517 ± .032
Q_w	.512	.504	.506	.516	.522	.484	.415	.494 ± .034	.512	.507	.505	.516	.496	.543	.558	.520 ± .021
$\tilde{F}_n$	.557	.730	.883	.846	.909	—	—	.785 ± .129	.557	.567	.621	.618	.657	.701	.703	.632 ± .054
12 m.	.608	.668	.749	.811	.840	.838	.946	.780 ± .106	.608	.629	.684	.673	.771	.733	.912	.716 ± .095
24 m.	.630	.688	.732	.780	.797	.818	.847	.756 ± .071	.630	.645	.679	.710	.690	.748	.786	.698 ± .051

Table 3: Prediction accuracies for varying experimental setups and thresholds l_a, l_r. All: results for model trained on all predictors; $\overline{|F|}$: family size only; f_r: relative token frequency of parent only; $\bar{z}^{(p)}$: trending behavior of parent only; D_w^U: user dissemination only; D_w^T: thread dissemination only; Q_w: topicality only. $\tilde{F}_n$: results for model trained on new children only. 12 m.: results for model trained on context intervals of 12 months; 24 m.: 24 months. Best accuracies per column are highlighted in gray, second-best accuracies in light gray.

	l_a (δ_a)								l_r (δ_r)							
	0.0	0.8	1.6	2.4	3.2	4.0	4.8	$\mu \pm \sigma$	1.0	1.2	1.4	1.6	1.8	2.0	2.2	$\mu \pm \sigma$
$\overline{\lvert F \rvert}$	.202	.296	.344	.393	.458	.495	.489	.382±.101	.202	.208	.223	.253	.286	.261	.204	.234±.031
f_r	.153	.145	.140	.152	.127	.128	.164	.144±.013	.153	.153	.148	.144	.129	.122	.125	.139±.012
$\bar{z}^{(p)}$	.168	.150	.149	.145	.154	.136	.138	.149±.010	.168	.172	.178	.179	.195	.218	.264	.196±.032
D_w^U	.156	.134	.119	.098	.084	.070	.058	.103±.033	.156	.156	.149	.136	.130	.117	.096	.134±.021
D_w^T	.162	.139	.124	.105	.081	.090	.067	.110±.031	.162	.158	.150	.145	.132	.147	.145	.148±.009
Q_w	.158	.137	.125	.108	.096	.081	.083	.113±.027	.158	.155	.152	.143	.128	.135	.167	.148±.013

Table 4: Importance loadings for individual features. Features as in Table 3. Importance loadings are calculated for different values of the thresholds l_a and l_r. Highest importance loadings for features are highlighted in gray, second-highest in light gray.

The strongest predictor for both models is type frequency with a feature importance of 39.3% and 25.3%, respectively (Table 4). Models trained only on this feature already achieve accuracies of 74.2% and 64.5%, respectively (Table 3). However, the effect of $\overline{|F|}$ is reversed: while larger morphological families have higher absolute growth values, which is in accordance with theories of lexical growth based on preferential attachment (Steyvers and Tenenbaum, 2005), smaller morphological families have higher relative growth rates. This can be explained by the observation that a large family needs much higher increases in family size to have the same relative growth rate as a small family. The fact that larger families are generally more likely to grow does not seem to counteract this imbalance.

We now analyze how different design choices impact the performance of our model.

Thresholds l_a and l_r. We systematically vary l_a and l_r in the range $0.0 \leq l_a \leq 4.8$ and $1.0 \leq l_r \leq 2.2$ to examine their influence on performance.[7] We find that accuracies for predicting larger δ_a and δ_r are considerably higher than for smaller increases (Table 3). For $l_a = 4.8$ (i.e., the family size increases by more than 4.8 members on average), the model has an error rate of 15.4%, which is less than half compared to the error rate of 38.2% for $l_a = 0.0$. This striking result is in line with studies on the predictability of extreme events in social media (Miotto and Altmann, 2014) and statistical physics (Hallerberg and Kantz, 2008) showing that extreme events are generally better predictable than non-extreme events.

We then train single-feature models for varying l_a and l_r. The best predictor for all values of l_a and l_r is $\overline{|F|}$ (Table 3). The overall second-best predictor is $\bar{z}^{(p)}$, even though it is (sometimes only marginally) outperformed by f_r and D_w^Ton several values of l_a and l_r.

To further analyze the relative importance of inidividual features, we examine the RF feature

[7]The number of positive examples with $\delta_a > 4.8$ and $\delta_r > 2.2$ is too small for achieving reliable results.

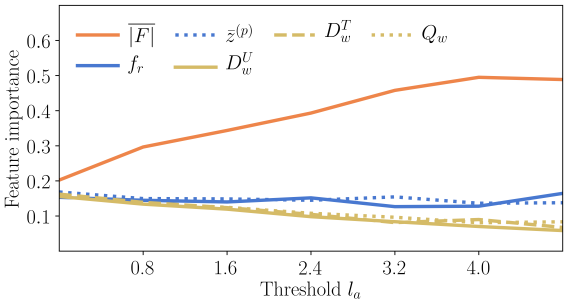


Figure 2: Feature importances of MFEP RF models (δ_a) for varying the absolute growth threshold l_a. The predictors are grouped into the classes *type frequency-based* ($\overline{|F|}$), *token frequency-based* (f_r, $\bar{z}^{(p)}$), and *dissemination-based* (D_w^U, D_w^T, Q_w).

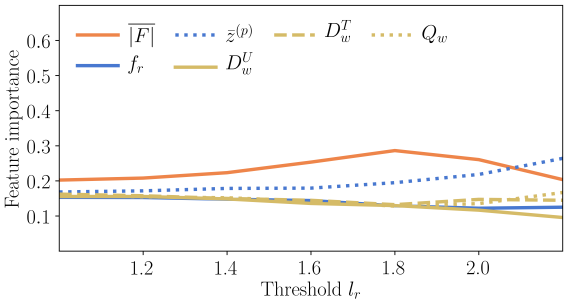


Figure 3: Feature importances of MFEP RF models (δ_r) for varying the relative growth threshold l_r. The predictors are grouped into the classes *type frequency-based* ($\overline{|F|}$), *token frequency-based* (f_r,$\bar{z}^{(p)}$), and *dissemination-based* (D_w^U, D_w^T, Q_w).

importance loadings for varying l_a and l_r (Table 4, Figure 2, Figure 3). While $\overline{|F|}$ is again the best predictor overall, $\bar{z}^{(p)}$much more clearly comes out as the second-best predictor: especially for δ_r, it steadily increases with higher values for l_r and even surpasses $\overline{|F|}$ for $l_r = 2.2$.

These results indicate that while the family size is most predictive of morphological family growth in general, high growth rates are particularly likely for families of trending parents (most of which have initially small family sizes). An example for the second case is the burst in the "trump" family before the 2016 presidential election illustrated in Section 1 (Figure 1, Table 1). This would explain how small morphological families can grow in the first place given the overall dominating importance of a large family size: small families need exogenous forcing (Altmann et al., 2011), i.e., external events leading to a burst in token frequency and a subsequent increase in type frequency. In order to test this hypothesis, we retrain the model for δ_a on small families ($1.5 \leq \overline{|F|} \leq 1.6$), varying

	l_a						
	0.0	0.2	0.4	0.6	0.8	1.0	$\mu \pm \sigma$
f_r	.175	.187	.196	.193	.165	.190	.184±.011
$\bar{z}^{(p)}$	.222	.216	.223	.253	.269	.234	.236±.019
D_w^U	.183	.198	.190	.181	.179	.190	.187±.006
D_w^T	.212	.196	.201	.192	.195	.233	.205±.014
Q_w	.208	.202	.190	.181	.192	.153	.188±.018

Table 5: Importance loadings for individual features with small families ($1.5 \leq \overline{|F|} \leq 1.6$). Features as in Table 3. Importance loadings are calculated for different values of the thresholds l_a. Highest importance loadings for features are highlighted in gray, second-highest in light gray.

$0.0 \leq l_a \leq 1$ (which is the range of δ_a for families of that size); $\bar{z}^{(p)}$has the highest feature importance for all values of l_a (Table 5).[8]

Furthermore, it is interesting to note that the frequency-based as well as the dissemination-based measures are considerably clumped together in feature importance space for absolute growth δ_a, with the frequency-based predictors topping the dissemination-based ones. This is in line with recent work on the relative importance of frequency and social dissemination in lexical change (Stewart and Eisenstein, 2018). Higher values in these features correlate with a higher likelihood of growth, except for topicality: here, growth is more likely with lower topicality, which as discussed above indicates higher topical dissemination.

Length of $i^{(c)}$ and $i^{(p)}$. In previous experiments, the length of $i^{(c)}$ and $i^{(p)}$ was set to 18 and 6 months, respectively. We now analyze how the choice of the interval length, specifically of the length of $i^{(c)}$, influences the performance of our MFEP model. We retrain the model for $0.0 \leq l_a \leq 4.8$ and $1.0 \leq l_r \leq 2.2$ with $|i^{(c)}| = 12$ and $|i^{(c)}| = 24$, i.e., the context intervals are six months shorter and longer than previously. The length of the probe interval is kept unchanged, $|i^{(p)}| = 6$.

The performances of the two MFEP models is comparable with $|i^{(c)}| = 18$ (Table 3). Both show top performance at $l_a = 4.8$ and $l_r = 2.2$. However, it is interesting to note that the performance with $|i^{(c)}| = 12$ tends to be better than $|i^{(c)}| = 24$ with large values for l_a and l_r, but worse with smaller values, suggesting that shorter context intervals have an advantage in predicting large increases in family size while longer context intervals have an advantage in predicting smaller increases.

[8]Even smaller families do not allow to vary l_a on such a large range due to data sparsity, but show similar results.

New children. In our main study design, growth in the families is not necessarily due to *new* children being added to the family, i.e., due to an increase of $|\tilde{F}_n|$. A rare but established English word $w \in \tilde{F}_o$ that only occurs a couple of times in the data counts as much to the growth as an innovative form. Here, we try to exclude fluctuations due to $\tilde{F}_o$ by excluding all words in the data that are listed on a comprehensive list of English words encompassing over 400,000 word types, an independent estimate of established words.[9] Training the model on the resulting data, we find that accuracies tend be be higher for δ_a[10] but lower for δ_r than corresponding accuracies on the full dataset (Table 3). This result indicates that our model is not only capable of forecasting the evolution of the entire family but also of predicting the birth of new morphological children.

Error analysis. The segmentation algorithm is doomed to produce a certain number of false positives. To get a clearer picture of its accuracy, we manually examine 500 randomly selected families from one month in the data. Macro-averaged over families, 8.8% of the words are errors, i.e., they do not belong to the morphological family assigned by the algorithm. However, the error rate is not distributed evenly: only 10 of the 500 families are responsible for more than 60% of the errors.

One frequent source of erroneous segmentations is incorrect orthography. The word "representatives", e.g., is frequently written as "represenatives" due to its being pronounced without the consonant "t". The algorithm then segments "represenatives" into "re+pre+senate+ive+s" and adds it to $\tilde{F}$("senate"). Another frequent case is the erroneous segmentation of emphatical repetitions of vowels, e.g., "heyy" is segmented as "hey+y" and added to $\tilde{F}$("hey"). Such false positives are a major source of distortion in the data.

7 Related work

Morphological families and productivity. The concept of morphological families was introduced in psycholinguistic work on lexical processing (Schreuder and Baayen, 1997; de Jong et al., 2000; del Prado Martín et al., 2004). These studies show that response latencies in lexical decision are not only influenced by *token frequency* but also by *type frequency*, i.e., the size of their morphological family. Morphological families have also been used for analyzing lexical change on historical time scales (Keller and Schultz, 2013, 2014). However, this work is not comparable to our study since it relies on dictionaries, which typically exclude the transparently formed, non-entrenched words in $\tilde{F}_n$ we are interested in (Bauer, 2001).

The main question of our study (how can we predict the growth of morphological families?) is related to a long-standing problem in traditional linguistic scholarship, i.e., what factors influence morphological productivity. The productivity of a morpheme is defined as its propensity to be used in novel combinations and traditionally understood to refer to bound morphemes (Haspelmath and Sims, 2010). Pierrehumbert and Granell (2018) highlight the fact that morphological productivity, just as morphology (Rácz et al., 2015) and other components of language (Labov, 1963) in general, is heavily influenced by social variation. Social groups differ in the morphological patterns they use and in the extent to which they extend these patterns to new words. This makes morphological productivity an exciting new area for future research in computational social science, and it further underscores the relevance of MFEP for that field.

Derivational morphology in NLP. Derivational morphology has received increasing attention in NLP recently. Key challenges include segmenting (Cotterell et al., 2016; Luo et al., 2017; Cotterell and Schütze, 2018), modeling the meaning (Lazaridou et al., 2013; Kisselew et al., 2015; Padó et al., 2016; Cotterell and Schütze, 2018), and predicting the form (Vylomova et al., 2017; Cotterell et al., 2017; Deutsch et al., 2018) as well as morphological well-formedness (Hofmann et al., 2020) of derivatives. Whereas all these studies approach derivational morphology from a *synchronic* standpoint, MFEP is to the best of our knowledge the first computational task that addresses *diachronic* aspects of derivation.

Lexical change in social media. Language change (Croft, 2000; Bybee, 2015) is most visible on the lexical level. New words like "detrumpify" attract attention, often becoming the subject of public discourse (Metcalf, 2002). Since innovations are taking place at a much faster rate on internet media (Crystal, 2004), social media have become a central resource for studies on lexical change over the last decade (Altmann et al., 2011; Garley

[9] The list is available on https://github.com/dwyl/english-words.

[10] We only trained the model for $0 \leq l_a \leq 3.2$ since there was not enough data for larger threshold values.

and Hockenmaier, 2012; Danescu-Niculescu-Mizil et al., 2013; Grieve et al., 2016; Kershaw et al., 2016; Sang, 2016; Stewart and Eisenstein, 2018; del Tredici and Fernández, 2018).

One central question in this field is: what factors determine whether a word will survive in the lexicon of an online community? Usage frequency is a well-known factor that influences the evolution of a word at historical time scales (Pagel et al., 2007). Studies on lexical change in online groups have shown that this is also true for shorter time scales (Altmann et al., 2011; Stewart and Eisenstein, 2018). Another main factor is the dissemination of a word, i.e., how widely a word is spread across different social and linguistic contexts. Generally, the more disseminated a word is, the more likely it is to grow. This holds for social dissemination across users and threads (Altmann et al., 2011) as well as linguistic dissemination across different lexical collocations (Stewart and Eisenstein, 2018).

The studies mentioned so far focus on token frequency. An exciting new approach looks instead at the meaning of words using diachronic word embeddings (Hamilton et al., 2016). del Tredici et al. (2019), e.g., explore short-term meaning shifts on Reddit and identify considerable changes even within a period of eight years.

A main goal of this study is to add a third approach to studies on lexical change in social media besides word frequency and word embeddings: word families. From a linguistic point of view, these three approaches can be viewed to be complementary: whereas word frequency is *context-independent*, both word embeddings and word families reflect *context-sensitive* measures. However, while word embeddings reflect proximity on the *utterance* level (which words are close to each other in spoken sentences?), word families reflect proximity on the *system* level (which words are close to each other in the mental lexicon?).

8 Conclusion

In this paper, we have proposed MFEP (Morphological Family Expansion Prediction), a new task that aims at predicting how morphological families evolve over time. We have shown that changes in morphological family size provide a fresh look at topical dynamics in social media, thus complementing token frequency as a metric.

Furthermore, we have presented a random forest model for MFEP that achieves very good accuracies, particularly in predicting extreme growth in morphological family size. The strongest predictor of growth is the morphological family size itself, an endogenous factor. However, the initial growth of small families is mainly driven by the trending behavior of the parent, an exogenous factor. This reflection of external events makes morphological families a promising tools for various fields drawing upon NLP techniques for tracing temporal dynamics in text (e.g., virality detection).

Overall, we see our study as an exciting step in the direction of bringing together computational social science and derivational morphology. In future work, we intend to further fine-tune our methodological apparatus for tackling MFEP.

Acknowledgements

Valentin Hofmann was funded by the Arts and Humanities Research Council and the German Academic Scholarship Foundation. This research was also supported by the European Research Council (Grant No. 740516). We thank the reviewers for their detailed and helpful comments.

References

Eduardo G. Altmann, Janet B. Pierrehumbert, and Adilson E. Motter. 2011. Niche as a determinant of word fate in online groups. *PloS ONE*, 6(5).

John R. Anderson and Robert Milson. 1989. Human memory: An adaptive perspective. *Psychological Review*, 96(4):703–719.

Georgios Baskozos, John M. Dawes, Jean S. Austin, Ana Antunes-Martins, Lucy McDermott, Alex J. Clark, Teodora Trendafilova, Jon G. Lees, Stephen B. McMahon, Jeffrey S. Mogil, Christine Orengo, and David L. Bennett. 2019. Comprehensive analysis of long noncoding rna expression in dorsal root ganglion reveals cell-type specificity and dysregulation after nerve injury. *Pain*, 160(2):463–485.

Laurie Bauer. 2001. *Morphological productivity*. Cambridge University Press, Cambridge, UK.

Laurie Bauer. 2019. *Rethinking morphology*. Edinburgh University Press, Edinburgh, UK.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45:5–32.

Marc Brysbaert, Michaël Stevens, Paweł Mandera, and Emmanuel Keuleers. 2016. The impact of word prevalence on lexical decision times: Evidence from the dutch lexicon project 2. *Journal of Experimental Psychology: Human Perception and Performance*, 42(3):441–458.

Joan Bybee. 1995. Regular morphology and the lexicon. *Language and Cognitive Processes*, 10(425-455).

Joan Bybee. 2015. *Language change*. Cambridge University Press, Cambridge, UK.

Kenneth Church. 2000. Empirical estimates of adaptation: The chance of two noriegas is closer to p/2 than p2. In *Conference on Computational Linguistics (COLING) 18*, pages 180–186.

Kenneth W. Church and William A. Gale. 1995. Poisson mixtures. *Natural Language Engineering*, 1(2):163–190.

Ryan Cotterell and Hinrich Schütze. 2018. Joint semantic synthesis and morphological analysis of the derived word. *Transactions of the Association for Computational Linguistics*, 6:33–48.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016. A joint model of orthography and morphological segmentation. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 15*.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017. Paradigm completion for derivational morphology. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017*.

William Croft. 2000. *Explaining language change: An evolutionary approach*. Pearson, Harlow, UK.

David Crystal. 1997. *The Cambridge encyclopedia of the English language*. Cambridge University Press, Cambridge, UK.

David Crystal. 2004. *Language and the internet*. Cambridge University Press, Cambridge, UK.

Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. No country for old members: User lifecycle and linguistic change in online communities. In *The Web Conference (WWW) 22*.

Fermín del Prado Martín, Raymond Bertram, Tuomo Häikiö, Robert Schreuder, and R. Harald Baayen. 2004. Morphological family size in a morphologically rich language: The case of finnish compared with dutch and hebrew. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(6):1271–1278.

Marco del Tredici and Raquel Fernández. 2018. The road to success: Assessing the fate of linguistic innovations in online communities. In *International Conference on Computational Linguistics (COLING) 27*.

Marco del Tredici, Raquel Fernández, and Gemma Boleda. 2019. Short-term meaning shift: A distributional exploration. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 17*.

Daniel Deutsch, John Hewitt, and Dan Roth. 2018. A distributional and orthographic aggregation model for english derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*.

Devin Gaffney and J. Nathan Matias. 2018. Caveat emptor, computational social science: Large-scale missing data in a widely-published reddit corpus. *PloS ONE*, 13(7):e0200162.

Matt Garley and Julia Hockenmaier. 2012. Beefmoves: Dissemination, diversity, and dynamics of english borrowings in a german hip hop forum. In *Annual Meeting of the Association for Computational Linguistics (ACL) 50*.

Jack Grieve, Andrea Nini, and Diansheng Guo. 2016. Analyzing lexical emergence in modern american english online. *English Language and Linguistics*, 21(1):99–127.

Sarah Hallerberg and Holger Kantz. 2008. Influence of the event magnitude on the predictability of an extreme event. *Physical Review E*, 77(1):011108.

William Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Annual Meeting of the Association for Computational Linguistics (ACL) 54*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Annual Meeting of the Association for Computational Linguistics (ACL) 49*.

Martin Haspelmath and Andrea D. Sims. 2010. *Understanding morphology*. Routledge, New York, NY.

Valentin Hofmann, Hinrich Schütze, and Janet B. Pierrehumbert. 2020. A graph auto-encoder model of derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Jörg D. Jescheniak and Willem J. Levelt. 1994. Word frequency effects in speech production: Retrieval of syntactic information and of phonological form. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 20(4):824–843.

Nivja de Jong, Robert Schreuder, and R. Harald Baayen. 2000. The morphological family size effect and morphology. *Language and Cognitive Processes*, 15(4/5):329–365.

Daniela B. Keller and Jörg Schultz. 2013. Connectivity, not frequency, determines the fate of a morpheme. *PloS ONE*, 8(7):e69945.

Daniela B. Keller and Jörg Schultz. 2014. Word formation is aware of morpheme family size. *PloS ONE*, 9(4):e93978.

Daniel Kershaw, Matthew Rowe, and Patrick Stacey. 2016. Towards modelling language innovation acceptance in online social networks. In *International Conference on Web Search and Data Mining (WSDM) 9*.

Max Kisselew, Sebastian Padó, Alexis Palmer, and Jan Šnajder. 2015. Obtaining a better understanding of distributional models of german derivational morphology. In *International Conference on Computational Semantics (IWCS) 11*.

William Labov. 1963. The social motivation of a sound change. *Word*, 19(3):273–309.

Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Annual Meeting of the Association for Computational Linguistics (ACL) 51*.

Jiaming Luo, Karthik Narasimhan, and Regina Barzilay. 2017. Unsupervised learning of morphological forests. *Transactions of the Association for Computational Linguistics*, 5:353–364.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA.

Allan Metcalf. 2002. *Predicting new words: The secret of their success*. Houghton Mifflin Company, Boston, MA.

José M. Miotto and Eduardo G. Altmann. 2014. Predictability of extreme events in social media. *PloS ONE*, 9(11):e111506.

Marcelo A. Montemurro and Damián H. Zanette. 2016. Coherent oscillations in word-use data from 1700 to 2008. *Palgrave Communications*, 2(1):405.

Sebastian Padó, Aurélie Herbelot, Max Kisselew, and Jan Šnajder. 2016. Predictability of distributional semantics in derivational word formation. In *International Conference on Computational Linguistics (COLING) 26*.

Mark Pagel, Quentin Atkinson, and Andrew Meade. 2007. Frequency of word-use predicts rates of lexical evolution throughout indo-european history. *Nature*, 449(7163):717–720.

Janet Pierrehumbert and Ramon Granell. 2018. On hapax legomena and morphological productivity. In *Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON) 15*.

Ingo Plag. 2003. *Word-formation in English*. Cambridge University Press, Cambridge, UK.

Péter Rácz, Janet Pierrehumbert, Jennifer Hay, and Viktória Papp. 2015. Morphological emergence. In Brian MacWhinney and William O'Grady, editors, *The handbook of language emergence*, pages 123–146. Wiley, Hoboken, NJ.

Erik T. Sang. 2016. Finding rising and falling words. In *Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*.

Robert Schreuder and R. Harald Baayen. 1997. How complex simplex words can be. *Journal of Memory and Language*, 37:118–139.

Ian Stewart and Jacob Eisenstein. 2018. Making "fetch" happen: The influence of social and linguistic context on nonstandard word growth and decline. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2018*.

Mark Steyvers and Joshua Tenenbaum. 2005. The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive Science*, 29(1):41–78.

Chenhao Tan and Lillian Lee. 2015. All who wander: On the prevalence and characteristics of multi-community engagement. In *International Conference on World Wide Web (WWW) 24*.

Yee W. Teh. 2006. A hierarchical bayesian language model based on pitman-yor processes. In *Annual Meeting of the Association for Computational Linguistics (ACL) 44*.

Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin, and Trevor Cohn. 2017. Context-aware prediction of derivational word-forms. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL) 15*.

5 DagoBERT

Contents

The third study focuses on a particular aspect of the predictability of derivational morphology, specifically which derivational affix is used to form a derivative. As opposed to the two studies reported in chapters 3 and 4, which examine derivatives in isolation, it provides the models with carrier sentences to guide them towards semantically and syntactically suitable affixes. This is also the first study of the thesis that analyzes a language model (specifically, BERT). I put a special emphasis on the question of how the input tokenization affects the language model's performance.

DagoBERT: Generating Derivational Morphology with a Pretrained Language Model

Valentin Hofmann[*‡], **Janet B. Pierrehumbert**[†*], **Hinrich Schütze**[‡]

[*]Faculty of Linguistics, University of Oxford
[†]Department of Engineering Science, University of Oxford
[‡]Center for Information and Language Processing, LMU Munich
valentin.hofmann@ling-phil.ox.ac.uk

Abstract

Can pretrained language models (PLMs) generate derivationally complex words? We present the first study investigating this question, taking BERT as the example PLM. We examine BERT's derivational capabilities in different settings, ranging from using the unmodified pretrained model to full finetuning. Our best model, DagoBERT (Derivationally and generatively optimized BERT), clearly outperforms the previous state of the art in derivation generation (DG). Furthermore, our experiments show that the input segmentation crucially impacts BERT's derivational knowledge, suggesting that the performance of PLMs could be further improved if a morphologically informed vocabulary of units were used.

1 Introduction

What kind of linguistic knowledge is encoded by pretrained language models (PLMs) such as ELMo (Peters et al., 2018), GPT-2 (Radford et al., 2019), and BERT (Devlin et al., 2019)? This question has attracted a lot of attention in NLP recently, with a focus on syntax (e.g., Goldberg, 2019) and semantics (e.g., Ethayarajh, 2019). It is much less clear what PLMs learn about other aspects of language. Here, we present the first study on the knowledge of PLMs about derivational morphology, taking BERT as the example PLM. Given an English cloze sentence such as `this jacket is _____ .` and a base such as `wear`, we ask: can BERT generate correct derivatives such as `unwearable`?

