Learning Computational Models of Non-Standard Language

Maria Ryskina

CMU-LTI-22-019

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15123
www.lti.cs.cmu.edu

Thesis Committee:
Matthew R. Gormley, Carnegie Mellon University (Co-chair)
Eduard Hovy, Carnegie Mellon University (Co-chair)
Taylor Berg-Kirkpatrick, University of California, San Diego (Co-chair)
David R. Mortensen, Carnegie Mellon University
Roger P. Levy, Massachusetts Institute of Technology

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies.

To my grandfathers, Mikhail Ryskin (1938–2017) and Iosif Cherniavsky (1932–2021).
May their memory be a blessing.

Abstract

Non-standard language such as novel words or creative spellings of existing ones often occurs in natural text corpora, posing significant challenges for natural language processing (NLP) models. While humans can successfully infer the meaning communicated in such non-standard ways, NLP models largely discard linguistic innovation as noise, ignoring its fundamentally non-random nature and losing valuable context. In this thesis, we focus on computational modeling of such creative phenomena, aiming to both improve the automatic processing of non-standardized text data and to learn more about the linguistic and cognitive factors that allow humans to produce and understand novel linguistic items.

We present empirical studies of several phenomena under the umbrella of non-standard language, characterized in terms of different linguistic units (orthographic, morphological, or lexical) and considered at different levels of granularity (from individual users to entire dialects or languages). First, we show how idiosyncratic spelling preferences reveal information about the user, with an application to the bibliographic task of identifying typesetters of historical printed documents. Second, we discuss the common patterns in user-specific orthographies and demonstrate that incorporating these patterns helps with unsupervised conversion of idiosyncratically romanized text into the conventional orthography of the language. Third, we consider word emergence in a dialect or language as a whole and, in two diachronic corpora studies, model the language-internal and language-external factors that drive it. Finally, we look at how continuous emergence of novel words is reconciled with the existing system of morphological rules, focusing on generalization to unseen lemmas in morphological inflection in several languages.

Acknowledgments

This thesis would not have been possible without my fantastic team of advisors: Taylor Berg-Kirkpatrick, Matt Gormley, and Ed Hovy. Taylor, my first Ph.D. advisor, has stood by me through all the highs and lows of graduate school; he helped me see myself as a researcher and showed me how powerful excitement and curiosity about one's own work can be. Matt taught me to stay grounded and prioritize my own happiness and well-being, both at work and outside of it: I can only hope to someday become this kind of caring and thoughtful mentor. Discussions with Ed were crucial for developing my own high-level research vision: he taught me to not be afraid of asking big questions and to better understand our field and academia more generally. Thank you all for your patient guidance while I took the time to figure out my own path.

I also want to thank my committee members, David Mortensen and Roger Levy. David has taught me most of what I know about linguistics: his support, open-mindedness, and great skill for making linguistics accessible to non-linguists have empowered me to pursue an interdisciplinary career. Roger's work has gotten me interested in studying language in the mind, and his enthusiasm and encouragement inspired me to go ahead with it.

I have learned so much from working together with many other outstanding researchers: Ella Rabinovich, Yulia Tsvetkov, Hannah Alpert-Abrams, Dan Garrette, Vivek Kulkarni, Abhilasha Ravichander, Siddharth Dalmia, Alan W Black, Janet Pierrehumbert, Valentin Hofmann, Andrew Runge, Lee Branstetter, and the CMU AIDA/OPERA team. I had a great time interning at DiDi Labs and am grateful to all my teammates there, most of all Kevin Knight, who encouraged me to pursue the weirdest of ideas. Exchanges with many more people in the field were key for contextualizing my ideas—special thanks to Kyle Gorman, Brian Roark, Badr Abdullah, and Omer Goldman.

Participation in big interdisciplinary initiatives has greatly broadened my horizons: in particular, I would like to thank the UniMorph contributors (especially Ekaterina Vylomova, Ryan Cotterell, Eleanor Chodroff, Tiago Pimentel, Jonathan North Washington, and Francis Tyers), and the Diverse Intelligences Summer Institute.

The Language Technologies Institute (LTI) has been my home for the last six years. I am grateful to all the faculty members who have supported me through advice, mentoring, and teaching: in particular, Anatole Gershman, Bob Frederking,

Lori Levin, Graham Neubig, and LP Morency. I also want to thank the administrative staff at the School of Computer Science for taking care of me throughout the program, in particular Stacey Young and Laura Winter.

Being a member of three research groups, I have had the privilege of being surrounded by many wonderful labmates: Junxian He, Kartik Goyal, John Wieting, Daniel Spokoyny, Nikita Srivatsan, Si Wu, Fatemeh Mireshghallah, Volkan Cirik, Varun Gangal, Dheeraj Rajagopal, Amanda Bertsch, Jed Yang, and many others. Fellow students and faculty of the LTI Linguistics Lab also deserve a special note.