The motivation for this study is twofold. On the one hand, we add to the growing body of work on the linguistic capabilities of PLMs. Most PLMs segment words into subword units (Bostrom and Durrett, 2020), e.g., `unwearable` is segmented into `un`, `##wear`, `##able` by BERT's WordPiece tokenizer (Wu et al., 2016). The fact that many of these subword units are derivational affixes suggests that PLMs might acquire knowledge about derivational morphology (Table 1), but this has not been tested. On the other hand, we are interested in derivation generation (DG) per se, a task that has been only addressed using LSTMs (Cotterell et al., 2017; Vylomova et al., 2017; Deutsch et al., 2018), not models based on Transformers like BERT.

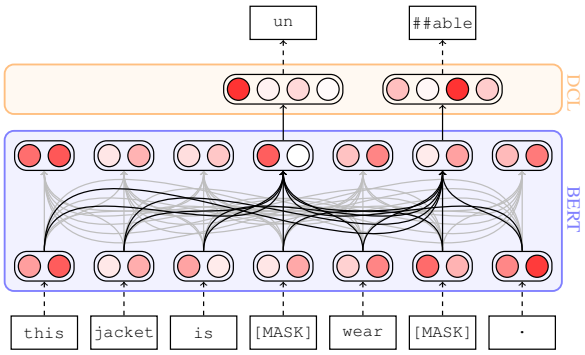


Figure 1: Basic experimental setup. We input sentences such as `this jacket is unwearable .` to BERT, mask out derivational affixes, and recover them using a derivational classification layer (DCL).

Type	Examples
Prefixes	`anti`, `auto`, `contra`, `extra`, `hyper`, `mega`, `mini`, `multi`, `non`, `proto`, `pseudo`
Suffixes	`##able`, `##an`, `##ate`, `##ee`, `##ess`, `##ful`, `##ify`, `##ize`, `##ment`, `##ness`, `##ster`

Table 1: Examples of derivational affixes in the BERT WordPiece vocabulary. Word-internal WordPiece tokens are marked with `##` throughout the paper.

Contributions. We develop the first framework for generating derivationally complex English words with a PLM, specifically BERT, and analyze BERT's performance in different settings. Our best model, DagoBERT (Derivationally and generatively optimized BERT), clearly outperforms an LSTM-based model, the previous state of the art.

We find that DagoBERT's errors are mainly due to syntactic and semantic overlap between affixes. Furthermore, we show that the input segmentation impacts how much derivational knowledge is available to BERT, both during training and inference. This suggests that the performance of PLMs could be further improved if a morphologically informed vocabulary of units were used. We also publish the largest dataset of derivatives in context to date.[1]

2 Derivational Morphology

Linguistics divides morphology into inflection and derivation. Given a lexeme such as `wear`, while inflection produces word forms such as `wears`, derivation produces new lexemes such as `unwearable`. There are several differences between inflection and derivation (Haspelmath and Sims, 2010), two of which are particularly important for the task of DG.[2]

First, derivation covers a much larger spectrum of meanings than inflection (Acquaviva, 2016), and it is not possible to predict in general with which of them a particular lexeme is compatible. This is different from inflectional paradigms, where it is automatically clear whether a certain form will exist (Bauer, 2019). Second, the relationship between form and meaning is more varied in derivation than inflection. On the one hand, derivational affixes tend to be highly polysemous, i.e., individual affixes can represent a number of related meanings (Lieber, 2019). On the other hand, several affixes can represent the same meaning, e.g., `ity` and `ness`. While such competing affixes are often not completely synonymous as in the case of `hyperactivity` and `hyperactiveness`, there are examples like `purity` and `pureness` or `exclusivity` and `exclusiveness` where a semantic distinction is more difficult to gauge (Bauer et al., 2013; Plag and Balling, 2020). These differences make learning functions from meaning to form harder for derivation than inflection.

Derivational affixes differ in how productive they are, i.e., how readily they can be used to create new lexemes (Plag, 1999). While the suffix `ness`, e.g., can attach to practically all English adjectives, the suffix `th` is much more limited in its scope of applicability. In this paper, we focus on productive affixes such as `ness` and exclude unproductive affixes such as `th`. Morphological productivity has been the subject of much work in psycholinguistics since it reveals implicit cognitive generalizations (see Dal and Namer (2016) for a review), making it an interesting phenomenon to explore in PLMs. Furthermore, in the context of NLP applications such as sentiment analysis, productively formed derivatives are challenging because they tend to have very low frequencies and often only occur once (i.e., they are hapaxes) or a few times in large corpora (Mahler et al., 2017). Our focus on productive derivational morphology has crucial consequences for dataset design (Section 3) and model evaluation (Section 4) in the context of DG.

3 Dataset of Derivatives

We base our study on a new dataset of derivatives in context similar in form to the one released by Vylomova et al. (2017), i.e., it is based on sentences with a derivative (e.g., `this jacket is unwearable .`) that are altered by masking the derivative (`this jacket is _____ .`). Each item in the dataset consists of (i) the altered sentence, (ii) the derivative (`unwearable`) and (iii) the base (`wear`). The task is to generate the correct derivative given the altered sentence and the base. We use sentential contexts rather than tags to represent derivational meanings because they better reflect the semantic variability inherent in derivational morphology (Section 2). While Vylomova et al. (2017) use Wikipedia, we extract the dataset from Reddit.[3] Since productively formed derivatives are not part of the language norm initially (Bauer, 2001), social media is a particularly fertile ground for our study.

For determining derivatives, we use the algorithm introduced by Hofmann et al. (2020a), which takes as input a set of prefixes, suffixes, and bases and checks for each word in the data whether it can be derived from a base using a combination of prefixes and suffixes. The algorithm is sensitive to morpho-orthographic rules of English (Plag, 2003), e.g., when `ity` is removed from `applicability`, the result is `applicable`, not `applicabil`. Here, we use BERT's prefixes, suffixes, and bases as input to the algorithm. Drawing upon a comprehensive list of 52 productive pre-

[1] We make our code and data publicly available at https://github.com/valentinhofmann/dagobert.

[2] It is important to note that the distinction between inflection and derivation is fuzzy (ten Hacken, 2014).

[3] We draw upon the entire Baumgartner Reddit Corpus, a collection of all public Reddit posts available at https://files.pushshift.io/reddit/comments/.

		P			S			PS		
Bin	μ_f	n_d	n_s	Examples	n_d	n_s	Examples	n_d	n_s	Examples
B1	.041	60,236	60,236	`antijonny`	39,543	39,543	`takeoverness`	20,804	20,804	`unaggregateable`
B2	.094	39,181	90,857	`antiastronaut`	22,633	52,060	`alaskaness`	8,661	19,903	`unnicknameable`
B3	.203	26,967	135,509	`antiyale`	14,463	71,814	`blockbusterness`	4,735	23,560	`unbroadcastable`
B4	.423	18,697	196,295	`antihomework`	9,753	100,729	`abnormalness`	2,890	29,989	`unbrewable`
B5	.868	13,401	287,788	`antiboxing`	6,830	145,005	`legalness`	1,848	39,501	`ungooglable`
B6	1.750	9,471	410,410	`antiborder`	4,934	211,233	`tragicness`	1,172	50,393	`uncopyrightable`
B7	3.515	6,611	573,442	`antimafia`	3,580	310,109	`lightweightness`	802	69,004	`unwashable`

Table 2: Data summary statistics. The table shows statistics of the data used in the study by frequency bin and affix type. We also provide example derivatives with `anti` (P), `ness` (S), and `un##able` (PS) for the different bins. μ_f: mean frequency per billion words; n_d: number of distinct derivatives; n_s: number of context sentences.

fixes and 49 productive suffixes in English (Crystal, 1997), we find that 48 and 44 of them are contained in BERT's vocabulary. We assign all fully alphabetic words with more than 3 characters in BERT's vocabulary except for stopwords and previously identified affixes to the set of bases, yielding a total of 20,259 bases. We then extract every sentence including a word that is derivable from one of the bases using at least one of the prefixes or suffixes from all publicly available Reddit posts.

The sentences are filtered to contain between 10 and 100 words, i.e., they provide more contextual information than the example sentence above.[4] See Appendix A.1 for details about data preprocessing. The resulting dataset comprises 413,271 distinct derivatives in 123,809,485 context sentences, making it more than two orders of magnitude larger than the one released by Vylomova et al. (2017).[5] To get a sense of segmentation errors in the dataset, we randomly pick 100 derivatives for each affix and manually count missegmentations. We find that the average precision of segmentations in the sample is .960±.074, with higher values for prefixes (.990±.027) than suffixes (.930±.093).

For this study, we extract all derivatives with a frequency $f \in [1, 128)$ from the dataset. We divide the derivatives into 7 frequency bins with $f = 1$ (B1), $f \in [2, 4)$ (B2), $f \in [4, 8)$ (B3), $f \in [8, 16)$ (B4), $f \in [16, 32)$ (B5), $f \in [32, 64)$ (B6), and $f \in [64, 128)$ (B7). Notice that we focus on low-frequency derivatives since we are interested in productive derivational morphology (Section 2). In addition, BERT is likely to have seen high-frequency derivatives multiple times during pretraining and might be able to predict the affix because it has memorized the connection between the base and the affix, not because it has knowledge of derivational morphology. BERT's pretraining corpus has 3.3 billion words, i.e., words in the lower frequency bins are very unlikely to have been seen by BERT before. This observation also holds for average speakers of English, who have been shown to encounter at most a few billion word tokens in their lifetime (Brysbaert et al., 2016).

Regarding the number of affixes, we confine ourselves to three cases: derivatives with one prefix (P), derivatives with one suffix (S), and derivatives with one prefix and one suffix (PS).[6] We treat these cases separately because they are known to have different linguistic properties. In particular, since suffixes in English can change the POS of a lexeme, the syntactic context is more affected by suffixation than by prefixation. Table 2 provides summary statistics for the seven frequency bins as well as example derivatives for P, S, and PS. For each bin, we randomly split the data into 60% training, 20% development, and 20% test. Following Vylomova et al. (2017), we distinguish the lexicon settings SPLIT (no overlap between bases in train, dev, and test) and SHARED (no constraint on overlap).

4 Experiments

4.1 Setup

To examine whether BERT can generate derivationally complex words, we use a cloze test: given a sentence with a masked word such as `this jacket is _____ .` and a base such as `wear`, the task is to generate the correct derivative such as `unwearable`. The cloze setup has been previously used in psycholinguistics to probe derivational morphology (Pierrehumbert, 2006; Apel and

[4] We also extract the preceding and following sentence for future studies on long-range dependencies in derivation. However, we do not exploit them in this work.

[5] Due to the large number of prefixes, suffixes, and bases, the dataset can be valuable for any study on derivational morphology, irrespective of whether or not it focuses on DG.

[6] We denote affix bundles, i.e., combinations of prefix and suffix, by juxtaposition, e.g., `un##able`.

Lawrence, 2011) and was introduced to NLP in this context by Vylomova et al. (2017).

In this work, we frame DG as an affix classification task, i.e., we predict which affix is most likely to occur with a given base in a given context sentence.[7] More formally, given a base b and a context sentence $\mathbf{x}$ split into left and right contexts $\mathbf{x}^{(l)} = (x_1, \ldots, x_{d-1})$ and $\mathbf{x}^{(r)} = (x_{d+1}, \ldots, x_n)$, with x_d being the masked derivative, we want to find the affix $\hat{a}$ such that

$$\hat{a} = \arg\max_a P\left(\psi(b, a) | \mathbf{x}^{(l)}, \mathbf{x}^{(r)}\right), \quad (1)$$

where ψ is a function mapping bases and affixes onto derivatives, e.g., $\psi(\texttt{wear}, \texttt{un\#\#able}) = \texttt{unwearable}$. Notice we do not model the function ψ itself, i.e., we only predict derivational categories, not the morpho-orthographic changes that accompany their realization in writing. One reason for this is that as opposed to previous work, our study focuses on low-frequency derivatives, for many of which ψ is not right-unique, e.g., `ungoogleable` and `ungooglable` or `celebrityness` and `celebritiness` occur as competing forms in the data.

As a result of the semantically diverse nature of derivation (Section 2), deciding whether a particular prediction $\hat{a}$ is correct or not is less straightforward than it may seem. Taking again the example sentence `this jacket is _____ .` with the masked derivative `unwearable`, compare the following five predictions:

- $\psi(b, \hat{a}) =$ `wearity`: ill-formed;
- $\psi(b, \hat{a}) =$ `wearer`: well-formed, syntactically incorrect (wrong POS);
- $\psi(b, \hat{a}) =$ `intrawearable`: well-formed, syntactically correct, semantically dubious;
- $\psi(b, \hat{a}) =$ `superwearable`: well-formed, syntactically correct, semantically possible, but did not occur in the example sentence;
- $\psi(b, \hat{a}) =$ `unwearable`: well-formed, syntactically correct, semantically possible, and did occur in the example sentence.

These predictions reflect increasing degrees of derivational knowledge. A priori, where to draw the line between correct and incorrect predictions on this continuum is not clear, especially with respect to the last two cases. Here, we apply the most conservative criterion: a prediction $\hat{a}$ is only judged correct if $\psi(b, \hat{a}) = x_d$, i.e., if $\hat{a}$ is the affix in the masked derivative. Thus, we ignore affixes that might potentially produce equally possible derivatives such as `superwearable`.

[7] In the case of PS, we predict which affix bundle (e.g., `un##able`) is most likely to occur.

Method	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$
HYP	.197	.228	.252	.278	.300	.315	.337	.272±.046
INIT	.184	.201	.211	.227	.241	.253	.264	.226±.027
TOK	.141	.157	.170	.193	.218	.245	.270	.199±.044
PROJ	.159	.166	.159	.175	.175	.184	.179	.171±.009

Table 3: Performance (MRR) of pretrained BERT for prefix prediction with different segmentations. Best score per column in gray, second-best in light-gray.

We use mean reciprocal rank (MRR), macro-averaged over affixes, as the evaluation measure (Radev et al., 2002). We calculate the MRR value of an individual affix a as

$$\text{MRR}_a = \frac{1}{|D_a|} \sum_{i \in D_a} \text{R}_i^{-1}, \quad (2)$$

where D_a is the set of derivatives containing a, and R_i is the predicted rank of a for derivative i. We set $\text{R}_i^{-1} = 0$ if $\text{R}_i > 10$. Denoting with $\mathcal{A}$ the set of all affixes, the final MRR value is given by

$$\text{MRR} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \text{MRR}_a \,. \quad (3)$$

4.2 Segmentation Methods

Since BERT distinguishes word-initial (`wear`) from word-internal (`##wear`) tokens, predicting prefixes requires the word-internal form of the base. However, only 795 bases in BERT's vocabulary have a word-internal form. Take as an example the word `unallowed`: both `un` and `allowed` are in the BERT vocabulary, but we need the token `##allowed`, which does not exist (BERT tokenizes the word into `una`, `##llo`, `##wed`). To overcome this problem, we test the following four segmentation methods:

HYP. We insert a hyphen between the prefix and the base in its word-initial form, yielding the tokens `un`, `-`, `allowed` in our example. Since both prefix and base are guaranteed to be in the BERT vocabulary (Section 3), and since there are no tokens starting with a hyphen in the BERT vocabulary, BERT always tokenizes words of the form prefix-hyphen-base into prefix, hyphen, and base, making this a natural segmentation for BERT.

	SHARED								SPLIT							
Model	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$
DagoBERT	.373	.459	.657	.824	.895	.934	.957	.728±.219	.375	.386	.390	.411	.412	.396	.417	.398±.014
BERT+	.296	.380	.497	.623	.762	.838	.902	.614±.215	.303	.313	.325	.340	.341	.353	.354	.333±.018
BERT	.197	.228	.252	.278	.300	.315	.337	.272±.046	.199	.227	.242	.279	.305	.307	.351	.273±.049
LSTM	.152	.331	.576	.717	.818	.862	.907	.623±.266	.139	.153	.142	.127	.121	.123	.115	.131±.013
RB	.064	.067	.064	.067	.065	.063	.066	.065±.001	.068	.064	.062	.064	.062	.064	.064	.064±.002

Table 4: Performance (MRR) of prefix (P) models. Best score per column in gray, second-best in light-gray.

	SHARED								SPLIT							
Model	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$
DagoBERT	.427	.525	.725	.868	.933	.964	.975	.774±.205	.424	.435	.437	.425	.421	.393	.414	.421±.014
BERT+	.384	.445	.550	.684	.807	.878	.921	.667±.197	.378	.387	.389	.380	.364	.364	.342	.372±.015
BERT	.229	.246	.262	.301	.324	.349	.381	.299±.052	.221	.246	.268	.299	.316	.325	.347	.289±.042
LSTM	.217	.416	.669	.812	.881	.923	.945	.695±.259	.188	.186	.173	.154	.147	.145	.140	.162±.019
RB	.071	.073	.069	.068	.068	.068	.068	.069±.002	.070	.069	.069	.071	.070	.069	.068	.069±.001

Table 5: Performance (MRR) of suffix (S) models. Best score per column in gray, second-best in light-gray.

	SHARED								SPLIT							
Model	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$
DagoBERT	.143	.355	.621	.830	.914	.940	.971	.682±.299	.137	.181	.199	.234	.217	.270	.334	.225±.059
BERT+	.103	.205	.394	.611	.754	.851	.918	.548±.296	.091	.128	.145	.182	.173	.210	.218	.164±.042
BERT	.082	.112	.114	.127	.145	.155	.190	.132±.032	.076	.114	.130	.177	.172	.226	.297	.170±.069
LSTM	.020	.338	.647	.781	.839	.882	.936	.635±.312	.015	.019	.026	.034	.041	.072	.081	.041±.024
RB	.002	.003	.003	.005	.006	.008	.012	.006±.003	.002	.004	.003	.006	.006	.007	.009	.005±.002

Table 6: Performance (MRR) of prefix-suffix (PS) models. Best score per column in gray, second-best in light-gray.

INIT. We simply use the word-initial instead of the word-internal form, segmenting the derivative into the prefix followed by the base, i.e., `un`, `allowed` in our example. Notice that this looks like two individual words to BERT since `allowed` is a word-initial unit.

TOK. To overcome the problem of INIT, we segment the base into word-internal tokens, i.e., our example is segmented into `un`, `##all`, `##owed`. This means that we use the word-internal counterpart of the base in cases where it exists.

PROJ. We train a projection matrix that maps embeddings of word-initial forms of bases to word-internal embeddings. More specifically, we fit a matrix $\hat{\mathbf{T}} \in \mathbb{R}^{m \times m}$ (m being the embedding size) via least squares,

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T}} ||\mathbf{ET} - \mathbf{E}_{\#\#}||_2^2, \qquad (4)$$

where $\mathbf{E}, \mathbf{E}_{\#\#} \in \mathbb{R}^{n \times m}$ are the word-initial and word-internal token input embeddings of bases with both forms. We then map bases with no word-internal form and a word-initial input token embedding $\mathbf{e}$ such as `allow` onto the projected word-internal embedding $\mathbf{e}^\top \hat{\mathbf{T}}$.

Model	SHARED	SPLIT
DagoBERT	.943	.615
LSTM	.824	.511
LSTM (V)	.830	.520

Table 7: Performance on Vylomova et al. (2017) dataset. We report accuracies for comparability. LSTM (V): LSTM in Vylomova et al. (2017). Best score per column in gray, second-best in light-gray.

We evaluate the four segmentation methods on the SHARED test data for P with pretrained $\text{BERT}_{\text{BASE}}$, using its pretrained language modeling head for prediction and filtering for prefixes. The HYP segmentation method performs best (Table 3) and is adopted for BERT models on P and PS.

4.3 Models

All BERT models use $\text{BERT}_{\text{BASE}}$ and add a derivational classification layer (DCL) with softmax activation for prediction (Figure 1). We examine three BERT models and two baselines. See Appendix A.2 for details about implementation, hyperparameter tuning, and runtime.

DagoBERT. We finetune both BERT and DCL on DG, a model that we call DagoBERT (short for

Type	Clusters
Prefixes	{bi, demi, fore, mini, proto, pseudo, semi, sub, tri}, {arch, extra, hyper, mega, poly, super, ultra}, {anti, contra, counter, neo, pro}, {mal, mis, over, under}, {inter, intra}, {auto, de, di, in, re, sur, un}, {ex, vice}, {non, post, pre}
Suffixes	{##al, ##an, ##ial, ##ian, ##ic, ##ite}, {##en, ##ful, ##ive, ##ly, ##y}, {##able, ##ish, ##less}, {##age, ##ance, ##ation, ##dom, ##ery, ##ess, ##hood, ##ism, ##ity, ##ment, ##ness}, {##ant, ##ee, ##eer, ##er, ##ette, ##ist, ##ous, ##ster}, {##ate, ##ify, ##ize}

Table 8: Prefix and suffix clusterings produced by Girvan-Newman after 4 graph splits on the DagoBERT confusion matrix. For reasons of space, we do not list clusters consisting of only one affix.

Derivationally **a**nd **g**eneratively **o**ptimized **BERT**). Notice that since BERT cannot capture statistical dependencies between masked tokens (Yang et al., 2019), all BERT-based models predict prefixes and suffixes independently in the case of PS.

BERT+. We keep the model weights of pretrained BERT fixed and only train DCL on DG. This is similar in nature to a probing task.

BERT. We use pretrained BERT and leverage its pretrained language modeling head as DCL, filtering for affixes, e.g., we compute the softmax only over prefixes in the case of P.

LSTM. We adapt the approach described in Vylomova et al. (2017), which combines the left and right contexts $\mathbf{x}^{(l)}$ and $\mathbf{x}^{(r)}$ of the masked derivative by means of two BiLSTMs with a character-level representation of the base. To allow for a direct comparison with BERT, we do not use the character-based decoder proposed by Vylomova et al. (2017) but instead add a dense layer for the prediction. For PS, we treat prefix-suffix bundles as units (e.g., `un##able`).

In order to provide a strict comparison to Vylomova et al. (2017), we also evaluate our LSTM and best BERT-based model on the suffix dataset released by Vylomova et al. (2017) against the reported performance of their encoder-decoder model.[8] Notice Vylomova et al. (2017) show that providing the LSTM with the POS of the derivative increases performance. Here, we focus on the more general case where the POS is not known and hence do not consider this setting.

Random Baseline (RB). The prediction is a random ranking of all affixes.

[8]The dataset is available at https://github.com/ivri/dmorph. While Vylomova et al. (2017) take morpho-orthographic changes into account, we only predict affixes, not the accompanying changes in orthography (Section 4.1).

5 Results

5.1 Overall Performance

Results are shown in Tables 4, 5, and 6. For P and S, DagoBERT clearly performs best. Pretrained BERT is better than LSTM on SPLIT but worse on SHARED. BERT+ performs better than pretrained BERT, even on SPLIT (except for S on B7). S has higher scores than P for all models and frequency bins, which might be due to the fact that suffixes carry POS information and hence are easier to predict given the syntactic context. Regarding frequency effects, the models benefit from higher frequencies on SHARED since they can connect bases with certain groups of affixes.[9] For PS, DagoBERT also performs best in general but is beaten by LSTM on one bin. The smaller performance gap as compared to P and S can be explained by the fact that DagoBERT as opposed to LSTM cannot learn statistical dependencies between two masked tokens (Section 4).

The results on the dataset released by Vylomova et al. (2017) confirm the superior performance of DagoBERT (Table 7). DagoBERT beats the LSTM by a large margin, both on SHARED and SPLIT. We also notice that our LSTM (which predicts derivational categories) has a very similar performance to the LSTM encoder-decoder proposed by Vylomova et al. (2017).

5.2 Patterns of Confusion

We now analyze in more detail the performance of the best performing model, DagoBERT, and contrast it with the performance of pretrained BERT. As a result of our definition of correct predictions (Section 4.1), the set of incorrect predictions is heterogeneous and potentially contains affixes resulting in equally possible derivatives. We are hence interested in patterns of confusion in the data.

[9]The fact that this trend also holds for pretrained BERT indicates that more frequent derivatives in our dataset also appeared more often in the data used for pretraining BERT.

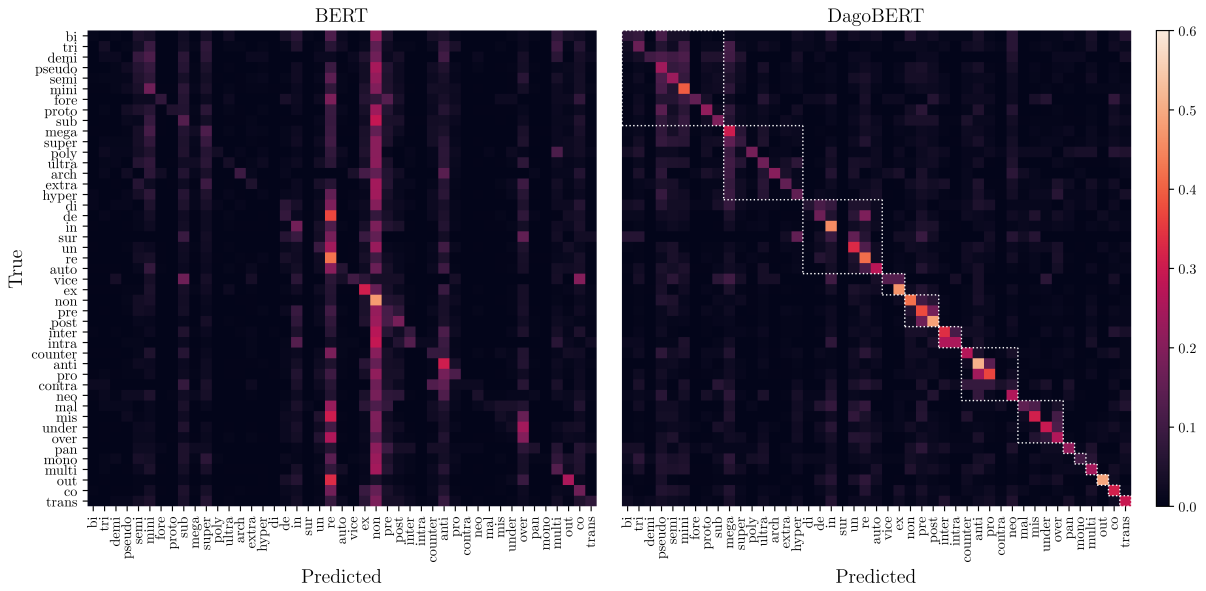


Figure 2: Prefixes predicted by BERT (left) and DagoBERT (right). Vertical lines indicate that a prefix has been overgenerated (particularly `re` and `non` in the left panel). The white boxes in the right panel highlight the clusters produced by Girvan-Newman after 4 graph splits.

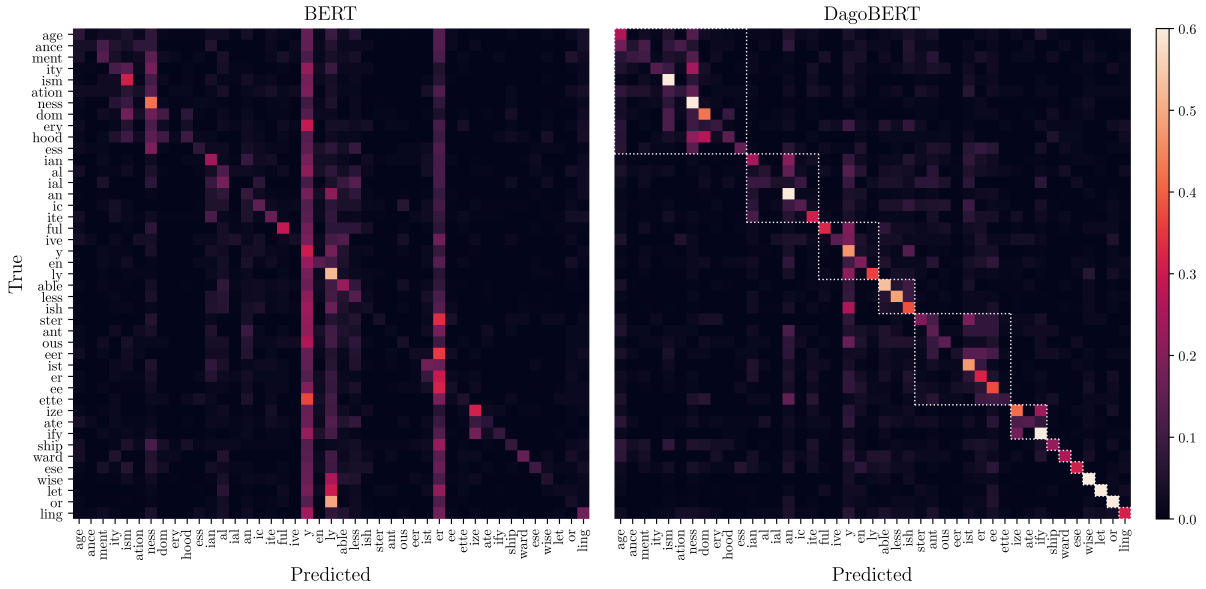


Figure 3: Suffixes predicted by pretrained BERT (left) and DagoBERT (right). Vertical lines indicate that a suffix has been overgenerated (particularly `y`, `ly`, and `er` in the left panel). The white boxes in the right panel highlight the clusters produced by Girvan-Newman after 4 graph splits.

We start by constructing the row-normalized confusion matrix $\mathbf{C}$ for the predictions of DagoBERT on the hapax derivatives (B1, SHARED) for P and S. Based on $\mathbf{C}$, we create a confusion graph $\mathcal{G}$ with adjacency matrix $\mathbf{G}$, whose elements are

$$G_{ij} = \lceil C_{ij} - \theta \rceil, \tag{5}$$

i.e., there is a directed edge from affix i to affix j if i was misclassified as j with a probability greater than θ. We set θ to 0.08.[10] To uncover the community structure of $\mathcal{G}$, we use the Girvan-Newman algorithm (Girvan and Newman, 2002), which clusters the graph by iteratively removing the edge with the highest betweenness centrality.

The resulting clusters reflect linguistically interpretable groups of affixes (Table 8). In particular, the suffixes are clustered in groups with common

[10] We tried other values of θ, but the results were similar.

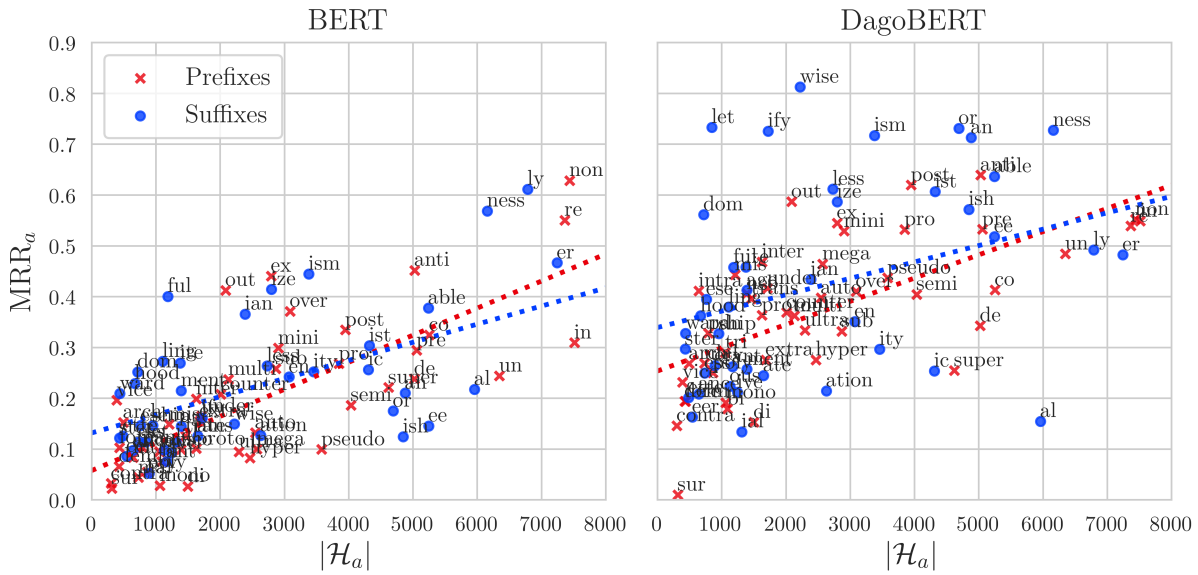


Figure 4: Correlation between number of hapaxes and MRR for pretrained BERT (left) and DagoBERT (right) on B1. The highly productive suffix `y` at $(12662, 0.49)$ (left) and $(12662, 0.62)$ (right) is not shown.

POS. These results are confirmed by plotting the confusion matrix with an ordering of the affixes induced by all clusterings of the Girvan-Newman algorithm (Figure 2, Figure 3). They indicate that even when DagoBERT does not predict the affix occurring in the sentence, it tends to predict an affix semantically and syntactically congruent with the ground truth (e.g., `ness` for `ity`, `ify` for `ize`, `inter` for `intra`). In such cases, it is often a more productive affix that is predicted in lieu of a less productive one. Furthermore, DagoBERT frequently confuses affixes denoting points on the same scale, often antonyms (e.g., `pro` and `anti`, `pre` and `post`, `under` and `over`). This can be related to recent work showing that BERT has difficulties with negated expressions (Ettinger, 2020; Kassner and Schütze, 2020). Pretrained BERT shows similar confusion patterns overall but overgenerates several affixes much more strongly than DagoBERT, in particular `re`, `non`, `y`, `ly`, and `er`, which are among the most productive affixes in English (Plag, 1999, 2003).

To probe the impact of productivity more quantitatively, we measure the cardinality of the set of hapaxes formed by means of a particular affix a in the entire dataset, $|\mathcal{H}_a|$, and calculate a linear regression to predict the MRR values of affixes based on $|\mathcal{H}_a|$. $|\mathcal{H}_a|$ is a common measure of morphological productivity (Baayen and Lieber, 1991; Pierrehumbert and Granell, 2018). This analysis shows a significant positive correlation for both prefixes ($R^2 = .566$, $F(1, 43) = 56.05$, $p < .001$) and suffixes ($R^2 = .410$, $F(1, 41) = 28.49$, $p < .001$): the more productive an affix, the higher its MRR value. This also holds for DagoBERT's predictions of prefixes ($R^2 = .423$, $F(1, 43) = 31.52$, $p < .001$) and suffixes ($R^2 = .169$, $F(1, 41) = 8.34$, $p < .01$), but the correlation is weaker, particularly in the case of suffixes (Figure 4).

5.3 Impact of Input Segmentation

We have shown that BERT can generate derivatives if it is provided with the morphologically correct segmentation. At the same time, we observed that BERT's WordPiece tokenizations are often morphologically incorrect, an observation that led us to impose the correct segmentation using hyphenation (HYP). We now examine more directly how BERT's derivational knowledge is affected by using the original WordPiece segmentations versus the HYP segmentations.

We draw upon the same dataset as for DG (SPLIT) but perform binary instead of multi-class classification, i.e., the task is to predict whether, e.g., `unwearable` is a possible derivative in the context `this jacket is _____ .` or not. As negative examples, we combine the base of each derivative (e.g., `wear`) with a randomly chosen affix different from the original affix (e.g., `##ation`) and keep the sentence context unchanged, resulting in a balanced dataset. We only use prefixed derivatives for this experiment.

	FROZEN								FINETUNED							
Segmentation	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$	B1	B2	B3	B4	B5	B6	B7	$\mu \pm \sigma$
Morphological	.634	.645	.658	.675	.683	.692	.698	.669±.022	.762	.782	.797	.807	.800	.804	.799	.793±.015
WordPiece	.572	.578	.583	.590	.597	.608	.608	.591±.013	.739	.757	.766	.769	.767	.755	.753	.758±.010

Table 9: Performance (accuracy) of BERT on morphological well-formedness prediction with morphologically correct segmentation versus WordPiece tokenization. Best score per column in gray.

We train binary classifiers using $\text{BERT}_{\text{BASE}}$ and one of two input segmentations, the morphologically correct segmentation or BERT's WordPiece tokenization. The BERT output embeddings for all subword units belonging to the derivative in question are max-pooled and fed into a dense layer with a sigmoid activation. We examine two settings: training only the dense layer while keeping BERT's model weights frozen (FROZEN), or finetuning the entire model (FINETUNED). See Appendix A.3 for details about implementation, hyperparameter tuning, and runtime.

Morphologically correct segmentation consistently outperforms WordPiece tokenization, both on FROZEN and FINETUNED (Table 9). We interpret this in two ways. Firstly, the type of segmentation used by BERT impacts how much derivational knowledge can be learned, with positive effects of morphologically valid segmentations. Secondly, the fact that there is a performance gap even for models with frozen weights indicates that a morphologically invalid segmentation can blur the derivational knowledge that is in principle available and causes BERT to force semantically unrelated words to have similar representations. Taken together, these findings provide further evidence for the crucial importance of morphologically valid segmentation strategies in language model pretraining (Bostrom and Durrett, 2020).

6 Related Work

PLMs such as ELMo (Peters et al., 2018), GPT-2 (Radford et al., 2019), and BERT (Devlin et al., 2019) have been the focus of much recent work in NLP. Several studies have been devoted to the linguistic knowledge encoded by the parameters of PLMs (see Rogers et al. (2020) for a review), particularly syntax (Goldberg, 2019; Hewitt and Manning, 2019; Jawahar et al., 2019; Lin et al., 2019) and semantics (Ethayarajh, 2019; Wiedemann et al., 2019; Ettinger, 2020). There is also a recent study examining morphosyntactic information in a PLM, specifically BERT (Edmiston, 2020).

There has been relatively little recent work on derivational morphology in NLP. Both Cotterell et al. (2017) and Deutsch et al. (2018) propose neural architectures that represent derivational meanings as tags. More closely related to our study, Vylomova et al. (2017) develop an encoder-decoder model that uses the context sentence for predicting deverbal nouns. Hofmann et al. (2020b) propose a graph auto-encoder that models the morphological well-formedness of derivatives.

7 Conclusion

We show that a PLM, specifically BERT, can generate derivationally complex words. Our best model, DagoBERT, clearly beats an LSTM-based model, the previous state of the art in DG. DagoBERT's errors are mainly due to syntactic and semantic overlap between affixes. Furthermore, we demonstrate that the input segmentation impacts how much derivational knowledge is available to BERT. This suggests that the performance of PLMs could be further improved if a morphologically informed vocabulary of units were used.

Acknowledgements

Valentin Hofmann was funded by the Arts and Humanities Research Council and the German Academic Scholarship Foundation. This research was also supported by the European Research Council (Grant No. 740516). We thank the reviewers for their detailed and helpful comments.

References

Paolo Acquaviva. 2016. Morphological semantics. In Andrew Hippisley and Gregory Stump, editors, *The Cambridge handbook of morphology*, pages 117–148. Cambridge University Press, Cambridge.

Kenn Apel and Jessika Lawrence. 2011. Contributions of morphological awareness skills to word-level reading and spelling in first-grade children with and without speech sound disorder. *Journal of Speech, Language, and Hearing Research*, 54(5):1312–1327.

R. Harald Baayen and Rochelle Lieber. 1991. Productivity and English derivation: A corpus-based study. *Linguistics*, 29(5).

Laurie Bauer. 2001. *Morphological productivity*. Cambridge University Press, Cambridge, UK.

Laurie Bauer. 2019. *Rethinking morphology*. Edinburgh University Press, Edinburgh, UK.

Laurie Bauer, Rochelle Lieber, and Ingo Plag. 2013. *The Oxford reference guide to English morphology*. Oxford University Press, Oxford, UK.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *arXiv 2004.03720*.

Marc Brysbaert, Michaël Stevens, Paweł Mandera, and Emmanuel Keuleers. 2016. How many words do we know? practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age. *Frontiers in Psychology*, 7:1116.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017. Paradigm completion for derivational morphology. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017*.

David Crystal. 1997. *The Cambridge encyclopedia of the English language*. Cambridge University Press, Cambridge, UK.

Georgette Dal and Fiammetta Namer. 2016. Productivity. In Andrew Hippisley and Gregory Stump, editors, *The Cambridge handbook of morphology*, pages 70–89. Cambridge University Press, Cambridge.

Daniel Deutsch, John Hewitt, and Dan Roth. 2018. A distributional and orthographic aggregation model for English derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 17*.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2019*.

Daniel Edmiston. 2020. A systematic analysis of morphological content in BERT models for multiple languages. In *arXiv 2004.03032*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2019*.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Michelle Girvan and Mark Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. In *arXiv 1901.05287*.

Pius ten Hacken. 2014. Delineating derivation and inflection. In Rochelle Lieber and Pavol Štekauer, editors, *The Oxford handbook of derivational morphology*, pages 10–25. Oxford University Press, Oxford.

Martin Haspelmath and Andrea D. Sims. 2010. *Understanding morphology*. Routledge, New York, NY.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 17*.

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2020a. Predicting the growth of morphological families from social and linguistic factors. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Valentin Hofmann, Hinrich Schütze, and Janet B. Pierrehumbert. 2020b. A graph auto-encoder model of derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Ganesh Jawahar, Benoit Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Annual Meeting of the Association for Computational Linguistics (ACL) 57*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Diederik P. Kingma and Jimmy L. Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR) 3*.

Rochelle Lieber. 2019. Theoretical issues in word formation. In Jenny Audring and Francesca Masini, editors, *The Oxford handbook of morphological theory*, pages 34–55. Oxford University Press, Oxford.

Yongjie Lin, Yi C. Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP) 2*.

Taylor Mahler, Willy Cheung, Micha Elsner, David King, Marie-Catherine de Marneffe, Cory Shain, Symon Stevens-Guille, and Michael White. 2017. Breaking NLP: Using morphosyntax, semantics, pragmatics and world knowledge to fool sentiment analysis systems. In *Workshop on Building Linguistically Generalizable NLP Systems 1*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2014*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 16*.

Janet B. Pierrehumbert. 2006. The statistical basis of an unnatural alternation. In Louis Goldstein, Douglas H. Whalen, and Catherine T. Best, editors, *Laboratory Phonology 8*, pages 81–106. De Gruyter, Berlin.

Janet B. Pierrehumbert and Ramon Granell. 2018. On hapax legomena and morphological productivity. In *Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON) 15*.

Ingo Plag. 1999. *Morphological productivity: Structural constraints in English derivation*. De Gruyter, Berlin.

Ingo Plag. 2003. *Word-formation in English*. Cambridge University Press, Cambridge, UK.

Ingo Plag and Laura W. Balling. 2020. Derivational morphology: An integrative perspective on some fundamental questions. In Vito Pirrelli, Ingo Plag, and Wolfgang U. Dressler, editors, *Word knowledge and word usage: A cross-disciplinary guide to the mental lexicon*, pages 295–335. De Gruyter, Berlin.

Dragomir Radev, Hong Qi, Harris Wu, and Weiguo Fan. 2002. Evaluating web-based question answering systems. In *International Conference on Language Resources and Evaluation (LREC) 3*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. In *arXiv 2002.12327*.

Chenhao Tan and Lillian Lee. 2015. All who wander: On the prevalence and characteristics of multi-community engagement. In *International Conference on World Wide Web (WWW) 24*.

Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin, and Trevor Cohn. 2017. Context-aware prediction of derivational word-forms. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL) 15*.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT make any sense? interpretable word sense disambiguation with contextualized embeddings. In *arXiv 1909.10430*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le V, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv 1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS) 33*.

A Appendices

A.1 Data Preprocessing

We filter the posts for known bots and spammers (Tan and Lee, 2015). We exclude posts written in a language other than English and remove strings containing numbers, references to users, and hyperlinks. Sentences are filtered to contain between 10 and 100 words. We control that derivatives do not appear more than once in a sentence.

A.2 Hyperparameters

We tune hyperparameters on the development data separately for each frequency bin (selection criterion: MRR). Models are trained with categorical cross-entropy as the loss function and Adam (Kingma and Ba, 2015) as the optimizer. Training and testing are performed on a GeForce GTX 1080 Ti GPU (11GB).

DagoBERT. We use a batch size of 16 and perform grid search for the learning rate $l \in \{1 \times 10^{-6}, 3 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$ and the number of epochs $n_e \in \{1, \ldots, 8\}$ (number of hyperparameter search trials: 32). All other

hyperparameters are as for $\text{BERT}_{\text{BASE}}$. The number of trainable parameters is 110,104,890.

BERT+. We use a batch size of 16 and perform grid search for the learning rate $l \in \{1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}\}$ and the number of epochs $n_e \in \{1, \ldots, 8\}$ (number of hyperparameter search trials: 32). All other hyperparameters are as for $\text{BERT}_{\text{BASE}}$. The number of trainable parameters is 622,650.

LSTM. We initialize word embeddings with 300-dimensional GloVe (Pennington et al., 2014) vectors and character embeddings with 100-dimensional random vectors. The BiLSTMs consist of three layers and have a hidden size of 100. We use a batch size of 64 and perform grid search for the learning rate $l \in \{1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}\}$ and the number of epochs $n_e \in \{1, \ldots, 40\}$ (number of hyperparameter search trials: 160). The number of trainable parameters varies with the type of the model due to different sizes of the output layer and is 2,354,345 for P, 2,354,043 for S, and 2,542,038 for PS models.[11]

Table 10 lists statistics of the validation performance over hyperparameter search trials and provides information about the best validation performance as well as corresponding hyperparameter configurations.[12] We also report runtimes for the hyperparameter search.

For the models trained on the Vylomova et al. (2017) dataset, hyperparameter search is identical as for the main models, except that we use accuracy as the selection criterion. Runtimes for the hyperparameter search in minutes are 754 for SHARED and 756 for SPLIT in the case of DagoBERT, and 530 for SHARED and 526 for SPLIT in the case of LSTM. Best validation accuracy is .943 ($l = 3 \times 10^{-6}$, $n_e = 7$) for SHARED and .659 ($l = 1 \times 10^{-5}$, $n_e = 4$) for SPLIT in the case of DagoBERT, and .824 ($l = 1 \times 10^{-4}$, $n_e = 38$) for SHARED and .525 ($l = 1 \times 10^{-4}$, $n_e = 33$) for SPLIT in the case of LSTM.

A.3 Hyperparameters

We use the HYP segmentation method for models with morphologically correct segmentation. We tune hyperparameters on the development data separately for each frequency bin (selection criterion: accuracy). Models are trained with binary cross-entropy as the loss function and Adam as the optimizer. Training and testing are performed on a GeForce GTX 1080 Ti GPU (11GB).

For FROZEN, we use a batch size of 16 and perform grid search for the learning rate $l \in \{1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}\}$ and the number of epochs $n_e \in \{1, \ldots, 8\}$ (number of hyperparameter search trials: 32). The number of trainable parameters is 769. For FINETUNED, we use a batch size of 16 and perform grid search for the learning rate $l \in \{1 \times 10^{-6}, 3 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$ and the number of epochs $n_e \in \{1, \ldots, 8\}$ (number of hyperparameter search trials: 32). The number of trainable parameters is 109,483,009. All other hyperparameters are as for $\text{BERT}_{\text{BASE}}$.

Table 11 lists statistics of the validation performance over hyperparameter search trials and provides information about the best validation performance as well as corresponding hyperparameter configurations. We also report runtimes for the hyperparameter search.

[11] Since models are trained separately on the frequency bins, slight variations are possible if an affix does not appear in a particular bin. The reported numbers are for B1.

[12] Since expected validation performance (Dodge et al., 2019) may not be correct for grid search, we report mean and standard deviation of the performance instead.

Model			SHARED B1	B2	B3	B4	B5	B6	B7	SPLIT B1	B2	B3	B4	B5	B6	B7
DagoBERT	P	μ	.349	.400	.506	.645	.777	.871	.930	.345	.364	.375	.383	.359	.359	.357
		σ	.020	.037	.096	.160	.154	.112	.064	.018	.018	.018	.019	.018	.017	.022
		max	.372	.454	.657	.835	.896	.934	.957	.368	.385	.399	.412	.397	.405	.392
		l	1e-5	3e-5	3e-5	3e-5	1e-5	3e-6	3e-6	3e-6	1e-5	3e-6	1e-6	3e-6	1e-6	1e-6
		n_e	3	8	8	8	5	8	6	5	3	3	5	1	1	1
	S	μ	.386	.453	.553	.682	.805	.903	.953	.396	.403	.395	.395	.366	.390	.370
		σ	.031	.058	.120	.167	.164	.118	.065	.033	.024	.020	.020	.019	.029	.027
		max	.419	.535	.735	.872	.933	.965	.976	.429	.430	.420	.425	.403	.441	.432
		l	3e-5	3e-5	3e-5	3e-5	1e-5	1e-5	3e-6	3e-5	1e-5	3e-6	1e-6	1e-6	1e-6	1e-6
		n_e	2	7	8	6	8	7	6	2	3	5	7	3	2	1
	PS	μ	.124	.214	.362	.554	.725	.840	.926	.119	.158	.175	.194	.237	.192	.176
		σ	.018	.075	.173	.251	.238	.187	.119	.013	.013	.011	.016	.020	.021	.018
		max	.146	.337	.620	.830	.915	.945	.970	.135	.177	.192	.219	.269	.235	.209
		l	1e-5	3e-5	3e-5	3e-5	1e-5	3e-5	1e-5	1e-5	3e-6	3e-6	1e-6	1e-6	1e-6	1e-6
		n_e	6	8	8	5	8	3	7	6	8	3	4	4	1	1
		τ	192	230	314	440	631	897	1,098	195	228	313	438	631	897	791
BERT+	P	μ	.282	.336	.424	.527	.655	.764	.860	.280	.298	.318	.324	.323	.324	.322
		σ	.009	.020	.046	.078	.090	.080	.051	.011	.007	.009	.013	.009	.012	.009
		max	.297	.374	.497	.633	.759	.841	.901	.293	.312	.334	.345	.341	.357	.346
		l	1e-4	1e-3	3e-3	1e-3	3e-4	3e-4	3e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
		n_e	7	8	8	8	8	8	8	5	8	7	2	4	1	1
	S	μ	.358	.424	.491	.587	.708	.817	.886	.369	.364	.357	.350	.337	.335	.332
		σ	.010	.018	.043	.073	.086	.072	.049	.010	.010	.010	.013	.017	.017	.009
		max	.372	.452	.557	.691	.806	.884	.925	.383	.377	.375	.372	.366	.377	.357
		l	1e-4	1e-3	1e-3	1e-3	1e-3	1e-3	3e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
		n_e	4	7	8	8	7	7	8	8	4	5	1	1	1	1
	PS	μ	.084	.152	.257	.419	.598	.741	.849	.083	.104	.127	.137	.158	.139	.136
		σ	.008	.024	.062	.116	.119	.099	.062	.009	.014	.015	.014	.017	.011	.008
		max	.099	.206	.371	.610	.756	.847	.913	.099	.131	.154	.170	.206	.173	.164
		l	1e-4	3e-3	3e-3	3e-3	1e-3	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
		n_e	7	8	8	8	8	8	8	5	3	3	1	1	1	1
		τ	81	102	140	197	285	406	568	80	101	140	196	283	400	563
LSTM	P	μ	.103	.166	.314	.510	.661	.769	.841	.089	.113	.107	.106	.103	.103	.116
		σ	.031	.072	.163	.212	.203	.155	.107	.019	.024	.020	.017	.010	.010	.013
		max	.159	.331	.583	.732	.818	.864	.909	.134	.152	.141	.138	.121	.120	.139
		l	1e-3	1e-3	1e-3	1e-3	3e-4	1e-4	3e-4	3e-4	3e-4	3e-4	3e-4	1e-4	1e-4	3e-4
		n_e	33	40	38	35	35	40	26	38	36	37	38	40	37	29
	S	μ	.124	.209	.385	.573	.721	.824	.881	.108	.133	.136	.132	.132	.127	.128
		σ	.037	.098	.202	.229	.206	.162	.111	.029	.034	.027	.015	.013	.012	.012
		max	.214	.422	.674	.812	.882	.925	.945	.192	.187	.179	.157	.159	.157	.153
		l	3e-4	1e-3	1e-3	1e-3	3e-4	3e-4	3e-4	3e-4	3e-4	3e-4	3e-4	3e-4	1e-4	1e-4
		n_e	40	40	37	31	37	38	39	37	38	37	39	38	39	29
	PS	μ	.011	.066	.255	.481	.655	.776	.848	.009	.015	.025	.035	.046	.032	.071
		σ	.005	.090	.256	.301	.276	.220	.177	.003	.004	.006	.006	.008	.008	.015
		max	.022	.346	.649	.786	.844	.886	.931	.016	.024	.038	.047	.065	.055	.104
		l	3e-3	3e-3	3e-3	3e-4	3e-4	3e-4	3e-4	3e-4	3e-3	3e-4	3e-4	3e-3	3e-3	3e-4
		n_e	38	40	39	40	33	40	39	40	39	23	32	28	15	31
		τ	115	136	196	253	269	357	484	100	120	142	193	287	352	489

Table 10: Validation performance statistics and hyperparameter search details. The table shows the mean (μ), standard deviation (σ), and maximum ($\max$) of the validation performance (MRR) on all hyperparameter search trials for prefix (P), suffix (S), and prefix-suffix (PS) models. It also gives the learning rate (l) and number of epochs (n_e) with the best validation performance as well as the runtime (τ) in minutes averaged over P, S, and PS for one full hyperparameter search (32 trials for DagoBERT and BERT+, 160 trials for LSTM).

Segmentation		FROZEN							FINETUNED						
		B1	B2	B3	B4	B5	B6	B7	B1	B2	B3	B4	B5	B6	B7
Morphological	μ	.617	.639	.650	.660	.671	.684	.689	.732	.760	.764	.750	.720	.692	.657
	σ	.010	.009	.009	.008	.014	.009	.009	.016	.017	.029	.052	.067	.064	.066
	$\max$	.628	.649	.660	.669	.682	.692	.698	.750	.779	.793	.802	.803	.803	.808
	l	3e-4	3e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-5	3e-6	3e-6	1e-6	1e-6	1e-6	1e-6
	n_e	8	5	7	7	5	3	5	4	8	5	8	3	2	1
	τ	137	240	360	516	765	1,079	1,511	378	578	866	1,243	1,596	1,596	1,793
WordPiece	μ	.554	.561	.569	.579	.584	.592	.596	.706	.730	.734	.712	.669	.637	.604
	σ	.011	.010	.011	.012	.011	.010	.011	.030	.021	.025	.052	.066	.061	.046
	$\max$	.568	.574	.582	.592	.597	.605	.608	.731	.752	.762	.765	.763	.759	.759
	l	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-5	3e-6	3e-6	1e-6	1e-6	1e-6	1e-6
	n_e	6	8	6	2	8	2	7	3	7	6	8	3	1	1
	τ	139	242	362	517	765	1,076	1,507	379	575	869	1,240	1,597	1,598	1,775

Table 11: Validation performance statistics and hyperparameter search details. The table shows the mean (μ), standard deviation (σ), and maximum ($\max$) of the validation performance (accuracy) on all hyperparameter search trials for classifiers using morphological and WordPiece segmentations. It also gives the learning rate (l) and number of epochs (n_e) with the best validation performance as well as the runtime (τ) in minutes for one full hyperparameter search (32 trials for both morphological and WordPiece segmentations).

6
Superbizarre Is Not Superb

Contents

The study reported in chapter 5 shows that a morphologically invalid input tokenization can deteriorate the capabilities of a language model. If this is the case, should morphology not inform the design of tokenization algorithms in the first place? The fourth study examines this question by means of several semantic probing tasks, comparing the standard language model tokenization with the performance that is achieved when the morphological gold segmentation is used as the tokenization.

Superbizarre Is Not Superb: Derivational Morphology Improves BERT's Interpretation of Complex Words

Valentin Hofmann[*‡], **Janet B. Pierrehumbert**[†*], **Hinrich Schütze**[‡]

[*]Faculty of Linguistics, University of Oxford
[†]Department of Engineering Science, University of Oxford
[‡]Center for Information and Language Processing, LMU Munich
valentin.hofmann@ling-phil.ox.ac.uk

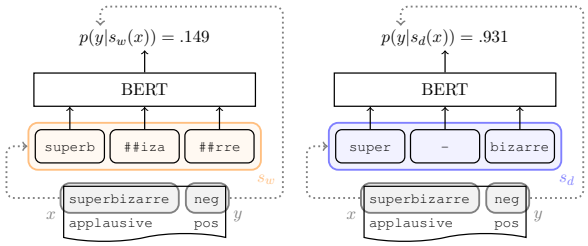
Abstract

How does the input segmentation of pretrained language models (PLMs) affect their interpretations of complex words? We present the first study investigating this question, taking BERT as the example PLM and focusing on its semantic representations of English derivatives. We show that PLMs can be interpreted as serial dual-route models, i.e., the meanings of complex words are either stored or else need to be computed from the subwords, which implies that maximally meaningful input tokens should allow for the best generalization on new words. This hypothesis is confirmed by a series of semantic probing tasks on which DelBERT (Derivation leveraging BERT), a model with derivational input segmentation, substantially outperforms BERT with WordPiece segmentation. Our results suggest that the generalization capabilities of PLMs could be further improved if a morphologically-informed vocabulary of input tokens were used.

1 Introduction

Pretrained language models (PLMs) such as BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), XLNet (Yang et al., 2019), ELECTRA (Clark et al., 2020), and T5 (Raffel et al., 2020) have yielded substantial improvements on a range of NLP tasks. What linguistic properties do they have? Various studies have tried to illuminate this question, with a focus on syntax (Hewitt and Manning, 2019; Jawahar et al., 2019) and semantics (Ethayarajh, 2019; Ettinger, 2020; Vulić et al., 2020).

One common characteristic of PLMs is their input segmentation: PLMs are based on fixed-size vocabularies of words and subwords that are generated by compression algorithms such as byte-pair encoding (Gage, 1994; Sennrich et al., 2016) and WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016). The segmentations produced by these



(a) BERT (s_w) (b) DelBERT (s_d)

Figure 1: Basic experimental setup. BERT with WordPiece segmentation (s_w) mixes part of the stem `bizarre` with the prefix `super`, creating an association with `superb` (left panel). DelBERT with derivational segmentation (s_d), on the other hand, separates prefix and stem by a hyphen (right panel). The two likelihoods are averaged across 20 models trained with different random seeds. The average likelihood of the true class is considerably higher with DelBERT than with BERT. While `superbizarre` has negative sentiment, `applausive` is an example of a complex word with positive sentiment.

algorithms are linguistically questionable at times (Church, 2020), which has been shown to worsen performance on certain downstream tasks (Bostrom and Durrett, 2020; Hofmann et al., 2020a). However, the wider implications of these findings, particularly with regard to the generalization capabilities of PLMs, are still poorly understood.

Here, we address a central aspect of this issue, namely how the input segmentation affects the semantic representations of PLMs, taking BERT as the example PLM. We focus on derivationally complex words such as `superbizarre` since they exhibit systematic patterns on the lexical level, providing an ideal testbed for linguistic generalization. At the same time, the fact that low-frequency and out-of-vocabulary words are often derivationally complex (Baayen and Lieber, 1991) makes our work relevant in practical settings, especially when many one-word expressions are involved, e.g., in query processing (Kacprzak et al., 2017).

The topic of this paper is related to the more fundamental question of how PLMs represent the meaning of complex words in the first place. So far, most studies have focused on methods of representation extraction, using ad-hoc heuristics such as averaging the subword embeddings (Pinter et al., 2020; Sia et al., 2020; Vulić et al., 2020) or taking the first subword embedding (Devlin et al., 2019; Heinzerling and Strube, 2019; Martin et al., 2020). While not resolving the issue, we lay the theoretical groundwork for more systematic analyses by showing that PLMs can be regarded as serial dual-route models (Caramazza et al., 1988), i.e., the meanings of complex words are either stored or else need to be computed from the subwords.

Contributions. We present the first study examining how the input segmentation of PLMs, specifically BERT, affects their interpretations of derivationally complex English words. We show that PLMs can be interpreted as serial dual-route models, which implies that maximally meaningful input tokens should allow for the best generalization on new words. This hypothesis is confirmed by a series of semantic probing tasks on which derivational segmentation substantially outperforms BERT's WordPiece segmentation. This suggests that the generalization capabilities of PLMs could be further improved if a morphologically-informed vocabulary of input tokens were used. We also publish three large datasets of derivationally complex words with corresponding semantic properties.[1]

2 How Are Complex Words Processed?

2.1 Complex Words in Psycholinguistics

The question of how complex words are processed has been at the center of psycholinguistic research over the last decades (see Leminen et al. (2019) for a recent review). Two basic processing mechanisms have been proposed: *storage*, where the meaning of complex words is listed in the mental lexicon (Manelis and Tharp, 1977; Butterworth, 1983; Feldman and Fowler, 1987; Bybee, 1988; Stemberger, 1994; Bybee, 1995; Bertram et al., 2000a), and *computation*, where the meaning of complex words is inferred based on the meaning of stem and affixes (Taft and Forster, 1975; Taft, 1979, 1981, 1988, 1991, 1994; Rastle et al., 2004; Taft, 2004; Rastle and Davis, 2008).

[1]We make our code and data available at https://github.com/valentinhofmann/superbizarre.

In contrasting with *single-route* frameworks, *dual-route* models allow for a combination of storage and computation. Dual-route models are further classified by whether they regard the processes of retrieving meaning from the mental lexicon and computing meaning based on stem and affixes as *parallel*, i.e., both mechanisms are always activated (Frauenfelder and Schreuder, 1992; Schreuder and Baayen, 1995; Baayen et al., 1997, 2000; Bertram et al., 2000b; New et al., 2004; Kuperman et al., 2008, 2009), or *serial*, i.e., the computation-based mechanism is only activated when the storage-based one fails (Laudanna and Burani, 1985; Burani and Caramazza, 1987; Caramazza et al., 1988; Burani and Laudanna, 1992; Laudanna and Burani, 1995; Alegre and Gordon, 1999).

Outside the taxonomy presented so far are recent models that assume multiple levels of representation as well as various forms of interaction between them (Rácz et al., 2015; Needle and Pierrehumbert, 2018). In these models, sufficiently frequent complex words are stored together with representations that include their internal structure. Complex-word processing is driven by analogical processes over the mental lexicon (Rácz et al., 2020).

2.2 Complex Words in NLP and PLMs

Most models of word meaning proposed in NLP can be roughly assigned to either the single-route or dual-route approach. Word embeddings that represent complex words as whole-word vectors (Deerwester et al., 1990; Mikolov et al., 2013a,b; Pennington et al., 2014) can be seen as single-route storage models. Word embeddings that represent complex words as a function of subword or morpheme vectors (Schütze, 1992; Luong et al., 2013) can be seen as single-route computation models. Finally, word embeddings that represent complex words as a function of subword or morpheme vectors as well as whole-word vectors (Botha and Blunsom, 2014; Qiu et al., 2014; Bhatia et al., 2016; Bojanowski et al., 2017; Athiwaratkun et al., 2018; Salle and Villavicencio, 2018) are most closely related to parallel dual-route approaches.

Where are PLMs to be located in this taxonomy? PLMs represent many complex words as whole-word vectors (which are fully stored). Similarly to how character-based models represent word meaning (Kim et al., 2016; Adel et al., 2017), they can also store the meaning of frequent complex words that are segmented into subwords, i.e., frequent sub-

word collocations, in their model weights. When the complex-word meaning is neither stored as a whole-word vector nor in the model weights, PLMs compute the meaning as a compositional function of the subwords. Conceptually, PLMs can thus be interpreted as serial dual-route models. While the parallelism has not been observed before, it follows logically from the structure of PLMs. The key goal of this paper is to show that the implications of this observation are borne out empirically.

As a concrete example, consider the complex words `stabilize`, `realize`, `finalize`, `mobilize`, `tribalize`, and `templatize`, which are all formed by adding the verbal suffix `ize` to a nominal or adjectival stem. Taking BERT, specifically $\text{BERT}_{\text{BASE}}$ (uncased) (Devlin et al., 2019), as the example PLM, the words `stabilize` and `realize` have individual tokens in the input vocabulary and are hence associated with whole-word vectors storing their meanings, including highly lexicalized meanings as in the case of `realize`. By contrast, the words `finalize` and `mobilize` are segmented into `final`, `##ize` and `mob`, `##ili`, `##ze`, which entails that their meanings are not stored as whole-word vectors. However, both words have relatively high absolute frequencies of 2,540 (`finalize`) and 6,904 (`mobilize`) in the English Wikipedia, the main dataset used to pretrain BERT (Devlin et al., 2019), which means that BERT can store their meanings in its model weights during pretraining.[2] Notice this is even possible in the case of highly lexicalized meanings as for `mobilize`. Finally, the words `tribalize` and `templatize` are segmented into `tribal`, `##ize` and `te`, `##mp`, `##lat`, `##ize`, but as opposed to `finalize` and `mobilize` they do not occur in the English Wikipedia. As a result, BERT cannot store their meanings in its model weights during pretraining and needs to compute them from the meanings of the subwords.

Seeing PLMs as serial dual-route models allows for a more nuanced view on the central research question of this paper: in order to investigate semantic generalization we need to investigate the representations of those complex words that activate the computation-based route. The words that do so are the ones whose meaning is neither stored as a whole-word vector nor in the model weights and hence needs to be computed compositionally as a function of the subwords (`tribalize` and `templatize` in the discussed examples). We hypothesize that the morphological validity of the segmentation affects the representational quality in these cases, and that the best generalization is achieved by maximally meaningful tokens. It is crucial to note this does not imply that the tokens have to be morphemes, but the segmentation boundaries need to coincide with morphological boundaries, i.e., groups of morphemes (e.g., `tribal` in the segmentation of `tribalize`) are also possible.[3] For `tribalize` and `templatize`, we therefore expect the segmentation `tribal`, `##ize` (morphologically valid since all segmentation boundaries are morpheme boundaries) to result in a representation of higher quality than the segmentation `te`, `##mp`, `##lat`, `##ize` (morphologically invalid since the boundaries between `te`, `##mp`, and `##lat` are not morpheme boundaries). On the other hand, complex words whose meanings are stored in the model weights (`finalize` and `mobilize` in the discussed examples) are expected to be affected by the segmentation to a much lesser extent: if the meaning of a complex word is stored in the model weights, it should matter less whether the specific segmentation activating that meaning is morphologically valid (`final`, `##ize`) or not (`mob`, `##ili`, `##ze`).[4]

3 Experiments

3.1 Setup

Analyzing the impact of different segmentations on BERT's semantic generalization capabilities is not straightforward since it is not clear a priori how to measure the quality of representations. Here, we devise a novel lexical-semantic probing task: we use BERT's representations for complex words to predict semantic dimensions, specifically sentiment and topicality (see Figure 1). For sentiment, given the example complex word `superbizarre`, the task is to predict that its sentiment is negative. For topicality, given the example complex word `isotopize`, the task is to predict that it is used in physics. We confine ourselves to binary predic-

[2] Previous research suggests that such lexical knowledge is stored in the lower layers of BERT (Vulić et al., 2020).

[3] This is in line with substantial evidence from linguistics showing that frequent groups of morphemes can be treated as semantic wholes (Stump, 2017, 2019).

[4] We expect the distinction between storage and computation of complex-word meaning for PLMs to be a continuum. While the findings presented here are consistent with this view, we defer a more in-depth analysis to future work.

Dataset	Dimension	$\lvert\mathcal{D}\rvert$	Class 1		Class 2	
			Class	Examples	Class	Example
Amazon	Sentiment	239,727	neg	overpriced, crappy	pos	megafavorite, applausive
ArXiv	Topicality	97,410	phys	semithermal, ozoneless	cs	autoencoded, rankable
Reddit	Topicality	85,362	ent	supervampires, spoilerful	dis	antirussian, immigrationism

Table 1: Dataset characteristics. The table provides information about the datasets such as the relevant semantic dimensions with their classes and example complex words. $\lvert\mathcal{D}\rvert$: number of complex words; neg: negative; pos: positive; phys: physics; cs: computer science; ent: entertainment; dis: discussion.

tion, i.e., the probed semantic dimensions always consist of two classes (e.g., positive and negative). The extent to which a segmentation supports a solution of this task is taken as an indicator of its representational quality.

More formally, let $\mathcal{D}$ be a dataset consisting of complex words x and corresponding classes y that instantiate a certain semantic dimension (e.g., sentiment). We denote with $s(x) = (t_1, \ldots, t_k)$ the segmentation of x into a sequence of k subwords. We ask how s impacts the capability of BERT to predict y, i.e., how $p(y|(s(x))$, the likelihood of the true semantic class y given a certain segmentation of x, depends on different choices for s. The two segmentation methods we compare in this study are BERT's standard WordPiece segmentation (Schuster and Nakajima, 2012; Wu et al., 2016), s_w, and a derivational segmentation that segments complex words into stems and affixes, s_d.

3.2 Data

Since existing datasets do not allow us to conduct experiments following the described setup, we create new datasets in a weakly-supervised fashion that is conceptually similar to the method proposed by Mintz et al. (2009): we employ large datasets annotated for sentiment or topicality, extract derivationally complex words, and use the dataset labels to establish their semantic classes.

For determining and segmenting derivationally complex words, we use the algorithm introduced by Hofmann et al. (2020b), which takes as input a set of prefixes, suffixes, and stems and checks for each word in the data whether it can be derived from a stem using a combination of prefixes and suffixes.[5] The algorithm is sensitive to morpho-orthographic rules of English (Plag, 2003), e.g., when the suffix ize is removed from isotopize, the result is isotope, not isotop. We follow Hofmann et al. (2020a) in using the prefixes, suffixes, and stems in BERT's WordPiece vocabulary as input to the algorithm. This means that all tokens used by the derivational segmentation are in principle also available to the WordPiece segmentation, i.e., the difference between s_w and s_d does not lie in the vocabulary per se but rather in the way the vocabulary is used. See Appendix A.1 for details about the derivational segmentation.

To get the semantic classes, we compute for each complex word which fraction of texts containing the word belongs to one of two predefined sets of dataset labels (e.g., reviews with four and five stars for positive sentiment) and rank all words accordingly. We then take the first and third tertiles of complex words as representing the two classes. We randomly split the words into 60% training, 20% development, and 20% test.

In the following, we describe the characteristics of the three datasets in greater depth. Table 1 provides summary statistics. See Appendix A.2 for details about data preprocessing.

Amazon. Amazon is an online e-commerce platform. A large dataset of Amazon reviews has been made publicly available (Ni et al., 2019).[6] We extract derivationally complex words from reviews with one or two (neg) as well as four or five stars (pos), discarding three-star reviews for a clearer separation (Yang and Eisenstein, 2017).

ArXiv. ArXiv is an open-access distribution service for scientific articles. Recently, a dataset of all papers published on ArXiv with associated metadata has been released.[7] For this study, we extract all articles from physics (phys) and computer science (cs), which we identify using ArXiv's subject classification. We choose physics and computer

[5] The distinction between inflectionally and derivationally complex words is notoriously fuzzy (Haspelmath and Sims, 2010; ten Hacken, 2014). We try to exclude inflection as far as possible (e.g., by removing problematic affixes such as ing) but are aware that a clear separation does not exist.

[6] https://nijianmo.github.io/amazon/index.html

[7] https://www.kaggle.com/Cornell-University/arxiv

	Amazon		ArXiv		Reddit	
Model	Dev	Test	Dev	Test	Dev	Test
DelBERT	.635 ± .001	.639 ± .002	.731 ± .001	.723 ± .001	.696 ± .001	.701 ± .001
BERT	.619 ± .001	.624 ± .001	.704 ± .001	.700 ± .002	.664 ± .001	.664 ± .003
Stem	.572 ± .003	.573 ± .003	.705 ± .001	.697 ± .001	.679 ± .001	.684 ± .002
Affixes	.536 ± .008	.539 ± .008	.605 ± .001	.603 ± .002	.596 ± .001	.596 ± .001

Table 2: Results. The table shows the average performance as well as standard deviation (F1) of 20 models trained with different random seeds. Best result per column highlighted in gray, second-best in light gray.

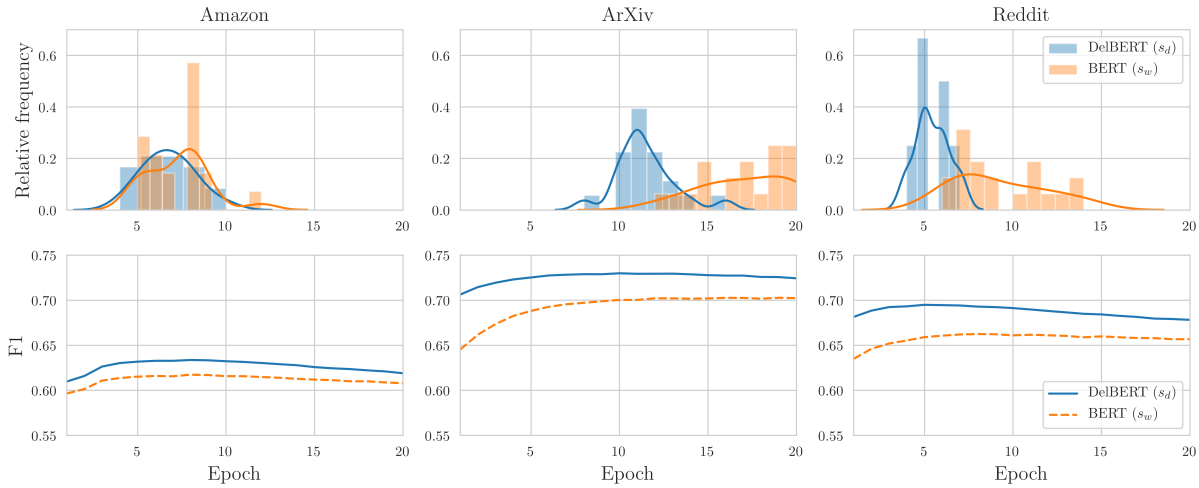


Figure 2: Convergence analysis. The upper panels show the distributions of the number of epochs after which the models reach their maximum validation performance. The lower panels show the trajectories of the average validation performance (F1) across epochs. The plots are based on 20 models trained with different random seeds. The convergence statistics for DelBERT and BERT are directly comparable because the optimal learning rate is the same (see Appendix A.3). DelBERT models reach their performance peak faster than BERT models.

science since we expect large topical distances for these classes (compared to alternatives such as mathematics and computer science).

Reddit. Reddit is a social media platform hosting discussions about various topics. It is divided into smaller communities, so-called subreddits, which have been shown to be a rich source of derivationally complex words (Hofmann et al., 2020c). Hofmann et al. (2020a) have published a dataset of derivatives found on Reddit annotated with the subreddits in which they occur.[8] Inspired by a content-based subreddit categorization scheme,[9] we define two groups of subreddits, an entertainment set (`ent`) consisting of the subreddits `anime`, `DestinyTheGame`, `funny`, `Games`, `gaming`, `leagueoflegends`, `movies`, `Music`, `pics`, and `videos`, as well as a discussion set (`dis`) consisting of the subreddits `askscience`, `atheism`, `conspiracy`, `news`, `Libertarian`, `politics`, `science`, `technology`, `TwoXChromosomes`, and `worldnews`, and extract all derivationally complex words occurring in them. We again expect large topical distances for these classes.

Given that the automatic creation of the datasets necessarily introduces noise, we measure human performance on 100 randomly sampled words per dataset, which ranges between 71% (Amazon) and 78% (ArXiv). These values can thus be seen as an upper bound on performance.

3.3 Models

We train two main models on each binary classification task: BERT with the standard WordPiece segmentation (s_w) and BERT using the derivational segmentation (s_d), a model that we refer to as DelBERT (**De**rivation **l**everaging **BERT**). BERT and DelBERT are identical except for the way in which they use the vocabulary of input tokens (but the vocabulary itself is also identical for both models).

[8] https://github.com/valentinhofmann/dagobert

[9] https://www.reddit.com/r/TheoryOfReddit/comments/1f7hqc/the_200_most_active_subreddits_categorized_by

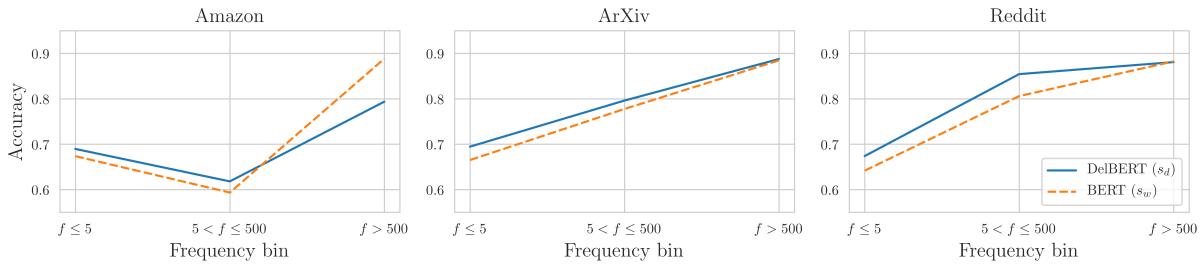


Figure 3: Frequency analysis. The plots show the average performance (accuracy) of 20 BERT and DelBERT models trained with different random seeds for complex words of low ($f \leq 5$), mid ($5 < f \leq 500$), and high ($f > 500$) frequency. On all three datasets, BERT performs similarly or better than DelBERT for complex words of high frequency but worse for complex words of low and mid frequency.

The specific BERT variant we use is $\text{BERT}_{\text{BASE}}$ (uncased) (Devlin et al., 2019). For the derivational segmentation, we follow previous work by Hofmann et al. (2020a) in separating stem and prefixes by a hyphen. We further follow Casanueva et al. (2020) and Vulić et al. (2020) in mean-pooling the output representations for all subwords, excluding BERT's special tokens. The mean-pooled representation is then fed into a two-layer feed-forward network for classification. To examine the relative importance of different types of morphological units, we train two additional models in which we ablate information about stems and affixes, i.e., we represent stems and affixes by the same randomly chosen input embedding.[10]

We finetune BERT, DelBERT, and the two ablated models on the three datasets using 20 different random seeds. We choose F1 as the evaluation measure. See Appendix A.3 for details about implementation and hyperparameters.

3.4 Results

DelBERT (s_d) outperforms BERT (s_w) by a large margin on all three datasets (Table 2). It is interesting to notice that the performance difference is larger for ArXiv and Reddit than for Amazon, indicating that the gains in representational quality are particularly large for topicality.

What is it that leads to DelBERT's increased performance? The ablation study shows that models using only stem information already achieve relatively high performance and are on par or even better than the BERT models on ArXiv and Reddit. However, the DelBERT models still perform substantially better than the stem models on all three datasets. The gap is particularly pronounced for Amazon, which indicates that the interaction between the meaning of stem and affixes is more complex for sentiment than for topicality. This makes sense from a linguistic point of view: while stems tend to be good cues for the topical associations of a complex word, sentiment often depends on semantic interactions between stems and affixes. For example, while the prefix `un` turns the sentiment of `amusing` negative, it turns the sentiment of `biased` positive. Such effects involving negation and antonymy are known to be challenging for PLMs (Ettinger, 2020; Kassner and Schütze, 2020) and might be one of the reasons for the generally lower performance on Amazon.[11] The performance of models using only affixes is much lower.

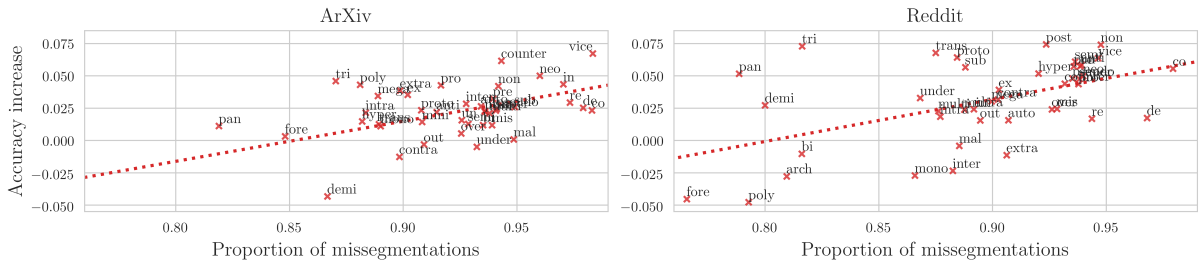
3.5 Quantitative Analysis

To further examine how BERT (s_w) and DelBERT (s_d) differ in the way they infer the meaning of complex words, we perform a convergence analysis. We find that the DelBERT models reach their peak in performance faster than the BERT models (Figure 2). This is in line with our interpretation of PLMs as serial dual-route models (see Section 2.2): while DelBERT operates on morphological units and can combine the subword meanings to infer the meanings of complex words, BERT's subwords do not necessarily carry lexical meanings, and hence the derivational patterns need to be stored by adapting the model weights. This is an additional burden, leading to longer convergence times and substantially worse overall performance.

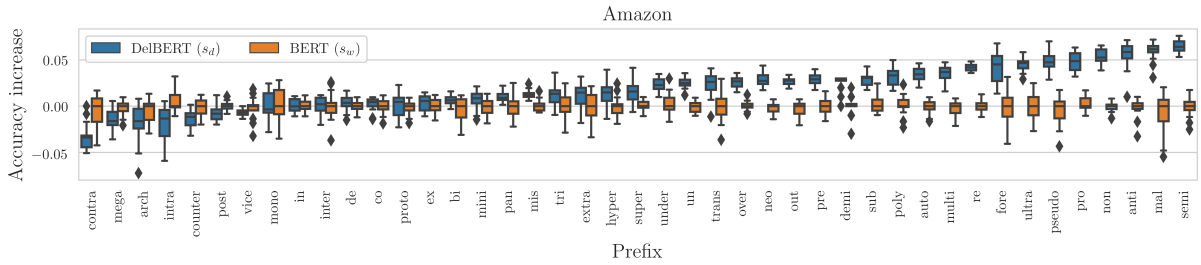
Our hypothesis that PLMs can use two routes

[10] For affix ablation, we use two different input embeddings for prefixes and suffixes.

[11] Another reason for the lower performance on sentiment is that the datasets were created automatically (see Section 3.2), and hence many complex words do not directly carry information about sentiment or topicality. The density of such words is higher for sentiment than topicality since the topic of discussion affects the likelihoods of most content words.



(a) Topicality prediction



(b) Sentiment prediction

Figure 4: Accuracy increase of DelBERT compared to BERT for prefixes. The plots show the accuracy increase as a function of the proportion of morphologically incorrect WordPiece segmentations (topicality prediction) and as ordered boxplot pairs centered on the median accuracy of BERT (sentiment prediction). Negative values mean that the DelBERT models have a lower accuracy than the BERT models for a certain prefix.

to process complex words (storage in weights and compositional computation based on input embeddings), and that the second route is blocked when the input segmentation is not morphological, suggests the existence of frequency effects: BERT might have seen frequent complex words multiple times during pretraining and stored their meaning in the model weights. This is less likely for infrequent complex words, making the capability to compositionally infer the meaning (i.e., the computation route) more important. We therefore expect the difference in performance between DelBERT (which should have an advantage on the computation route) and BERT to be larger for infrequent words. To test this hypothesis, we split the complex words of each dataset into three bins of low ($f \leq 5$), mid ($5 < f \leq 500$), and high ($f > 500$) absolute frequencies, and analyze how the performance of BERT and DelBERT differs on the three bins. For this and all subsequent analyses, we merge development and test sets and use accuracy instead of F1 since it makes comparisons across small sets of data points more interpretable. The results are in line with our hypothesis (Figure 3): BERT performs worse than DelBERT on complex words of low and mid frequencies but achieves very similar (ArXiv, Reddit) or even better (Amazon) accuracies on high-frequency complex words. These results strongly suggest that two different mechanisms are involved, and that BERT has a disadvantage for complex words that do not have a high frequency. At the same time, the slight advantage of BERT on high-frequency complex words indicates that it has high-quality representations of these words in its weights, which DelBERT cannot exploit since it uses a different segmentation.

We are further interested to see whether the affix type has an impact on the relative performance of BERT and DelBERT. To examine this question, we measure the accuracy increase of DelBERT as compared to BERT for individual affixes, averaged across datasets and random seeds. We find that the increase is almost twice as large for prefixes ($\mu = .023$, $\sigma = .017$) than for suffixes ($\mu = .013$, $\sigma = .016$), a difference that is shown to be significant by a two-tailed Welch's t-test ($d = .642$, $t(82.97) = 2.94$, $p < .01$).[12] Why is having access to the correct morphological segmentation more advantageous for prefixed than suffixed complex words? We argue that there are two key factors at play. First, the WordPiece tokenization sometimes generates the morphologically correct segmenta-

[12] We use a Welch's instead of Student's t-test since it does not assume that the distributions have equal variance.

Dataset	x	y	$s_d(x)$	μ_p	$s_w(x)$	μ_p
Amazon	applausive	pos	applause, ##ive	.847	app, ##laus, ##ive	.029
	superannoying	neg	super, -, annoying	.967	super, ##ann, ##oy, ##ing	.278
	overseasoned	neg	over, -, seasoned	.956	overseas, ##oned	.219
ArXiv	isotopize	phy	isotope, ##ize	.985	iso, ##top, ##ize	.039
	antimicrosoft	cs	anti, -, microsoft	.936	anti, ##mic, ##ros, ##oft	.013
	inkinetic	phy	in, -, kinetic	.983	ink, ##ine, ##tic	.035
Reddit	prematuration	dis	premature, ##ation	.848	prem, ##at, ##uration	.089
	nonmultiplayer	ent	non, -, multiplayer	.950	non, ##mu, ##lt, ##ip, ##layer	.216
	promosque	dis	pro, -, mosque	.961	promo, ##sque	.066

Table 3: Error analysis. The table gives example complex words that are consistently classified correctly by DelBERT and incorrectly by BERT. x: complex word; y: semantic class; $s_d(x)$: derivational segmentation; μ_p: average likelihood of true semantic class across 20 models trained with different random seeds; $s_w(x)$: WordPiece segmentation. For the complex words shown, μ_p is considerably higher with DelBERT than with BERT.

tion, but it does so with different frequencies for prefixes and suffixes. To detect morphologically incorrect segmentations, we check whether the WordPiece segmentation keeps the stem intact, which is in line with our definition of morphological validity (Section 2.2) and provides a conservative estimate of the error rate. For prefixes, the WordPiece tokenization is seldom correct (average error rate: $\mu = .903$, $\sigma = .042$), whereas for suffixes it is correct about half the time ($\mu = .503$, $\sigma = .213$). Hence, DelBERT gains a greater advantage for prefixed words. Second, prefixes and suffixes have different linguistic properties that affect the prediction task in unequal ways. Specifically, whereas suffixes have both syntactic and semantic functions, prefixes have an exclusively semantic function and always add lexical-semantic meaning to the stem (Giraudo and Grainger, 2003; Beyersmann et al., 2015). As a result, cases such as `unamusing` where the affix boundary is a decisive factor for the prediction task are more likely to occur with prefixes than suffixes, thus increasing the importance of a morphologically correct segmentation.[13]

Given the differences between sentiment and topicality prediction, we expect variations in the relative importance of the two identified factors: (i) in the case of sentiment the advantage of s_d should be maximal for affixes directly affecting sentiment; (ii) in the case of topicality its advantage should be the larger the higher the proportion of incorrect segmentations for a particular affix, and hence the more frequent the cases where DelBERT has access to the stem while BERT does not. To test this hypothesis, we focus on predictions for prefixed complex words. For each dataset, we measure for individual prefixes the accuracy increase of the DelBERT models as compared to the BERT models, averaged across random seeds, as well as the proportion of morphologically incorrect segmentations produced by WordPiece. We then calculate linear regressions to predict the accuracy increases based on the proportions of incorrect segmentations. This analysis shows a significant positive correlation for ArXiv ($R^2 = .304$, $F(1, 41) = 17.92$, $p < 0.001$) and Reddit ($R^2 = .270$, $F(1, 40) = 14.80$, $p < 0.001$) but not for Amazon ($R^2 = .019$, $F(1, 41) = .80$, $p = .375$), which is in line with our expectations (Figure 4a). Furthermore, ranking the prefixes by accuracy increase for Amazon confirms that the most pronounced differences are found for prefixes that can change the sentiment such as `non`, `anti`, `mal`, and `pseudo` (Figure 4b).

3.6 Qualitative Analysis

Besides quantitative factors, we are interested in identifying qualitative contexts in which DelBERT has a particular advantage compared to BERT. To do so, we filter the datasets for complex words that are consistently classified correctly by DelBERT and incorrectly by BERT. Specifically, we compute for each word the average likelihood of the true semantic class across DelBERT and BERT models, respectively, and rank words according to the likelihood difference between both model types. Examining the words with the most extreme differences, we observe three classes (Table 3).

First, the addition of a suffix is often connected with morpho-orthographic changes (e.g., the deletion of a stem-final `e`), which leads to a segmentation of the stem into several subwords

[13] Notice that there are suffixes with similar semantic effects (e.g., `less`), but they are less numerous.

since the truncated stem is not in the WordPiece vocabulary (`applausive`, `isotopize`, `prematuration`). The model does not seem to be able to recover the meaning of the stem from the subwords. Second, the addition of a prefix has the effect that the word-internal (as opposed to word-initial) form of the stem would have to be available for proper segmentation. Since this form rarely exists in the WordPiece vocabulary, the stem is segmented into several subwords (`superannoying`, `antimicrosoft`, `nonmultiplayer`). Again, it does not seem to be possible for the model to recover the meaning of the stem. Third, the segmentation of prefixed complex words often fuses the prefix with the first characters of the stem (`overseasoned`, `inkinetic`, `promosque`). This case is particularly detrimental since it not only makes it difficult to recover the meaning of the stem but also creates associations with unrelated meanings, sometimes even opposite meanings as in the case of `superbizarre`. The three classes thus underscore the difficulty of inferring the meaning of complex words from the subwords when the whole-word meaning is not stored in the model weights and the subwords are not morphological.

4 Related Work

Several recent studies have examined how the performance of PLMs is affected by their input segmentation. Tan et al. (2020) show that tokenizing inflected words into stems and inflection symbols allows BERT to generalize better on non-standard inflections. Bostrom and Durrett (2020) pretrain RoBERTa with different tokenization methods and find tokenizations that align more closely with morphology to perform better on a number of tasks. Ma et al. (2020) show that providing BERT with character-level information also leads to enhanced performance. Relatedly, studies from automatic speech recognition have demonstrated that morphological decomposition improves the perplexity of language models (Fang et al., 2015; Jain et al., 2020). Whereas these studies change the vocabulary of input tokens (e.g., by adding special tokens), we show that even when keeping the pretrained vocabulary fixed, employing it in a morphologically correct way leads to better performance.[14]

[14]There are also studies that analyze morphological aspects of PLMs without a focus on questions surrounding segmentation (Edmiston, 2020; Klemen et al., 2020).

Most NLP studies on derivational morphology have been devoted to the question of how semantic representations of derivationally complex words can be enhanced by including morphological information (Luong et al., 2013; Botha and Blunsom, 2014; Qiu et al., 2014; Bhatia et al., 2016; Cotterell and Schütze, 2018), and how affix embeddings can be computed (Lazaridou et al., 2013; Kisselew et al., 2015; Padó et al., 2016). Cotterell et al. (2017), Vylomova et al. (2017), and Deutsch et al. (2018) propose sequence-to-sequence models for the generation of derivationally complex words. Hofmann et al. (2020a) address the same task using BERT. In contrast, we analyze how different input segmentations affect the semantic representations of derivationally complex words in PLMs, a question that has not been addressed before.

5 Conclusion

We have examined how the input segmentation of PLMs, specifically BERT, affects their interpretations of derivationally complex words. Drawing upon insights from psycholinguistics, we have deduced a conceptual interpretation of PLMs as serial dual-route models, which implies that maximally meaningful input tokens should allow for the best generalization on new words. This hypothesis was confirmed by a series of semantic probing tasks on which DelBERT, a model using derivational segmentation, consistently outperformed BERT using WordPiece segmentation. Quantitative and qualitative analyses further showed that BERT's inferior performance was caused by its inability to infer the complex-word meaning as a function of the subwords when the complex-word meaning was not stored in the weights. Overall, our findings suggest that the generalization capabilities of PLMs could be further improved if a morphologically-informed vocabulary of input tokens were used.

Acknowledgements

This work was funded by the European Research Council (#740516) and the Engineering and Physical Sciences Research Council (EP/T023333/1). The first author was also supported by the German Academic Scholarship Foundation and the Arts and Humanities Research Council. We thank the reviewers for their helpful comments.

References

Heike Adel, Ehsaneddin Asgari, and Hinrich Schütze. 2017. Overview of character-based models for natural language processing. In *International Conference on Computational Linguistics and Intelligent Text Processing (CICLing) 18*.

Maria Alegre and Peter Gordon. 1999. Frequency effects and the representational status of regular inflections. *Journal of Memory and Language*, 40:41–61.

Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. 2018. Probabilistic fasttext for multi-sense word embeddings. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*.

R. Harald Baayen, Ton Dijkstra, and Robert Schreuder. 1997. Singulars and plurals in Dutch: Evidence for a parallel dual-route model. *Journal of Memory and Language*, 37:94–117.

R. Harald Baayen and Rochelle Lieber. 1991. Productivity and English derivation: A corpus-based study. *Linguistics*, 29(5).

R. Harald Baayen, Robert Schreuder, and Richard Sproat. 2000. Morphology in the mental lexicon: A computational model for visual word recognition. In Frank van Eynde and Dafydd Gibbon, editors, *Lexicon development for speech and language processing*, pages 267–293. Springer, Dordrecht.

Raymond Bertram, Matti Laine, R. Harald Baayen, Robert Schreuder, and Jukka Hyönä. 2000a. Affixal homonymy triggers full-form storage, even with inflected words, even in a morphologically rich language. *Cognition*, 74:B13–B25.

Raymond Bertram, Robert Schreuder, and R. Harald Baayen. 2000b. The balance of storage and computation in morphological processing: The role of word formation type, affixal homonymy, and productivity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(2):489–511.

Elisabeth Beyersmann, Johannes C. Ziegler, and Jonathan Grainger. 2015. Differences in the processing of prefixes and suffixes revealed by a letter-search task. *Scientific Studies of Reading*, 19(5):360–373.

Parminder Bhatia, Robert Guthrie, and Jacob Eisenstein. 2016. Morphological priors for probabilistic neural word embeddings. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2016*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Jan A. Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *International Conference on Machine Learning (ICML) 31*.

Cristina Burani and Alfonso Caramazza. 1987. Representation and processing of derived words. *Language and Cognitive Processes*, 2(3-4):217–227.

Cristina Burani and Alessandro Laudanna. 1992. Units of representation for derived words in the lexicon. In Ram Frost and Leonard Katz, editors, *Orthography, phonology, morphology, and meaning*, pages 361–376. North-Holland, Amsterdam.

Brian Butterworth. 1983. Lexical representation. In Brian Butterworth, editor, *Language production: Development, writing and other language processes*, pages 257–294. Academic Press, London.

Joan Bybee. 1988. Morphology as lexical organization. In Michael Hammond and Michael Noonan, editors, *Theoretical approaches to morphology: Approaches in modern linguistics*, pages 119–141. Academic Press, San Diego, CA.

Joan Bybee. 1995. Regular morphology and the lexicon. *Language and Cognitive Processes*, 10(425-455).

Alfonso Caramazza, Alessandro Laudanna, and Cristina Romani. 1988. Lexical access and inflectional morphology. *Cognition*, 28(297-332).

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Workshop on Natural Language Processing for Conversational AI 2*.

Kenneth Church. 2020. Emerging trends: Subwords, seriously? *Natural Language Engineering*, 26(3):375–382.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pretraining text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR) 8*.

Ryan Cotterell and Hinrich Schütze. 2018. Joint semantic synthesis and morphological analysis of the derived word. *Transactions of the Association for Computational Linguistics*, 6:33–48.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017. Paradigm completion for derivational morphology. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017*.

Scott Deerwester, Susan T. Dumais, George Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Daniel Deutsch, John Hewitt, and Dan Roth. 2018. A distributional and orthographic aggregation model for English derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2019*.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2019*.

Daniel Edmiston. 2020. A systematic analysis of morphological content in BERT models for multiple languages. In *arXiv 2004.03032*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2019*.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Hao Fang, Mari Ostendorf, Peter Baumann, and Janet B. Pierrehumbert. 2015. Exponential language modeling using morphological features and multitask learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2410–2421.

Laurie B. Feldman and Carol A. Fowler. 1987. The inflected noun system in Serbo-Croatian: Lexical representation of morphological structure. *Memory and Cognition*, 15(1):1–12.

Uli H. Frauenfelder and Robert Schreuder. 1992. Constraining psycholinguistic models of morphological processing and representation: The role of productivity. In Geert Booij and Jaap van Marle, editors, *Yearbook of morphology 1991*, volume 26, pages 165–183. Kluwer, Dordrecht.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Hélène Giraudo and Jonathan Grainger. 2003. On the role of derivational affixes in recognizing complex words: Evidence from masked priming. In R. Harald Baayen and Robert Schreuder, editors, *Morphological structure in language processing*, pages 209–232. De Gruyter, Berlin.

Pius ten Hacken. 2014. Delineating derivation and inflection. In Rochelle Lieber and Pavol Štekauer, editors, *The Oxford handbook of derivational morphology*, pages 10–25. Oxford University Press, Oxford.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Annual Meeting of the Association for Computational Linguistics (ACL) 49*.

Martin Haspelmath and Andrea D. Sims. 2010. *Understanding morphology*. Routledge, New York, NY.

Benjamin Heinzerling and Michael Strube. 2019. Sequence tagging with contextual and non-contextual subword representations: A multilingual evaluation. In *Annual Meeting of the Association for Computational Linguistics (ACL) 57*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2019*.

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2020a. DagoBERT: Generating derivational morphology with a pretrained language model. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2020b. Predicting the growth of morphological families from social and linguistic factors. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Valentin Hofmann, Hinrich Schütze, and Janet B. Pierrehumbert. 2020c. A graph auto-encoder model of derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Abhilash Jain, Aku Rouhe, Stig-Arne Grönroos, and Mikko Kurimo. 2020. Finnish ASR with deep transformer models. In *Conference of the International Speech Communication Association (INTERSPEECH) 21*.

Ganesh Jawahar, Benoit Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Annual Meeting of the Association for Computational Linguistics (ACL) 57*.

Emilia Kacprzak, Laura M. Koesten, Luis-Daniel Ibáñez, Elena Simperl, and Jeni Tennison. 2017. A query log analysis of dataset search. In *International Conference on Web Engineering (ICWE) 17*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Conference on Artificial Intelligence (AAAI) 30*.

Diederik P. Kingma and Jimmy L. Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR) 3*.

Max Kisselew, Sebastian Padó, Alexis Palmer, and Jan Šnajder. 2015. Obtaining a better understanding of distributional models of german derivational morphology. In *International Conference on Computational Semantics (IWCS) 11*.

Matej Klemen, Luka Krsnik, and Marko Robnik-Šikonja. 2020. Enhancing deep neural networks with morphological information. In *arXiv 2011.12432*.

Victor Kuperman, Raymond Bertram, and R. Harald Baayen. 2008. Morphological dynamics in compound processing. *Language and Cognitive Processes*, 23(7-8):1089–1132.

Victor Kuperman, Robert Schreuder, Raymond Bertram, and R. Harald Baayen. 2009. Reading of polymorphemic Dutch compounds: Towards a multiple route model of lexical processing. *Journal of Experimental Psychology: Human Perception and Performance*, 35(3):876–895.

Alessandro Laudanna and Cristina Burani. 1985. Address mechanisms to decomposed lexical entries. *Linguistics*, 23(5).

Alessandro Laudanna and Cristina Burani. 1995. Distributional properties of derivational affixes: Implications for processing. In Laurie B. Feldman, editor, *Morphological aspects of language processing*, pages 345–364. Lawrence Erlbaum, Hillsdale, NJ.

Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Annual Meeting of the Association for Computational Linguistics (ACL) 51*.

Alina Leminen, Eva Smolka, Jon Duñabeitia, and Christos Pliatsikas. 2019. Morphological processing in the brain: The good (inflection), the bad (derivation) and the ugly (compounding). *Cortex*, 116:4–44.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Conference on Computational Natural Language Learning (CoNLL) 17*.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. CharBERT: Character-aware pre-trained language model. In *International Conference on Computational Linguistics (COLING) 28*.

Leon Manelis and David A. Tharp. 1977. The processing of affixed words. *Memory and Cognition*, 5(6):690–695.

Louis Martin, Benjamin Muller, Pedro J. Suárez, Yoann Dupont, Laurent Romary, de la Clergerie, Éric V., Djamé Seddah, and Benoit Sagot. 2020. CamemBERT: A tasty French language model. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS) 26*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Annual Meeting of the Association for Computational Linguistics (ACL) 47*.

Jeremy M. Needle and Janet B. Pierrehumbert. 2018. Gendered associations of english morphology. *Journal of the Association for Laboratory Phonology*, 9(1):119.

Boris New, Marc Brysbaert, Juan Segui, Ludovic Ferrand, and Kathleen Rastle. 2004. The processing of singular and plural nouns in french and english. *Journal of Memory and Language*, 51(4):568–585.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2019*.

Sebastian Padó, Aurélie Herbelot, Max Kisselew, and Jan Šnajder. 2016. Predictability of distributional semantics in derivational word formation. In *International Conference on Computational Linguistics (COLING) 26*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2014*.

Yuval Pinter, Cassandra L. Jacobs, and Jacob Eisenstein. 2020. Will it unblend? In *Findings of Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Ingo Plag. 2003. *Word-formation in English*. Cambridge University Press, Cambridge, UK.

Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Co-learning of word representations and morpheme representations. In *International Conference on Computational Linguistics (COLING) 25*.

Péter Rácz, Clay Beckner, Jennifer Hay, and Janet B. Pierrehumbert. 2020. Morphological convergence as on-line lexical analogy. *Language*, 96(4):735–770.

Péter Rácz, Janet B. Pierrehumbert, Jennifer Hay, and Viktória Papp. 2015. Morphological emergence. In Brian MacWhinney and William O'Grady, editors, *The handbook of language emergence*, pages 123–146. Wiley, Hoboken, NJ.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Kathleen Rastle and Matthew H. Davis. 2008. Morphological decomposition based on the analysis of orthography. *Language and Cognitive Processes*, 23(7-8):942–971.

Kathleen Rastle, Matthew H. Davis, and Boris New. 2004. The broth in my brother's brothel: Morpho-orthographic segmentation in visual word recognition. *Psychonomic Bulletin and Review*, 11(6):1090–1098.

Alexandre Salle and Aline Villavicencio. 2018. Incorporating subword information into matrix factorization word embeddings. In *Workshop on Subword/Character LEvel Models 2*.

Robert Schreuder and R. Harald Baayen. 1995. Modeling morphological processing. In Laurie B. Feldman, editor, *Morphological aspects of language processing*, pages 131–154. Lawrence Erlbaum, Hillsdale, NJ.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 37*.

Hinrich Schütze. 1992. Word space. In *Advances in Neural Information Processing Systems (NIPS) 5*, pages 895–902.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Annual Meeting of the Association for Computational Linguistics (ACL) 54*.

Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. 2020. Tired of topic models? Clusters of pretrained word embeddings make for fast and good topics too! In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Joseph P. Stemberger. 1994. Rule-less morphology at the phonology-lexicon interface. In Susan D. Lima, Roberta Corrigan, and Gregory Iverson, editors, *The reality of linguistic rules*, pages 147–169. John Benjamins, Amsterdam.

Gregory Stump. 2017. Rule conflation in an inferential-realizational theory of morphotactics. *Acta Linguistica Academica*, 64(1):79–124.

Gregory Stump. 2019. Some sources of apparent gaps in derivational paradigms. *Morphology*, 29(2):271–292.

Marcus Taft. 1979. Recognition of affixed words and the word frequency effect. *Memory and Cognition*, 7(4):263–272.

Marcus Taft. 1981. Prefix stripping revisited. *Journal of Verbal Learning and Verbal Behavior*, 20:289–297.

Marcus Taft. 1988. A morphological-decomposition model of lexical representation. *Linguistics*, 26:657–667.

Marcus Taft. 1991. *Reading and the mental lexicon*. Lawrence Erlbaum, Hove, UK.

Marcus Taft. 1994. Interactive-activation as a framework for understanding morphological processing. *Language and Cognitive Processes*, 9(3):271–294.

Marcus Taft. 2004. Morphological decomposition and the reverse base frequency effect. *The Quarterly Journal of Experimental Psychology*, 57(4):745–765.

Marcus Taft and Kenneth I. Forster. 1975. Lexical storage and retrieval of prefixed words. *Journal of Verbal Learning and Verbal Behavior*, 14:638–647.

Samson Tan, Shafiq Joty, Lav R. Varshney, and Min-Yen Kan. 2020. Mind your inflections! Improving NLP for non-standard Englishes with base-inflection encoding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Ivan Vulić, Edoardo M. Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin, and Trevor Cohn. 2017. Context-aware prediction of derivational word-forms. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL) 15*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le V, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith

Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv 1609.08144*.

Yi Yang and Jacob Eisenstein. 2017. Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics*, 5:295–307.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS) 33*.

A Appendices

A.1 Derivational Segmentation

Let A be a set of derivational affixes and S a set of stems. To determine the derivational segmentation of a word w, we employ an iterative algorithm. Define the set B_1^A of w as the words that remain when one derivational affix from A is removed from w. For example, `unlockable` can be segmented into `un`, `lockable` and `unlock`, `able` so $B_1^A(\texttt{unlockable}) = \{\texttt{lockable}, \texttt{unlock}\}$ (we assume that `un` and `able` are in A). We then iteratively create $B_{i+1}^A(w) = \bigcup_{b \in B_i^A(w)} B_1^A(b)$, i.e., we iteratively remove affixes from w. We stop as soon as $B_{i+1}^A(w) \cap S \neq \emptyset$. The element in this intersection, together with the used affixes from A, forms the derivational segmentation of w.[15] If there is no i such that $B_{i+1}^A(w) \cap S \neq \emptyset$, w does not have a derivational segmentation. The algorithm is sensitive to most morpho-orthographic rules of English (Plag, 2003), e.g., when the suffix `ize` is removed from `isotopize`, the resulting word is `isotope`, not `isotop`.

In this paper, we follow Hofmann et al. (2020a) in using BERT's prefixes, suffixes, and stems as input to the algorithm. Specifically, we assign 46 productive prefixes and 44 productive suffixes in BERT's vocabulary to A and all fully alphabetic words with more than 3 characters in BERT's vocabulary (excluding stopwords and affixes) to S, resulting in a total of 20,259 stems. This means that we only consider derivational segmentations that are possible given BERT's vocabulary.

[15] If $|B_{i+1}^A(w) \cap S| > 1$ (rarely the case in practice), the element with the lowest number of suffixes is chosen.

A.2 Data Preprocessing

We exclude texts written in a language other than English and remove strings containing numbers as well as hyperlinks. We follow Han and Baldwin (2011) in reducing repetitions of more than three letters (`niiiiice`) to three letters.

A.3 Hyperparameters

The feed-forward network has a ReLU activation after the first layer and a sigmoid activation after the second layer. The first layer has 100 dimensions. We apply dropout of 0.2 after the first layer. All other hyperparameters are as for $\text{BERT}_{\text{BASE}}$ (uncased) (Devlin et al., 2019). The number of trainable parameters is 109,559,241.

We use a batch size of 64 and perform grid search for the number of epochs $n \in \{1, \ldots, 20\}$ and the learning rate $l \in \{1 \times 10^{-6}, 3 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$ (selection criterion: F1 score). We tune l on Reddit (80 hyperparameter search trials per model type) and use the best configuration (which is identical for all model types) for 20 training runs with different random seeds on all three datasets (20 hyperparameter search trials per model type, dataset, and random seed). Models are trained with binary cross-entropy as the loss function and Adam (Kingma and Ba, 2015) as the optimizer. Experiments are performed on a GeForce GTX 1080 Ti GPU (11GB).

Table 4 lists statistics of the validation performance over hyperparameter search trials and provides information about best hyperparameter configurations as well as runtimes.[16] See also Section 3.5 and particularly Figure 2 in the main text, where we present a detailed analysis of the convergence behavior of the two main model types examined in this study (DelBERT and BERT).

[16] Since expected validation performance (Dodge et al., 2019) may not be correct for grid search, we report mean and standard deviation of the performance instead.

	Amazon					ArXiv					Reddit				
Model	μ	σ	n	l	τ	μ	σ	n	l	τ	μ	σ	n	l	τ
DelBERT	.627	.007	6.75	3e-06	67.73	.725	.006	11.45	3e-06	28.69	.687	.006	5.45	3e-06	25.56
BERT	.612	.006	7.30	3e-06	66.18	.693	.015	17.05	3e-06	28.04	.657	.007	9.25	3e-06	25.06
Stem	.556	.016	9.85	3e-06	67.43	.699	.005	8.15	3e-06	28.56	.670	.006	6.00	3e-06	25.39
Affixes	.519	.008	5.55	3e-06	67.70	.599	.004	7.50	3e-06	28.43	.593	.003	9.35	3e-06	25.49

Table 4: Validation performance statistics and hyperparameter search details. The table shows the mean (μ) and standard deviation (σ) of the validation performance (F1) on all hyperparameter search trials, the number of epochs (n) and learning rate (l) with the best validation performance, and the runtime (τ) in minutes for one full hyperparameter search (20 trials). The numbers are averaged across 20 training runs with different random seeds.

7

An Embarrassingly Simple Method

Contents

The study reported in chapter 6 shows that a morphologically informed input tokenization can boost the performance of language models. However, the proposed method relies on having access to the morphological gold segmentation, which is not always available in practice. The fifth study addresses this problem: it develops a recursive algorithm that does not require any external resources while still leading to tokenizations with a substantially improved morphological quality. I evaluate the performance of several language models equipped with this tokenization method and compare against the same language models using their standard tokenizers.

An Embarrassingly Simple Method to Mitigate Undesirable Properties of Pretrained Language Model Tokenizers

Valentin Hofmann[*‡], **Hinrich Schütze**[‡], **Janet B. Pierrehumbert**[†*]

[*]Faculty of Linguistics, University of Oxford
[†]Department of Engineering Science, University of Oxford
[‡]Center for Information and Language Processing, LMU Munich
valentin.hofmann@ling-phil.ox.ac.uk

Abstract

We introduce FLOTA (Few Longest Token Approximation), a simple yet effective method to improve the tokenization of pretrained language models (PLMs). FLOTA uses the vocabulary of a standard tokenizer but tries to preserve the morphological structure of words during tokenization. We evaluate FLOTA on morphological gold segmentations as well as a text classification task, using BERT, GPT-2, and XLNet as example PLMs. FLOTA leads to performance gains, makes inference more efficient, and enhances the robustness of PLMs with respect to whitespace noise.

1 Introduction

The first step in NLP architectures using pretrained language models (PLMs) is to map text to a sequence of tokens corresponding to input embeddings. The tokenizers used to accomplish this have been shown to exhibit various undesirable properties such as generating segmentations that blur word meaning (Bostrom and Durrett, 2020; Church, 2020; Hofmann et al., 2021) and generalizing suboptimally to new domains (Tan et al., 2020; Hong et al., 2021; Sachidananda et al., 2021).

In this paper, we propose **FLOTA** (**F**ew **Lo**ngest **T**oken **A**pproximation), a simple yet effective method to mitigate some shortcomings of PLM tokenizers. FLOTA is motivated by the following hypothesis: rather than finding a segmentation that *covers all characters* of a word but *destroys its morphological structure*, it can be more beneficial to find a segmentation that *does not cover all characters* but *preserves key aspects of the morphology*. We confirm this hypothesis in this paper.

Our study investigates three PLMs and corresponding tokenizers: BERT (base, uncased; Devlin et al., 2019), which uses WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016), GPT-2 (base, cased; Radford et al., 2019), which uses byte-pair encoding (BPE; Gage, 1994; Sennrich et al., 2016), and XLNet (base, cased; Yang et al., 2019), which uses Unigram (Kudo, 2018). We find that FLOTA increases the morphological quality of all tokenizers as evaluated on human-annotated gold segmentations as well as the performance of all PLMs on a text classification challenge set.

Contributions. We introduce FLOTA, a simple yet effective method to improve the tokenization of PLMs during finetuning. FLOTA uses the vocabulary of a standard tokenizer but tries to preserve the morphological structure of words during tokenization. We show that FLOTA has three advantages compared to standard tokenization: (i) it can increase the performance of PLMs on certain tasks, sometimes substantially; (ii) it makes inference more efficient by shortening the processed token sequences; (iii) it enhances the robustness of PLMs with respect to certain types of noise in the data. All this is achieved *without requiring any additional parameters or resources* compared to vanilla PLM finetuning. We also release a text classification challenge set that can serve as a benchmark for future studies on PLM tokenizers.[1]

2 Few Longest Token Approximation

Let V be a set of tokens that constitute the vocabulary of a tokenizer. For the tokenizers discussed in this paper, V contains words, subwords, and characters. Let ϕ be a model used by the tokenizer to map text to a sequence of tokens from V.

FLOTA (Few Longest Token Approximation) discards ϕ and uses V in a modified way. Given a word w not in V, FLOTA tokenizes it by determining the longest substring $s \in V$ of w, returning s, and recursing on $w \setminus s$, the string(s) remaining when s is removed from w. We stop after k recursive calls or when the residue is null. Figure 1 provides pseudocode. For the example word *undesirable* and $k = 2$, FLOTA first searches on

[1]We make our code and data available at https://github.com/valentinhofmann/flota.

MAXSUBWORDSPLIT(w, V)

$l = length(w)$
for $j = l$ **downto** 0
  for $i = 0$ **to** $l - j + 1$
    $s = w[i \,..\, i + j]$
    if $s \in V$
      $r = w[0 \,..\, i] \oplus_j w[i + j \,..\, l]$
      return s, r, i

FLOTATOKENIZE(w, k, V)

$s, r, i =$ MAXSUBWORDSPLIT(w, V)
if k == 1 or $hyphen(r)$
  $F = \{\}$
  $F[i] = s$
  return F
$F =$ FLOTATOKENIZE$(r, k - 1, V)$
$F[i] = s$
return F

Figure 1: FLOTA pseudocode. FLOTA is based on a recursive function FLOTATOKENIZE that uses a hash table F to store the longest substring s and its index i on each recursive call. s and i are found by means of a second function MAXSUBWORDSPLIT, which also returns a residue r. In practice, to ensure correct indexing throughout different recursive calls as well as prevent using discontinuous substrings for tokenization, we compute r using an operation $\oplus_j$ that concatenates two strings by putting j (length of s) hyphens between them. The recursion stops after k recursive calls or when r only consists of hyphens (determined by a boolean function $hyphen$). The hash table returned by FLOTATOKENIZE is converted to a tokenization using a simple wrapper function that sorts the found substrings by their indices (not shown). If MAXSUBWORDSPLIT does not find a substring $s \in V$, FLOTATOKENIZE returns an empty hash table (not shown).

undesirable and finds `desirable`, then searches on `un---------` and finds `un`, then stops (since $k = 2$; it would also stop for $k > 2$ since the residue is null) and returns the tokenization `un`, `desirable`. The WordPiece tokenization, on the other hand, is `und`, `es`, `ira`, `ble`.

FLOTA is guided by the following observations: many words not in V are made up of smaller and typically more frequent elements that determine their meaning (e.g., they are derivatives such as *undesirable*); many of these elements are in V.[2] By recursively searching for the longest substrings, we hope to recover the most important meaningful elements. This is also why it makes sense to stop after k recursions: if FLOTA returns the most important meaningful elements as the first few tokens, we expect to not lose much by stopping.

[2]Existing tokenizers have been shown to be able to recover these elements only to a very limited extent (Bostrom and Durrett, 2020; Church, 2020; Hofmann et al., 2021).

Model	Tokenization	$\overline{C}$	$\overline{R}$	$\overline{M}$
BERT	FIRST	.869	.817	.664
BERT	LONGEST	.865	.797	.664
BERT	FLOTA	**.990**	**.876**	**.896**
GPT-2	FIRST	.878	.674	.625
GPT-2	LONGEST	.874	.674	.625
GPT-2	FLOTA	**.988**	**.845**	**.861**
XLNet	FIRST	.886	.820	.724
XLNet	LONGEST	902	.845	.756
XLNet	FLOTA	**.992**	**.900**	**.922**

Table 1: Morphological quality. $\overline{C}$: morphological coverage ($k = 2$); $\overline{R}$: stem recall; $\overline{M}$: full match.

3 Evaluation on Gold Segmentations

English inflection is simple, but the language has highly complex word formation, i.e., derivation and compounding (Cotterell et al., 2017; Pierrehumbert and Granell, 2018). To evaluate the morphological quality of FLOTA against the standard tokenizers, we thus focus on derivatives and compounds.

Data. Our evaluation uses CELEX (Baayen et al., 1995) and LADEC (Gagné et al., 2019), two large datasets of human-annotated gold segmentations of morphologically complex words. We merge both datasets and extract all words consisting of a prefix and a stem (prefixed derivatives), a stem and a suffix (suffixed derivatives), or two stems (compounds). We create for each PLM a subset of words where both morphological elements (i.e., stems and affixes) are in the tokenizer vocabulary, but the word itself is not in the tokenizer vocabulary. In such cases, a word needs to be segmented, and it is guaranteed that *the gold segmentation is possible* given the tokenizer vocabulary. This procedure results in 11,272, 11,253, 10,848 words for BERT, GPT-2, XLNet, respectively.

Experimental Setup. We define three metrics to analyze how closely FLOTA matches the gold segmentations. We compare against two alternative tokenization strategies: representing words as the k first tokens returned by the standard tokenizer (FIRST) and representing words as the k longest tokens returned by the standard tokenizer (LONGEST). Recall that the WordPiece tokenization of the running example *undesirable* is `und`, `es`, `ira`, `ble`. With $k = 3$, FIRST is `und`, `es`, `ira` (i.e., it simply returns the first k tokens) and LONGEST is `und`, `ira`, `ble` (i.e., it returns the k

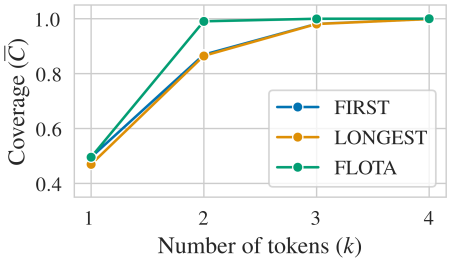


Figure 2: Morphological coverage for varying k.

longest tokens in the order in which they occur in the standard tokenization).

Morphological coverage. We analyze what proportion of morphological elements is covered by each tokenization strategy for varying k, a measure that we call *morphological coverage*, C. For *undesirable* and $k = 3$, FIRST and LONGEST contain *un* ($C = 0.5$) while FLOTA contains both *un* and *desirable* ($C = 1$). We compute the mean morphological coverage across all words, $\overline{C}$.

We find that for all three tokenizers, FLOTA already covers about 99% of the morphological elements with just $k = 2$, a value that FIRST and LONGEST only reach with $k = 4$ (Table 1, Figure 2), indicating that FLOTA needs considerably fewer tokens than the standard tokenization to convey the same amount of semantic and syntactic information. This can also be seen by examining the average number of tokens needed to fully tokenize a word (i.e., $k = \infty$), with the values for FLOTA (BERT: 2.02; GPT-2: 2.03; XLNet: 2.02) being lower than the values for the standard tokenization (BERT: 2.30; GPT-2: 2.23; XLNet: 2.26). The pairwise differences are statistically significant ($p < 0.001$) as shown by two-tailed t-tests.

Stem recall. Given its relevance for the overall lexical meaning of a word, we are interested in how often FLOTA returns the stem at $k = 1$. We test this using a measure that we call *stem recall*, R ($R = 1$ if the token is the stem,[3] otherwise $R = 0$), and compute the mean stem recall $\overline{R}$ across all words. We again compare with FIRST and LONGEST. Notice the stem according to the gold segmentation is longer than the second morphological element in 97% of the examined complex words, which means that LONGEST provides a close estimate of how often the full standard tokenization contains the stem (since any other element in the full standard tokenization is shorter and hence very unlikely to be the stem).

[3]For compounds: one of the two stems.

FLOTA returns the stem considerably more often than either FIRST or LONGEST, but there are clear differences between the models (Table 1): for GPT-2, FLOTA increases $\overline{R}$ by more than 15% while the difference amounts to 5% for XLNet.

Full match. Extending the evaluation of stem recall, we examine whether the tokenization at $k = 2$ is identical to the gold segmentation (which always has two elements) using a measure that we call *full match*, M ($M = 1$ if the tokenization exactly matches the gold segmentation, otherwise $M = 0$). We again compute the mean value $\overline{M}$ across all words. Here, the values for both FIRST and LONGEST are identical to the performance of the full standard tokenization: for the full standard tokenization to exactly match a segmentation of two elements, it must consist of two tokens, and hence it is necessarily equal to both its first two tokens and its longest two tokens.[4] Table 1 shows that FLOTA substantially improves $\overline{M}$.

The evaluation on gold segmentations indicates that FLOTA increases the morphological quality of PLM tokenizers compared to the standard tokenization and simple alternatives. We also find underlying differences in the morphological quality of the tokenizers, with BPE and Unigram lying at the negative and positive extremes, in line with prior work (Bostrom and Durrett, 2020). Our analysis shows that WordPiece lies in between.

4 Evaluation on Downstream Task

We investigate whether the enhanced quality of FLOTA tokenizations translates to performance on downstream tasks. We focus on text classification as one of the most common tasks in NLP.

Data. We create two text classification challenge sets based on ArXiv,[5] each consisting of three datasets. Specifically, for the subject areas of computer science, maths, and physics, we extract titles for the 20 most frequent subareas (e.g., *Computation and Language*). We then sample 100/1,000 titles per subarea, resulting in three text classification datasets of 2,000/20,000 titles each, which we bundle together as ArXiv-S/L. Our sampling

[4]Surprisingly, this does not hold for Unigram, which sometimes creates *separate* start-of-word tokens; e.g., the Unigram tokenization of *americanize* is _, `american`, `ize`, where _ is a start-of-word token. Notice that in such cases (with $k = 2$), LONGEST (`american`, `ize`) matches the gold segmentation while FIRST (_, `american`) does not, explaining the performance difference for XLNet.

[5]`kaggle.com/Cornell-University/arxiv`

	ArXiv-S		ArXiv-L	
Model	Dev	Test	Dev	Test
BERT	.469	.470	.674	.659
+FLOTA	**.491**	**.485**	**.675**	**.661**
GPT-2	.329	.324	.526	.507
+FLOTA	**.353**	**.382**	**.558**	**.542**
XLNet	.435	**.454**	.660	.641
+FLOTA	**.446**	.428	**.664**	**.646**

Table 2: Performance. FLOTA leads to gains in averaged F1, particularly for BERT and GPT-2. Performance breakdowns for the individual datasets forming ArXiv-S/L are provided in Appendix A.3.

ensures that ArXiv-S/L require challenging generalization from a small number of short training examples with highly complex language. See Appendix A.1 for more details.

Experimental Setup. We split the six datasets of ArXiv-S and ArXiv-L into 60% train, 20% dev, and 20% test. We then train the three PLMs with classification heads on the six train splits, once with the standard tokenizers and once with FLOTA. See Appendix A.2 for hyperparameters. For FLOTA, we treat k as an additional tunable hyperparameter. We use F1 as the evaluation metric.

Performance. The FLOTA models perform better than the models with standard tokenization, albeit to varying degrees for the three PLMs (Table 2). The difference is most pronounced for GPT-2, with FLOTA resulting in large performance gains of up to 5%. In addition, GPT-2 performs worse than the other two PLMs on all datasets, suggesting that BPE is generally not a good fit for complex language. BERT also clearly benefits from using FLOTA, particularly on ArXiv-S. Out of the three considered PLMs, XLNet obtains the smallest performance gain from using FLOTA, but it still benefits in the majority of cases.

The advantage of FLOTA mirrors the differences observed in the morphological analysis, indicating that *FLOTA helps close the morphological quality gap* between standard tokenizations and gold segmentations. Where the gap is large, gains due to FLOTA are large (GPT-2/BPE); where it is small, gains due to FLOTA are small (XLNet/Unigram). BERT/WordPiece again lies in between.

Impact of k. To test how the performance varies with k, we focus on BERT and compare the FLOTA models for $k \in \{1, 2, 3, 4\}$ with the two alternatives FIRST and LONGEST from Section 3. See Appendix A.4 for hyperparameters.

Figure 3 shows that FLOTA only drops slightly as we decrease k, with the minimum F1 at $k = 1$

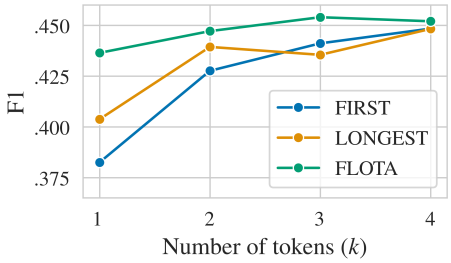


Figure 3: FLOTA is less impaired by smaller values of k (maximum number of tokens per word) than FIRST/LONGEST. Results are averaged F1 of BERT on ArXiv-S (dev/test merged).

		FLOTA			
Model	ST	$k = 1$	$k = 2$	$k = 3$	$k = 4$
BERT	12.9	8.3	10.5	11.4	11.6
GPT-2	12.9	8.3	10.7	11.5	11.8
XLNet	13.6	8.3	10.9	11.9	12.2

Table 3: Average sequence length of titles (ArXiv-L, physics). ST: standard tokenization.

(43.6%) lying less than 2% below the maximum F1 at $k = 3$ (45.4%). In contrast, FIRST and LONGEST drop substantially as we decrease k; for FIRST, the minimum F1 at $k = 1$ (38.2%) lies more than 6% below the maximum F1 at $k = 4$ (44.8%). The fact that FLOTA is more effective at preserving performance while reducing the number of tokens aligns with the observation that it covers a larger number of morphemes and hence more semantic and syntactic content than FIRST and LONGEST for small k (Section 3).

Efficiency. FLOTA allows to reduce the number of tokens used to tokenize text by varying k. Since the attention mechanism scales quadratically with sequence length (Peng et al., 2021), this has beneficial effects on the computational cost involved with employing a model trained using FLOTA. We empirically find that even for $k = 4$ (the largest value used in the experiments), token sequences generated by FLOTA are on average shorter than the token sequences generated by the standard tokenizations. Table 3 shows for one dataset (ArXiv-L, physics) the average sequence length of titles encoded with the standard tokenization versus FLOTA with varying $k \in \{1, 2, 3, 4\}$ for the three PLMs.

Robustness. To examine robustness against noise, a well-known problem for PLMs (Pruthi et al., 2019), we focus on missing whitespace between words (Soni et al., 2019). We randomly drop the whitespace between two adjacent words with

	ArXiv-S (N)		ArXiv-L (N)	
Model	Dev	Test	Dev	Test
BERT	.428	.412	.579	.554
+FLOTA	**.486**	**.447**	**.652**	**.632**
GPT-2	.313	.315	.481	.463
+FLOTA	**.359**	**.357**	**.541**	**.518**
XLNet	.392	.397	.609	.589
+FLOTA	**.434**	**.421**	**.641**	**.623**

Table 4: Performance with noise (N). FLOTA clearly increases F1 on ArXiv-S/L for all PLMs when input is noisy. See Appendix A.5 for hyperparameters. Performance breakdowns for the individual datasets forming ArXiv-S/L are provided in Appendix A.6.

probability $p = 0.3$ in ArXiv-S/L. We use *unseen* noise, i.e., we only inject noise during evaluation, not training, which is the more realistic and challenging scenario (Xue et al., 2021).

The results show that synthetic noise increases the performance gap between FLOTA and standard tokenization (Table 4). While there is a drop in performance for all models compared to the experiments without noise, the drop is much more pronounced for standard tokenization; e.g., BERT's performance on ArXiv-L (test) drops by 3% with FLOTA, but by 10% without it.

5 Limitations

While we find FLOTA to work well on text classification, there are tasks for which FLOTA might prove a less suitable tokenization method: e.g., for small values of k, FLOTA often discards suffixes, which can be important for tasks with a syntactic component such as POS tagging.

Similar considerations hold for transfer to languages other than English: e.g., in the case of languages with a non-linear morphology such as Arabic, FLOTA is expected to inherit the insufficiencies of the underlying tokenizer (Alkaoud and Syed, 2020; Antoun et al., 2020).

6 Related Work

The question how PLMs are affected by their tokenizer has attracted growing interest recently. Bostrom and Durrett (2020), Church (2020), Klein and Tsarfaty (2020), and Hofmann et al. (2021) focus on the linguistic properties of tokenizers. We contribute to this line of work by conducting the first comparative analysis of all three common PLM tokenizers and releasing a challenge set as a benchmark for future studies. Another strand of research has sought to improve PLM tokenizers by training models from scratch (Clark et al., 2021; Si et al., 2021; Xue et al., 2021; Zhang et al., 2021) or modifying the tokenizer during finetuning, mostly by adding tokens and corresponding embeddings (Chau et al., 2020; Tan et al., 2020; Hong et al., 2021; Sachidananda et al., 2021). FLOTA crucially differs in that it can be used during finetuning but does not add any parameters to the PLM. Furthermore, there has been work improving tokenization by variously exploiting the probabilistic nature of tokenizers (Kudo, 2018; Provilkov et al., 2020; Cao and Rimell, 2021). By contrast, our method does not need access to the underlying model.

Our study also relates to computational work on derivational morphology (Cotterell et al., 2017; Vylomova et al., 2017; Cotterell and Schütze, 2018; Deutsch et al., 2018; Hofmann et al., 2020a,b,c) and word segmentation (Cotterell et al., 2016; Kann et al., 2016; Ruzsics and Samardžić, 2017; Mager et al., 2019, 2020; Seker and Tsarfaty, 2020; Amrhein and Sennrich, 2021). We are the first to systematically evaluate the segmentations of PLM tokenizers on human-annotated gold data.

Conceptually, the findings of our study are in line with evidence from the cognitive sciences that knowledge of a longer (i.e., more detailed and informative) sequence takes priority over any knowledge about smaller sequences (Caramazza et al., 1988; Laudanna and Burani, 1995; Baayen et al., 1997; Needle and Pierrehumbert, 2018).

7 Conclusion

We introduce FLOTA (Few Longest Token Approximation), a simple yet effective method to improve the tokenization of pretrained language models (PLMs). FLOTA uses the vocabulary of a standard tokenizer but tries to preserve the morphological structure of words during tokenization. FLOTA leads to performance gains, makes inference more efficient, and substantially enhances the robustness of PLMs with respect to whitespace noise.

Acknowledgements

This work was funded by the European Research Council (#740516) and the Engineering and Physical Sciences Research Council (EP/T023333/1). The first author was also supported by the German Academic Scholarship Foundation and the Arts and Humanities Research Council. We thank the reviewers for their helpful comments.

Ethical Considerations

FLOTA shortens the average length of sequences processed by PLMs, thus reducing their energy requirements, a desirable property given their otherwise detrimental environmental footprint (Schwartz et al., 2019; Strubell et al., 2019).

References

Mohamed Alkaoud and Mairaj Syed. 2020. On the importance of tokenization in Arabic embedding models. In *Arabic Natural Language Processing Workshop (WANLP) 5*.

Chantal Amrhein and Rico Sennrich. 2021. How suitable are subword segmentation strategies for translating non-concatenative morphology? In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*.

R. Harald Baayen, Ton Dijkstra, and Robert Schreuder. 1997. Singulars and plurals in Dutch: Evidence for a parallel dual-route model. *Journal of Memory and Language*, 37:94–117.

R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1995. *The CELEX lexical database (CD-ROM)*. Linguistic Data Consortium, Philadelphia, PA.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Kris Cao and Laura Rimell. 2021. You should evaluate your language model on marginal likelihood over tokenisations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2021*.

Alfonso Caramazza, Alessandro Laudanna, and Cristina Romani. 1988. Lexical access and inflectional morphology. *Cognition*, 28(297-332).

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Kenneth Church. 2020. Emerging trends: Subwords, seriously? *Natural Language Engineering*, 26(3):375–382.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. CANINE: Pre-training an efficient tokenization-free encoder for language representation. In *arXiv 2103.06874*.

Ryan Cotterell and Hinrich Schütze. 2018. Joint semantic synthesis and morphological analysis of the derived word. *Transactions of the Association for Computational Linguistics*, 6:33–48.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016. A joint model of orthography and morphological segmentation. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2016*.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017. Paradigm completion for derivational morphology. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017*.

David Crystal. 1997. *The Cambridge encyclopedia of the English language*. Cambridge University Press, Cambridge, UK.

Daniel Deutsch, John Hewitt, and Dan Roth. 2018. A distributional and orthographic aggregation model for English derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2019*.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Christina L. Gagné, Thomas L. Spalding, and Daniel Schmidtke. 2019. LADEC: The large database of English compounds. *Behavior Research Methods*, 51(5):2152–2179.

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2020a. DagoBERT: Generating derivational morphology with a pretrained language model. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2020b. Predicting the growth of morphological families from social and linguistic factors. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Improving BERT's interpretations of complex words with derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 59*.

Valentin Hofmann, Hinrich Schütze, and Janet B. Pierrehumbert. 2020c. A graph auto-encoder model of derivational morphology. In *Annual Meeting of*

the Association for Computational Linguistics (ACL) 58.

Jimin Hong, Taehee Kim, Hyesu Lim, and Jaegul Choo. 2021. AVocaDo: Strategy for adapting vocabulary to downstream domain. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2021*.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2016*.

Diederik P. Kingma and Jimmy L. Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR) 3*.

Stav Klein and Reut Tsarfaty. 2020. Getting the ##life out of living: How adequate are word-pieces for modelling complex morphology? In *Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON) 17*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*.

Alessandro Laudanna and Cristina Burani. 1995. Distributional properties of derivational affixes: Implications for processing. In Laurie B. Feldman, editor, *Morphological aspects of language processing*, pages 345–364. Lawrence Erlbaum, Hillsdale, NJ.

Manuel Mager, Özlem Çetinoğlu, and Katharina Kann. 2019. Subword-level language identification for intra-word code-switching. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2019*.

Manuel Mager, Özlem Çetinoğlu, and Katharina Kann. 2020. Tackling the low-resource challenge for canonical segmentation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Jeremy M. Needle and Janet B. Pierrehumbert. 2018. Gendered associations of english morphology. *Journal of the Association for Laboratory Phonology*, 9(1):119.

Hao Peng, Jungo Kasai, Nikolaos Pappas, Dani Yogatama, Zhaofeng Wu, Lingpeng Kong, Roy Schwartz, and Noah A. Smith. 2021. ABC: Attention with bounded-memory control. In *arXiv 2110.02488*.

Janet B. Pierrehumbert and Ramon Granell. 2018. On hapax legomena and morphological productivity. In *Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON) 15*.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Annual Meeting of the Association for Computational Linguistics (ACL) 57*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Tatyana Ruzsics and Tanja Samardžić. 2017. Neural sequence-to-sequence learning of internal word structure. In *Conference on Computational Natural Language Learning (CoNLL) 21*.

Vin Sachidananda, Jason S. Kessler, and Yi-An Lai. 2021. Efficient domain adaptation of language models via adaptive tokenization. In *Workshop on Simple and Efficient Natural Language Processing (SustaiNLP) 2*.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 37*.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. Green AI. In *arXiv 1907.10597*.

Amit Seker and Reut Tsarfaty. 2020. A pointer network architecture for joint morphological segmentation and tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Annual Meeting of the Association for Computational Linguistics (ACL) 54*.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2021. SHUOWEN-JIEZI: Linguistically informed tokenizers for Chinese language model pretraining. In *arXiv 2106.00400*.

Sandeep Soni, Lauren F. Klein, and Jacob Eisenstein. 2019. Correcting whitespace errors in digitized historical texts. In *Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature 3*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Annual Meeting of the Association for Computational Linguistics (ACL) 57*.

Samson Tan, Shafiq Joty, Lav R. Varshney, and Min-Yen Kan. 2020. Mind your inflections! Improving NLP for non-standard Englishes with base-inflection encoding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin, and Trevor Cohn. 2017. Context-aware prediction of derivational word-forms. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL) 15*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le V, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv 1609.08144*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2021. ByT5: Towards a token-free future with pre-trained byte-to-byte models. In *arXiv 2105.13626*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS) 33*.

Xinsong Zhang, Pengshuai Li, and Hang Li. 2021. AMBERT: A pre-trained language model with multi-grained tokenization. In *Findings of the Association for Computational Linguistics: ACL 2021*.

A Appendix

A.1 Preprocessing

We exclude texts written in a language other than English and lowercase all words. We exclude titles with less than three and more than ten words. For each title, we compute the proportion of words starting with a productive prefix from the list provided by Crystal (1997). During sampling, we then weight titles by this proportion in order to make the language contained within the datasets as complex and challenging as possible.

A.2 Hyperparameters

The vocabulary size is 28,996 for BERT, 50,257 for GPT-2, and 32,000 for XLNet. The number of trainable parameters is 109,497,620 for BERT, 124,455,168 for GPT-2, and 117,324,308 for XLNet. The classification head for all three models uses softmax as the activation function.

We use a batch size of 64 and perform grid search for the number of epochs $n \in \{1, \ldots, 20\}$ and the learning rate $l \in \{1 \times 10^{-5}, 3 \times 10^{-5}, 1 \times 10^{-4}\}$ (selection criterion: F1). We tune l on ArXiv-L (physics) and use the best configuration on all datasets. For the FLOTA models, we additionally tune $k \in \{1, 2, 3, 4\}$ (selection criterion: F1). Models are trained with categorical cross-entropy as the loss function and Adam (Kingma and Ba, 2015) as the optimizer. Experiments are performed on a GeForce GTX 1080 Ti GPU (11GB).

A.3 Performance

Table 5 provides breakdowns of the performance for the individual datasets forming ArXiv-S/L.

A.4 Hyperparameters

All hyperparameters are as for the main experiment (see Appendix A.2). For the learning rate, we use the best configuration from the main experiment. For FIRST and LONGEST, we tune $k \in \{1, 2, 3, 4\}$ (selection criterion: F1), identically to FLOTA in the main experiment.

A.5 Hyperparameters

All hyperparameters are as for the main experiment (see Appendix A.2). For the learning rate, we use the best configuration from the main experiment.

A.6 Performance

Table 6 provides breakdowns of the performance for the individual datasets forming ArXiv-S/L.

	ArXiv-S						ArXiv-L					
	Dev			Test			Dev			Test		
Model	CS	MATH	PHYS	CS	MATH	PHYS	CS	MATH	PHYS	CS	MATH	PHYS
BERT	.546	.358	.502	.498	.407	.504	.682	.660	.679	.649	.653	.675
+FLOTA	.546	.414	.514	.483	.404	.567	.677	.663	.686	.652	.658	.672
GPT-2	.354	.281	.353	.316	.261	.395	.493	.506	.578	.465	.498	.559
+FLOTA	.348	.313	.398	.370	.323	.454	.520	.549	.603	.498	.540	.587
XLNet	.473	.357	.476	.489	.358	.515	.654	.643	.684	.627	.642	.655
+FLOTA	.450	.402	.486	.415	.346	.522	.660	.651	.681	.633	.641	.665

Table 5: Performance (F1). CS: computer science; MATH: mathematics; PHYS: physics.

	ArXiv-S (N)						ArXiv-L (N)					
	Dev			Test			Dev			Test		
Model	CS	MATH	PHYS	CS	MATH	PHYS	CS	MATH	PHYS	CS	MATH	PHYS
BERT	.479	.333	.470	.566	.566	.605	.417	.338	.481	.531	.544	.588
+FLOTA	.548	.400	.511	.652	.640	.664	.452	.372	.518	.631	.620	.644
GPT-2	.336	.261	.342	.452	.461	.530	.326	.252	.366	.423	.454	.511
+FLOTA	.358	.316	.402	.514	.527	.582	.370	.296	.405	.481	.511	.562
XLNet	.431	.311	.433	.607	.594	.625	.470	.300	.421	.587	.576	.605
+FLOTA	.432	.398	.474	.646	.623	.655	.435	.360	.466	.627	.612	.631

Table 6: Performance (F1) with noise (N). CS: computer science; MATH: mathematics; PHYS: physics.

8 Conclusion

Contents

8.1 Summary of Contributions

In line with the objectives of this thesis (§1.4), the contributions fall into two parts. First, the research of this thesis shows that including semantic and extralinguistic factors into models of derivational morphology enables a degree of predictability previosuly deemed impossible by some scholars. The following distillation of the key findings will follow the three aspects of predictability highlighted in the introduction.

8.1.1 Predicting the Existence of Derivatives

Chapter 3 examines the problem of predicting how likely a particular derivative, consisting of a base and a set of affixes, is to exist. It proposes a model, the so-called *derivational graph auto-encoder*, that has two crucial properties: (i) it operates on a (bipartite) graph representation of the mental lexicon, allowing it to naturally capture analogical

effects in derivation (Arndt-Lappe, 2014), and (ii) it leverages (prediction-based) meaning representations of both bases and affixes. These two characteristics make it possible for the derivational graph auto-encoder to learn semantic facets of analogical mechanisms in the mental lexicon, e.g., the impact of lexical-semantic gangs (Alegre and Gordon, 1999). The architectural choice of a graph neural network that is trained in an end-to-end fashion on a self-supervised objective allows the model to detect the relevant semantic-analogical patterns itself. In the experiments, which are based on derivatives taken from Reddit (Baumgartner et al., 2020), the derivational graph auto-encoder outperforms models without access to meaning representations, ranging from a relatively naive bigram model that implements classical ideas about affix ordering constraints in derivational morphology to more sophisticated models such as an ablated model that is identical to the derivational graph auto-encoder, with the difference that it does not have access to the semantic representations of base and affixes. The paper also analyzes how different extralinguistic contexts (represented by topically distinct subreddits) affect the propensity of speakers to form certain derivatives and finds strong effects, which are reflected by the predictions of the derivational graph auto-encoder.

8.1.2 Predicting the Future Creation of Derivatives

Chapter 4 examines a problem closely related to predicting the existence of derivatives, but with a twist: while in chapter 3 the test derivatives are taken from the same time point as the train derivatives (i.e., a *synchronic* derivational system is divided into train and test sets), chapter 4 conducts *diachronic* splits, i.e., models are trained on a derivational system at a certain point in time and tested on the derivational system at a later point in time. To make the task feasible, I focus on the question of whether the morphological family of a base will grow or not in the future, i.e., whether a base will become the basis for the creation of new derivatives in the future (and not how exactly these derivatives will look like). The derivatives are again taken from Reddit, which allows us to examine what subgroups of the user base create a certain derivative. In the experiments, I train random forest models on this task, focusing on the relative predictive power of language-internal factors (e.g., the size of the morphological family) vs. language-external factors

(temporal trend behavior and social dissemination). The results show that while the strongest predictor of growth is the morphological family size (a language-internal factor), especially the initial growth of morphological families is driven by language-external factors—for bases with small morphological families (i.e., only one or two lexemes), the language-external factors turn out to be the key determining factor for whether speakers will create corresponding derivatives in the future.

8.1.3 Predicting the Affixes Used to Create Derivatives

Chapter 5 examines the task of predicting the affix that is used to create a certain derivative, drawing upon a language model. Specifically, the chapter examines a large set of English derivational affixes and provides the language model with carrier sentences to guide it towards the subset of semantically and syntactically possible affixes. The study finds that the language model is capable of performing this task surprisingly well, substantially better than baselines with worse semantic representations.

However, chapter 5 also highlights a severe shortcoming of current language models: when the input tokenization is morphologically invalid, i.e., when the boundaries between tokens do not correspond to morpheme boundaries (e.g., *prem-at-uration*), the capability of the language model to predict derivation reduces drastically. This finding can be seen as a direct consequence of the central role of semantics for the predictability of derivation: when the input tokenization cuts through morphemes, the resulting sequence of tokens will have several elements that do not bear a semantic meaning. If good meaning representations are a key ingredient for successful predictive models of derivational morphology, it is expected that semantically deteriorated token sequences should impede the language model's performance. In chapter 5, I specifically show that high performance can be restored by forcing the language model to use a morphologically valid tokenization in cases where the default tokenization would be problematic.

Thus, all three chapters provide evidence for the crucial role of semantic and extralinguistic factors for predicting derivation. Including these two factors consistently improves performance across tasks and experiments. At the same time, chapter 5 points toward a more far-reaching issue: if meaningful input tokens improve morphological

capabilities (or else semantically deteriorated input tokens decrease them), should this not be a guiding principle for designing tokenization algorithms in the first place? This idea motivates the research presented in the final two chapters.

8.1.4 Improving Tokenization for Language Models

Chapters 6 and 7 both examine what effects making the input tokenization more congruent with morphology has on the general performance of language models (e.g., on topic classification or sentiment analysis). While chapter 6 imposes a morphological gold segmentation taken from a morphological resource on the language model, chapter 7 develops a recursive algorithm that, for a given derivative, automatically finds tokens that cover the core of its lexical semantics. Both methods substantially improve performance, which underscores that meaningful tokens are relevant not only for predicting derivation, but for modeling language more generally.

8.2 Discussion

8.2.1 Relevance for Linguistics

The first part of the thesis shows that new methods and datasets from NLP make it possible to predict derivational morphology to a larger extent than previously thought. What is the relevance of this finding for the theory of morphology? In the following, I want to highlight three aspects in particular: its implications for the inflection-derivation split, the notion of morphological productivity, and the mechanisms underlying the acquisition and processing of derivational morphology.

Linguistics traditionally divides morphology into inflection and derivation. As has been pointed out in the introduction (§1.2), many scholars assume that the distinction between inflection and derivation is a continuum rather than a dichotomy (e.g., Bochner, 1992; Haspelmath and Sims, 2010; ten Hacken, 2014), with predictability being seen as one dimension on this continuum, potentially even the most important one (Bauer, 2019). Crucially, the results presented in the first part of this thesis cast doubt on the assumption that predictability constitutes a distinguishing factor between inflection and derivation

on its own: explicitly including dimensions previously left aside due to methodological challenges (e.g., lexical semantics) substantially improves the predictability of derivational morphology. This suggests that the difference in predictability between inflection and derivation is *methodological* in nature, not *intrinsic* to morphology. In other words, it is not the case that derivation is *per se* less predictable than inflection, but capturing the factors that are key for predicting derivation requires methods that are—or were—not available. Thus, the findings of this thesis indicate that predictability does not hold up as a clear distinguishing factor between inflection and derivation.

The results presented in the first part of the thesis are also relevant to the notion of morphological productivity (§2.1): while prior work has focused on the productivity of derivational affixes (e.g., Baayen and Lieber, 1991; Baayen, 1992, 1993; Bauer, 2001; Baayen, 2005), my analyses show that bases possess a distinct productivity profile as well, which interacts with the productivity of derivational affixes. This is reflected on the level of sociolinguistics: while prior work has found that the productivity of derivational affixes is characterized by a substantial degree of extralinguistic variability (Plag et al., 1999; Säily, 2011, 2014, 2016; Needle and Pierrehumbert, 2018), my analyses highlight that the productivity of bases exhibits a similar form of extralinguistic variability. Taken together, this suggests that the probability that a derivative is formed should be seen as a joint function of the productivity of the base and the derivational affix.

Finally, the results presented in the first part of the thesis have implications for the debate about what mechanisms underlie the acquisition and processing of derivational morphology (§2.1): the systems that achieve state-of-the-art results are neural networks such as language models that do not maintain any symbolic representations, suggesting that it is not necessary to assume the existence of symbolic rules (e.g., Embick and Marantz, 2008) for explaining derivational morphology. While neural networks do not strictly constitute analogical models, it is important to note that their functioning heavily depends on similarity relations between input items (Plunkett and Marchman, 1991), which is at least conceptually similar to analogical models. In this sense, my findings are complementary to recent research showing that analogical models predict derivational morphology best (Arndt-Lappe, 2014).

8.2.2 Relevance for NLP

The second part of the thesis shows that a morphologically valid tokenization strategy can improve the performance of language models. The fact that language models are the backbone of most current NLP systems means that this finding is of potential value for various subfields of NLP, especially those that deal with morphologically complex language (e.g., NLP for science, NLP for medicine). However, there has been a lot of progress in NLP compared to the language models examined in this thesis—are the results relevant even for the latest and most capable language models? In the following, I will discuss more recent work and argue that morphology indeed continues to be important for tokenization, albeit in nuanced ways.

On the one hand, extending the research program of this thesis, recent studies have examined how morphological properties of the tokenizer affect the performance of the latest language models. Particularly relevant is the study by Weissweiler et al. (2023), who test the morphological generalization capabilities of GPT-3.5 (Ouyang et al., 2022) and GPT-4 (OpenAI et al., 2023), finding that these language models are affected less by tokenization than smaller language models. This suggests that scaling (i.e., increasing the model size and the training data) can to a certain extent offset the bottleneck imposed by a morphologically suboptimal tokenization—put differently, while smaller language models require the inductive bias of a morphologically informed tokenization, this seems to be less important for larger language models. Further research is needed to confirm and further explore this hypothesis.

On the other hand, taking a more global perspective, it is important to note that for many languages of the world, there is not enough text data that would allow them to benefit from this hypothetical scaling effect. Currently, tokenizers are especially suboptimal for low-resource languages, many of which possess a rich morphology, and this is reflected by a worse language model performance (Ahia et al., 2023). Thus, to build language technology that does not treat speakers of certain languages unfairly, it continues to be vital to take morphology into account.

8.3 Open Challenges

There are still many open challenges that could not be addressed within the scope of this thesis. In the following, I will briefly discuss the most prominent ones.

Regarding the prediction of derivational morphology (i.e., the first set of papers), while the performance of the proposed models is generally better than baselines from prior work, there still is a gap compared to perfect performance. One of the factors responsible for this, which has come up repeatedly in the qualitative and quantitative analyses (cf. §5.5, §6.3), is the fact that current methods for meaning representation in NLP (both word embeddings and language models) struggle to encode scalar semantics (Garí Soler and Apidianaki, 2021; Lorge and Pierrehumbert, 2023). In the future, the development of novel ways to train NLP systems such as grounding (Bisk et al., 2020) could lead to vector spaces in which scalar meanings are represented in more direct ways, thus potentially alleviating this issue. A second factor that might play a role is the way in which extralinguistic factors are modeled in this thesis: extralinguistic factors are included as discrete variables (e.g., subreddits), whereas in reality these factors are continuous. For example, not all users posting to a subreddit have the same vocabulary, and it is hence likely that there are derivational differences even within one and the same subreddit. One promising approach to take this into account would be to represent the structure of users not as a set of subreddits but rather as a social network, where similarities between users are encoded by their distance in the social network (McPherson et al., 2001). Such an approach would make it possible to include extralinguistic factors via graph encoders, a method that is becoming increasingly popular in NLP (e.g., del Tredici et al., 2019b; Mishra et al., 2019; Hofmann et al., 2021).

Regarding tokenization, while the proposed methods substantially improve performance on the examined tasks, there are linguistic phenomena that they cannot capture, either. For example, there are effects of morphological structure ambiguity (Bauer, 2019): the derivative *undoable* can either have the structure *un-doable* (i.e., 'cannot be done') or the structure *undo-able* (i.e., 'can be undone'). Ideally, these two structures would correspond to different tokenizations, which is not possible with any discussed

tokenization strategy. Being able to model such effects would require tokenization to be *contextualized* (i.e., depend on the immediate syntagmatic context)—developing such methods is a promising avenue for future research.

8.4 Wider Implications

The research presented in this thesis shows that NLP and deep learning more generally can greatly benefit linguistic research, a view that is still contested by many scholars in linguistics (e.g., Katzir, 2023), including Noam Chomsky, who said in 2021 (cited after Mahowald et al., 2023):

> "We have to ask here a certain question: is [deep learning] engineering or is it science? [...] On engineering grounds, it's kind of worth having, like a bulldozer. Does it tell you anything about human language? Zero."

Contrasting with such a pessimistic view, this thesis adds to the growing body of work arguing for a closer integration of computational approaches into core areas of linguistics (e.g., Linzen, 2019; Pater, 2019; Baroni, 2022).

On the other hand, at the moment perhaps even more widespread than the view that NLP cannot benefit linguistics might be the view that linguistics cannot benefit NLP, which is neatly encapsulated by the infamous dictum that "Every time we fire a phonetician/linguist, the performance of our system goes up" (Moore, 2005). The findings of this thesis show that even today, linguistic insights continue to be relevant for the development of human language technology.

Bibliography

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. Do all languages cost the same? Tokenization in the era of commercial language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2023*, 2023.

Adam Albright and Bruce Hayes. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, 90(2):119–161, 2003.

Maria Alegre and Peter Gordon. Rule-based versus associative processes in derivational morphology. *Brain and Language*, 68(1):347–354, 1999.

Eduardo G. Altmann, Janet B. Pierrehumbert, and Adilson E. Motter. Niche as a determinant of word fate in online groups. *PloS ONE*, 6(5):e19009, 2011.

Ben Ambridge. Against stored abstractions: A radical exemplar model of language acquisition. *First Language*, 40(5-6):509–559, 2020.

Samaneh Aminikhanghahi and Diane J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367, 2017.

Stephen R. Anderson. *A-Morphous Morphology*. Cambridge University Press, Cambridge, UK, 1992.

Sabine Arndt-Lappe. Analogy in suffix rivalry: The case of English -ity and -ness. *English Language & Linguistics*, 18(3):497–548, 2014.

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

Ehsaneddin Asgari, Christoph Ringlstetter, and Hinrich Schütze. EmbLexChange at SemEval-2020 task 1: Unsupervised embedding-based detection of lexical semantic changes. In *International Workshop on Semantic Evaluation (SemEval) 2020*, 2020.

Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. Probabilistic FastText for multi-sense word embeddings. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*, 2018.

R. Harald Baayen. Quantitative aspects of morphological productivity. In Geert Booij and Jaap van Marle, editors, *Yearbook of Morphology 1991*, pages 109–149. Kluwer, Dordrecht, 1992.

R. Harald Baayen. On frequency, transparency and productivity. In Geert Booij and Jaap van Marle, editors, *Yearbook of Morphology 1992*, pages 181–208. Kluwer, Dordrecht, 1993.

R. Harald Baayen. Morphological productivity. In Reinhard Köhler, Gabriel Altmann, and Rajmund G. Piotrowski, editors, *Quantitative Linguistics: An International Handbook*, pages 243–255. De Gruyter, Berlin, 2005.

R. Harald Baayen and Rochelle Lieber. Productivity and English derivation: A corpus-based study. *Linguistics*, 29(5), 1991.

R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers. *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, Philadelphia, PA, 1995.

Matthew Baerman, Greville G. Corbett, and and Dunstan Brown, editors. *Defective Paradigms: Missing Forms and What They Tell Us*. Oxford University Press, Oxford, UK, 2010.

Robert Bamler and Stephan Mandt. Dynamic word embeddings. In *International Conference on Machine Learning (ICML) 34*, 2017.

Marco Baroni. On the proper role of linguistically-oriented deep net analysis in linguistic theorizing. In *arXiv 2106.08694*. 2022.

Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2010*, 2010.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Annual Meeting of the Association for Computational Linguistics (ACL) 52*, 2014.

Laurie Bauer. *Introducing Linguistic Morphology*. Edinburgh University Press, Edinburgh, UK, 1988.

Laurie Bauer. *Morphological Productivity*. Cambridge University Press, Cambridge, UK, 2001.

Laurie Bauer. *Rethinking Morphology*. Edinburgh University Press, Edinburgh, UK, 2019.

Laurie Bauer, Rochelle Lieber, and Ingo Plag. *The Oxford Reference Guide to English Morphology*. Oxford University Press, Oxford, UK, 2013.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. The pushshift Reddit dataset. In *International AAAI Conference on Weblogs and Social Media (ICWSM) 14*, 2020.

Raymond Bertram, Matti Laine, R. Harald Baayen, Robert Schreuder, and Jukka Hyönä. Affixal homonymy triggers full-form storage, even with inflected words, even in a morphologically rich language. *Cognition*, 74:B13–B25, 2000.

Elisabeth Beyersmann, Johannes C. Ziegler, and Jonathan Grainger. Differences in the processing of prefixes and suffixes revealed by a letter-search task. *Scientific Studies of Reading*, 19(5):360–373, 2015.

Parminder Bhatia, Robert Guthrie, and Jacob Eisenstein. Morphological priors for probabilistic neural word embeddings. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2016*, 2016.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Harry Bochner. *Simplicity in Generative Morphology*. De Gruyter, Berlin, 1992.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Gemma Boleda. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234, 2020.

Olivier Bonami and Jana Strnadová. Paradigm structure and predictability in derivational morphology. *Morphology*, 29(2):167–197, 2019.

Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

Jan A. Botha and Phil Blunsom. Compositional morphology for word representations and language modelling. In *International Conference on Machine Learning (ICML) 31*, 2014.

Laurel J. Brinton and Elizabeth C. Traugott. *Lexicalization and Language Change*. Cambridge University Press, Cambridge, UK, 2005.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *arXiv 2005.14165*. 2020.

Kris Cao and Laura Rimell. You should evaluate your language model on marginal likelihood over tokenisations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2021*, 2021.

Steve Chandler. The analogical modeling of linguistic categories. *Language and Cognition*, 9(1):52–87, 2017.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

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

Noam Chomsky and Morris Halle. *The Sound Pattern of English*. Harper & Row, New York, NY, 1968.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. CANINE: Pre-training an efficient tokenization-free encoder for language representation. In *arXiv 2103.06874*. 2021.

Ryan Cotterell, Hinrich Schütze, and Jason Eisner. Morphological smoothing and extrapolation of word embeddings. In *Annual Meeting of the Association for Computational Linguistics (ACL) 54*, 2016.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. Paradigm completion for derivational morphology. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017*, 2017.

Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON) 6*, 2002.

David Crystal. *Language and the Internet*. Cambridge University Press, Cambridge, UK, 2004.

Walter Daelemans and Antal van den Bosch. *Memory-Based Language Processing*. Cambridge University Press, Cambridge, UK, 2005.

Robert Daland, Andrea D. Sims, and Janet Pierrehumbert. Much ado about nothing: A social network model of Russian paradigmatic gaps. In *Annual Meeting of the Association of Computational Linguistics (ACL) 45*, 2007.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. Understanding and improving morphological learning in the neural machine translation decoder. In *International Joint Conference on Natural Language Processing (IJCNLP) 8*, 2017.

Srayan Datta, Chanda Phelan, and Eytan Adar. Identifying misaligned inter-group links and communities. *Proceedings of the ACM on Human-Computer Interaction*, 1:1–23, 2017.

Scott Deerwester, Susan T. Dumais, George Furnas, Thomas Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

Marco del Tredici, Raquel Fernández, and Gemma Boleda. Short-term meaning shift: A distributional exploration. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2019*, 2019a.

Marco del Tredici, Diego Marcheggiani, Sabine Schulte im Walde, and Raquel Fernández. You shall know a user by the company it keeps: Dynamic representations for social media users in NLP. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2019*, 2019b.

Daniel Deutsch, John Hewitt, and Dan Roth. A distributional and orthographic aggregation model for English derivational morphology. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2019*, 2019.

Cícero N. dos Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *International Conference on Machine Learning (ICML) 31*, 2014.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. Time-out: Temporal referencing for robust modeling of lexical semantic change. In *Annual Meeting of the Association for Computational Linguistics (ACL) 57*, 2019.

David Embick and Alec Marantz. Architecture and blocking. *Linguistic Inquiry*, 39(1):1–53, 2008.

Katrin Erk. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653, 2012.

Hao Fang, Mari Ostendorf, Peter Baumann, and Janet B. Pierrehumbert. Exponential language modeling using morphological features and multi-task learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2410–2421, 2015.

Laurie Feldman. Beyond orthography and phonology: Differences between inflections and derivations. *Journal of Memory and Language*, 33(4):442–470, 1994.

John Rupert Firth. *Studies in Linguistic Analysis*. Blackwell, Oxford, UK, 1957.

Livio Gaeta. On the double nature of productivity in inflectional morphology. *Morphology*, 17(2):181–205, 2007.

Philip Gage. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38, 1994.

Aina Garí Soler and Marianna Apidianaki. Scalar adjective identification and multilingual ranking. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2021*, 2021.

Hélène Giraudo and Jonathan Grainger. On the role of derivational affixes in recognizing complex words: Evidence from masked priming. In R. Harald Baayen and Robert Schreuder, editors, *Morphological Structure in Language Processing*, pages 209–232. De Gruyter, Berlin, 2003.

Hongyu Gong, Suma Bhat, and Pramod Viswanath. Enriching word embeddings with temporal and spatial information. In *Conference on Computational Natural Language Learning (CoNLL) 24*, 2020.

Laura M. Gonnerman and Elaine S. Andersen. Graded semantic and phonological similarity effects in morphologically complex words. In Sabrina Bendjaballah, Wolfgang U. Dressler, Oskar E. Pfeiffer, and Maria D. Voeikova, editors, *Morphology 2000*, pages 137–148. John Benjamins, Amsterdam, 2002.

William Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In *Annual Meeting of the Association for Computational Linguistics (ACL) 54*, 2016.

Martin Haspelmath and Andrea D. Sims. *Understanding Morphology*. Routledge, New York, NY, 2010.

Jennifer Hay. Lexical frequency in morphology: Is everything relative? *Linguistics*, 39(6):1041–1070, 2001.

Jennifer Hay and R. Harald Baayen. Shifting paradigms: Gradient structure in morphology. *Trends in Cognitive Sciences*, 9(7):342–348, 2005.

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. Dynamic contextualized word embeddings. In *Annual Meeting of the Association for Computational Linguistics (ACL) 59*, 2021.

Valentin Hofmann, Hinrich Schütze, and Janet B. Pierrehumbert. The Reddit politosphere: A large-scale text and network resource of online political discourse. *International AAAI Conference on Weblogs and Social Media (ICWSM) 16*, 2022.

Jimin Hong, Taehee Kim, Hyesu Lim, and Jaegul Choo. AVocaDo: Strategy for adapting vocabulary to downstream domain. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2021*, 2021.

Abhilash Jain, Aku Rouhe, Stig-Arne Grönroos, and Mikko Kurimo. Finnish ASR with deep transformer models. In *Conference of the International Speech Communication Association (INTERSPEECH) 21*, 2020.

Roni Katzir. Why large language models are poor theories of human linguistic cognition. A reply to Piantadosi (2023), 2023.

Adam Kilgarriff and Gregory Grefenstette. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3):333–348, 2003.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. Temporal analysis of language through neural language models. In *Workshop on Language Technologies and Computational Social Science*, 2014.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-aware neural language models. In *Conference on Artificial Intelligence (AAAI) 30*, 2016.

Max Kisselew, Sebastian Padó, Alexis Palmer, and Jan Šnajder. Obtaining a better understanding of distributional models of German derivational morphology. In *International Conference on Computational Semantics (IWCS) 11*, 2015.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Annual Meeting of the Association for Computational Linguistics (ACL) 56*, 2018.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2018*, 2018.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Statistically significant detection of linguistic change. In *The Web Conference (WWW) 24*, 2015.

Srijan Kumar, William Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In *The Web Conference (WWW) 27*, 2018.

Victor Kuperman, Robert Schreuder, Raymond Bertram, and R. Harald Baayen. Reading of polymorphemic Dutch compounds: Towards a multiple route model of lexical processing. *Journal of Experimental Psychology: Human Perception and Performance*, 35(3):876–895, 2009.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. Diachronic word embeddings and semantic shifts: A survey. In *International Conference on Computational Linguistics (COLING) 27*, 2018.

Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Annual Meeting of the Association for Computational Linguistics (ACL) 51*, 2013.

Alessandro Lenci. Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31, 2008.

Rochelle Lieber. Theoretical issues in word formation. In Jenny Audring and Francesca Masini, editors, *The Oxford Handbook of Morphological Theory*, pages 34–55. Oxford University Press, Oxford, UK, 2019.

Tal Linzen. What can linguistics and deep learning contribute to each other? Response to Pater. *Language*, 95(1):e99–e108, 2019.

Jiayi Liu, Donghua Yang, Kaiqi Zhang, Hong Gao, and Jianzhong Li. Anomaly and change point detection for time series with concept drift. *World Wide Web*, 26(5):3229–3252, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. In *arXiv 1907.11692*. 2019.

Isabelle Lorge and Janet Pierrehumbert. Not wacky vs. definitely wacky: A study of scalar adverbs in pretrained language models. In *arXiv 2305.16426*. 2023.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. Better word representations with recursive neural networks for morphology. In *Conference on Computational Natural Language Learning (CoNLL) 17*, 2013.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. CharBERT: Character-aware pre-trained language model. In *International Conference on Computational Linguistics (COLING) 28*, 2020.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. Dissociating language and thought in large language models: A cognitive perspective. In *arXiv 2301.06627*. 2023.

Miller McPherson, Lynn Smith-Lovin, and James M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The NomBank project: An interim report. In *Frontiers in Corpus Annotation*, 2004.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS) 26*, 2013b.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2013*, 2013c.

Einat Minkov, Kristina Toutanova, and Hisami Suzuki. Generating complex morphology for machine translation. In *Annual Meeting of the Association of Computational Linguistics (ACL) 45*, 2007.

Pushkar Mishra, Marco del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. Abusive language detection with graph convolutional networks. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2019*, 2019.

Roger K. Moore. Results from a survey of attendees at ASRU 1997 and 2003. In *Conference of the International Speech Communication Association (INTERSPEECH) 6*, 2005.

J. Needle and J. Pierrehumbert. Gendered associations of English morphology. *Laboratory Phonology*, 9 (1), 2018.

Boris New, Marc Brysbaert, Juan Segui, Ludovic Ferrand, and Kathleen Rastle. The processing of singular and plural nouns in French and English. *Journal of Memory and Language*, 51(4):568–585, 2004.

Robert M. Nosofsky. Relations between exemplar-similarity and likelihood models of classification. *Journal of Mathematical Psychology*, 34(4):393–418, 1990.

Randal S. Olson and Zachary P. Neal. Navigating the massive world of Reddit: Using backbone networks to map user interests in social media. *PeerJ Computer Science*, 2015.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David

Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 technical report. In *arXiv 2303.08774*. 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *arXiv 2203.02155*. 2022.

Sebastian Padó, Aurélie Herbelot, Max Kisselew, and Jan Šnajder. Predictability of distributional semantics in derivational word formation. In *International Conference on Computational Linguistics (COLING) 26*, 2016.

Mark Pagel, Quentin Atkinson, and Andrew Meade. Frequency of word-use predicts rates of lexical evolution throughout Indo-European history. *Nature*, 449(7163):717–720, 2007.

Joe Pater. Generative linguistics and neural networks at 60: Foundation, friction, and fusion. *Language*, 95 (1):e41–e74, 2019.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2014*, 2014.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2018*, 2018.

Janet Pierrehumbert. The dynamic lexicon. In Abigail Cohn, Cécile Fougeron, and Marie Huffman, editors, *The Oxford Handbook of Laboratory Phonology*, pages 173–183. Oxford University Press, Oxford, UK, 2012.

Ingo Plag. *Morphological Productivity: Structural Constraints in English Derivation*. De Gruyter, Berlin, 1999.

Ingo Plag. *Word-Formation in English*. Cambridge University Press, Cambridge, UK, 2003.

Ingo Plag and Laura W. Balling. Derivational morphology: An integrative perspective on some fundamental questions. In Vito Pirrelli, Ingo Plag, and Wolfgang U. Dressler, editors, *Word Knowledge and Word Usage: A Cross-Disciplinary Guide to the Mental Lexicon*, pages 295–335. De Gruyter, Berlin, 2020.

Ingo Plag, Christiane Dalton-Puffer, and R. Harald Baayen. Morphological productivity across speech and writing. *English Language and Linguistics*, 3(2):209–228, 1999.

Kim Plunkett and Virginia Marchman. U-shaped learning and frequency effects in a multi-layered perception: Implications for child language acquisition. *Cognition*, 38(1):43–102, 1991.

Martin Pömsl and Roman Lyapin. CIRCE at SemEval-2020 task 1: Ensembling context-free and context-dependent word representations. In *International Workshop on Semantic Evaluation (SemEval) 2020*, 2020.

Ondřej Pražák, Pavel Přibáň, Stephen Taylor, and Jakub Sido. UWB at SemEval-2020 task 1: Lexical semantic change detection. In *International Workshop on Semantic Evaluation (SemEval) 2020*, 2020.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. BPE-dropout: Simple and effective subword regularization. In *Annual Meeting of the Association for Computational Linguistics (ACL) 58*, 2020.

Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. Co-learning of word representations and morpheme representations. In *International Conference on Computational Linguistics (COLING) 25*, 2014.

Péter Rácz, Janet B. Pierrehumbert, Jennifer Hay, and Viktória Papp. Morphological emergence. In Brian MacWhinney and William O'Grady, editors, *The Handbook of Language Emergence*, pages 123–146. Wiley, Hoboken, NJ, 2015.

Péter Rácz, Clay Beckner, Jennifer Hay, and Janet B. Pierrehumbert. Morphological convergence as on-line lexical analogy. *Language*, 96(4):735–770, 2020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Kathleen Rastle, Matthew H. Davis, and Boris New. The broth in my brother's brothel: Morpho-orthographic segmentation in visual word recognition. *Psychonomic Bulletin and Review*, 11(6): 1090–1098, 2004.

Brian Roark and Richard Sproat. *Computational Approaches to Morphology and Syntax*. Oxford University Press, Oxford, UK, 2007.

Alex Rosenfeld and Katrin Erk. Deep neural models of semantic shift. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2018*, pages 474–484, 2018.

Maja Rudolph and David Blei. Dynamic embeddings for language evolution. In *The Web Conference (WWW) 27*, 2018.

David E. Rumelhart and James L. McClelland. On learning the past tenses of English verbs, 1985.

Vin Sachidananda, Jason S. Kessler, and Yi-An Lai. Efficient domain adaptation of language models via adaptive tokenization. In *Workshop on Simple and Efficient Natural Language Processing (SustaiNLP) 2*, 2021.

Tanja Säily. Variation in morphological productivity in the BNC: Sociolinguistic and methodological considerations. *Corpus Linguistics and Linguistic Theory*, 7(1):119–141, 2011.

Tanja Säily. *Sociolinguistic Variation in English Derivational Productivity: Studies and Methods in Diachronic Corpus Linguistics*. Société Néophilologique, Helsinki, 2014.

Tanja Säily. Sociolinguistic variation in morphological productivity in eighteenth-century English. *Corpus Linguistics and Linguistic Theory*, 12(1):129–151, 2016.

Alexandre Salle and Aline Villavicencio. Incorporating subword information into matrix factorization word embeddings. In *Workshop on Subword/Character LEvel Models 2*, 2018.

Dominik Schlechtweg and Sabine Schulte im Walde. Simulating lexical semantic change from sense-annotated data. In *International Conference on the Evolution of Language (EvoLang) 13*, 2020.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. Diachronic usage relatedness (DURel): A framework for the annotation of lexical semantic change. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2018*, 2018.

Dominik Schlechtweg, Anna Hätty, Marco del Tredici, and Sabine Schulte im Walde. A wind of change: Detecting and evaluating lexical semantic change across times and domains. In *Annual Meeting of the Association for Computational Linguistics (ACL) 57*, 2019.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *International Workshop on Semantic Evaluation (SemEval) 2020*, 2020.

Mike Schuster and Kaisuke Nakajima. Japanese and Korean voice search. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 37*, 2012.

Hinrich Schütze. Dimensions of meaning. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC) 1992*, 1992a.

Hinrich Schütze. Word space. In *Advances in Neural Information Processing Systems (NIPS) 5*, 1992b.

Mark S. Seidenberg and Laura M. Gonnerman. Explaining derivational morphology as the convergence of codes. *Trends in Cognitive Sciences*, 4(9):353–361, 2000.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Annual Meeting of the Association for Computational Linguistics (ACL) 54*, 2016.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. SHUOWEN-JIEZI: Linguistically informed tokenizers for Chinese language model pretraining. In *arXiv 2106.00400*. 2021.

Royal Skousen. *Analogical Modeling of Language*. Kluwer, Dordrecht, 1989.

Royal Skousen. *Analogy and Structure*. Kluwer, Dordrecht, 1992.

Ingrid Sonnenstuhl and Axel Huth. Processing and representation of German -n plurals: A dual mechanism approach. *Brain and Language*, 81(1-3):276–290, 2002.

Ian Stewart and Jacob Eisenstein. Making "fetch" happen: The influence of social and linguistic context on nonstandard word growth and decline. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2018*, 2018.

Gregory Stump. Inflection. In Andrew Spencer and Arnold M. Zwicky, editors, *The Handbook of Morphology*, pages 13–43. Wiley, Hoboken, NJ, 2001.

Gregory Stump. *Inflectional Paradigms: Content and Form at the Syntax-Morphology Interface*. Cambridge University Press, Cambridge, UK, 2015.

Marcus Taft. Interactive-activation as a framework for understanding morphological processing. *Language and Cognitive Processes*, 9(3):271–294, 1994.

Marcus Taft and Kenneth I. Forster. Lexical storage and retrieval of prefixed words. *Journal of Verbal Learning and Verbal Behavior*, 14:638–647, 1975.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. Survey of computational approaches to lexical semantic change detection. In *arXiv 1811.06278*. 2018.

Samson Tan, Shafiq Joty, Lav R. Varshney, and Min-Yen Kan. Mind your inflections! Improving NLP for non-standard Englishes with base-inflection encoding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*, 2020.

Pius ten Hacken. Delineating derivation and inflection. In Rochelle Lieber and Pavol Štekauer, editors, *The Oxford Handbook of Derivational Morphology*, pages 10–25. Oxford University Press, Oxford, UK, 2014.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.

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

Lyan Verwimp, Joris Pelemans, Hugo van Hamme, and Patrick Wambacq. Character-word LSTM language models. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL) 15*, 2017.

Sami Virpioja, Minna Lehtonen, Annika Hultén, Henna Kivikari, Riitta Salmelin, and Krista Lagus. Using statistical models of morphology in the search for optimal units of representation in the human mental lexicon. *Cognitive Science*, 42(3):939–973, 2018.

Ivan Vulić, Nikola Mrksic, Roi Reichart, Diarmuid Ó. Séaghdha, Steve Young, and Anna Korhonen. Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. In *Annual Meeting of the Association for Computational Linguistics (ACL) 55*, 2017.

Ivan Vulić, Edoardo M. Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. Probing pretrained language models for lexical semantics. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*, 2020.

Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin, and Trevor Cohn. Context-aware prediction of derivational word-forms. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL) 15*, 2017.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schütze, Kemal Oflazer, and David Mortensen. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2023*, 2023.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le V, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv 1609.08144*. 2016.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. In *arXiv 2105.13626*. 2021.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, 2019.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. Dynamic word embeddings for evolving semantic discovery. In *International Conference on Web Search and Data Mining (WSDM) 11*, 2018.

Xinsong Zhang, Pengshuai Li, and Hang Li. AMBERT: A pre-trained language model with multi-grained tokenization. In *Findings of the Association for Computational Linguistics: ACL 2021*, 2021.

Authorship Statements

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

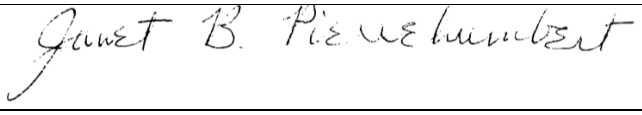
Title of Paper	A Graph Auto-encoder Model of Derivational Morphology
Publication Status	xPublished □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Valentin Hofmann, Hinrich Schütze, and Janet Pierrehumbert. 2020. A Graph Auto-encoder Model of Derivational Morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1127–1138.

Student Confirmation

Student Name:	Valentin Hofmann		
Contribution to the Paper	VH developed the initial idea, ran the experiments, and conducted the analysis. HS and JP advised. VH wrote the paper text, which was then edited by HS and JP. Figures were generated by VH.		
Signature		Date	04/10/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Janet Pierrehumbert		
Supervisor comments The description of the contributions is accurate.		
	Date	04/10/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

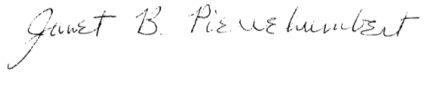
Title of Paper	Predicting the Growth of Morphological Families from Social and Linguistic Factors
Publication Status	xPublished □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020. Predicting the Growth of Morphological Families from Social and Linguistic Factors. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7273–7283.

Student Confirmation

Student Name:	Valentin Hofmann		
Contribution to the Paper	The initial idea emerged in discussions between VH and JP. VH operationalized the idea, ran the experiments, and conducted the analysis. JP and HS advised. VH wrote the paper text, which was then edited by JP and HS. Figures were generated by VH.		
Signature		Date	04/10/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Janet Pierrehumbert		
Supervisor comments The description of the contributions is accurate.		
	Date	04/10/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

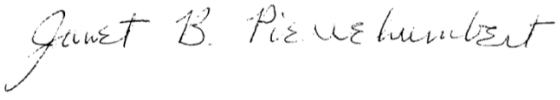
Title of Paper	DagoBERT: Generating Derivational Morphology with a Pretrained Language Model
Publication Status	xPublished □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020. DagoBERT: Generating Derivational Morphology with a Pretrained Language Model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3848–3861.

Student Confirmation

Student Name:	Valentin Hofmann		
Contribution to the Paper	VH developed the initial idea, ran the experiments, and conducted the analysis. JP and HS advised. VH wrote the paper text, which was then edited by JP and HS. Figures were generated by VH.		
Signature		Date	04/10/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Janet Pierrehumbert		
Supervisor comments The description of the contributions is accurate.		
Signature 	Date	04/10/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

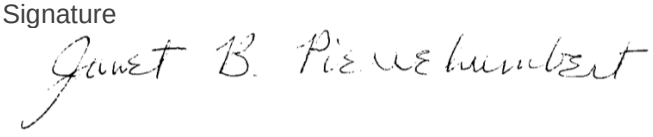
Title of Paper	Superbizarre Is Not Superb: Derivational Morphology Improves BERT's Interpretation of Complex Words
Publication Status	xPublished □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre Is Not Superb: Derivational Morphology Improves BERT's Interpretation of Complex Words. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 3594–3608.

Student Confirmation

Student Name:	Valentin Hofmann		
Contribution to the Paper	JP drew VH's attention to the literature on unsupervised learning of morphological systems for use in natural language processing algorithms. VH developed the connection to the codebooks currently used in large language models, ran the experiments, and conducted the analysis. JP and HS advised. VH wrote the paper text, which was then edited by JP and HS. Figures were generated by VH.		
Signature		Date	04/10/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Janet Pierrehumbert		
Supervisor comments The description of the contributions is accurate.		
Signature 	Date	04/10/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

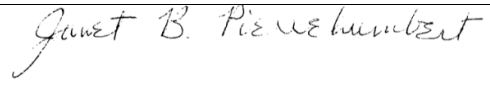
Title of Paper	An Embarrassingly Simple Method to Mitigate Undesirable Properties of Pretrained Language Model Tokenizers
Publication Status	xPublished □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An Embarrassingly Simple Method to Mitigate Undesirable Properties of Pretrained Language Model Tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 385–393.

Student Confirmation

Student Name:	Valentin Hofmann		
Contribution to the Paper	VH developed the initial idea, ran the experiments, and conducted the analysis. HS and JP advised. VH wrote the paper text, which was then edited by HS and JP. Figures were generated by VH.		
Signature		Date	04/10/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Janet Pierrehumbert		
Supervisor comments The description of the contributions is accurate.		
	Date	04/10/2023

This completed form should be included in the thesis, at the end of the relevant chapter.