A heartfelt thank you to everyone whose friendship made these six years in LTI so memorable: Eva Spiliopoulou, Shruti Palaskar, Chirag Nagpal, Paul Michel, Khyathi Chandu, Aakanksha Naik, Hai Pham, Chaitanya Malaviya, Cindy Wang, Shuyan Zhou, Anjalie Field, Alex Wilf, Benjamin Elizalde, and Leonie Weissweiler, among many others. To my "Friendsgiving" crew—Shruti Rijhwani, Sid Dalmia, Abhilasha Ravichander, Rajat Kulshreshtha, Deepak Gopinath, Samridhi Choudhary—I can't thank you enough, I would never have gotten this far without having you by my side.

Thanks to the communities of Dor Hadash and Dor Ha'Ba for welcoming me so warmly and for reminding me what matters most—you truly are my *chevra*. Thanks to everyone at CMU C#: making music with you was a much-needed break. Thanks to all the folks at Queer in AI, especially William Agnew, Arjun Subramonian, Luca Soldaini, and Pranav A, for teaching me so much about the world and my place in it.

I am infinitely grateful to my parents, Nikita and Liubov, who have always worked so hard to give me every opportunity to succeed. Finally, none of this would have been possible without the unwavering support of my husband Stas: thank you for being there for me every step of the way. Your proposal is the one I will always accept with no revisions.

Contents

Chapter 1

Introduction

> *Language is the ultimate participatory democracy. To put it in technological terms, language is humanity's most spectacular open source project.*
>
> — Gretchen McCulloch, *Because Internet*

Variation is a natural property of human languages—although the need for communicative efficiency imposes its constraints on linguistic expression, the same words can be spoken, signed, or written in many different ways while still conveying the same literal meaning. It is not only an inalienable property of languages with low degrees of standardization, but also an instrument of creative expression, challenging the prescriptivist norms, and signaling social and situational meaning, such as one's regional identity or the level of formality of the conversation (Jaffe and Walton, 2000; Parrish, 2021; Nguyen et al., 2021). Although modern Natural Language Processing (NLP) methods largely omit the sociolinguistic aspect, focusing instead on distilling the conventional meaning from these varied expressions, we find that they are still unable to handle the full scope of linguistic innovation found in user-generated content such as social media data. Spellings that diverge from the dictionary-attested ones (*e.g.* 2nite for tonight or $p34k for speak) or novel blends of existing lexemes (*e.g.* procrastibaker, a combination of procrastinate and baker; Pinter et al., 2020) are most often discarded by NLP systems as random noise, along with artifacts like typos or data corruption. However, these creative spelling and word choices are fundamentally non-random: they are intentional, and their intended meaning can be deduced by other language users.

In this thesis, we focus on linguistic variation in written texts, which can manifest on multiple levels: orthographic (non-standard spellings), morphological (non-standard inflections), lexical (novel words), and various combinations of those. The larger goal behind this research is to study the shared foundation that allows language users to successfully communicate the intended

meaning in such creative acts. This common background could include, for example, phonological knowledge (*e.g.* that nite and night would be pronounced the same way) or shared perception of similarity between symbols (*e.g.* that the digit 0 resembles the character o). In a more applied sense, we aim to discover and model the language-internal and language-external constraints that shape linguistic variation and drive language change, and to build them directly into NLP applications, enabling better processing of non-standard linguistic items.

We also consider linguistic variation at different levels of social granularity. We start by modeling idiosyncratic choices of individual users, *i.e.* people directly involved in the process of rendering the text as a sequence of characters (Chapter 2). We than show that *user-level* decisions are not independent from one another but form more general patterns, which are in turn grounded in shared *group-level* perceptions of similarity or relatedness between linguistic units (Chapter 3). Besides grouping users by their preferences, we can also treat all of them as a single group for a *lect-level* (focused on a particular language, dialect, or sociolect as a whole) analysis of language evolution, using diachronic corpora as a proxy (Chapter 4, Chapter 5). Finally, since large-scale language change is driven by the linguistic practices of each participating user, we draw a connection between user-level acceptability judgments and lect-level survival of word forms, focusing specifically on morphological inflection (Chapter 6).

The detailed outline of the contributions of this thesis is presented below:

- In **Chapter 2**, we describe our work on compositor attribution in the First Folio of Shakespeare (Ryskina et al., 2017). This early modern document was typeset on a printing press by a group of workers (compositors), who injected their own spelling preferences into the text: for example, the spellings dear, deare, and deere are used interchangeably throughout the book, as all three were permitted by the non-standardized orthography of the time. Historical bibliography scholars have used these spelling choices to identify which pages were set by the same worker, relying on the assumption that each compositor was consistent in their preferences. We propose an automated approach to solving this problem and show that it can successfully reproduce the bibliographers' judgments about the number of compositors involved in the production of the book. Going beyond the spelling choices for specific words, we extend our analysis to higher-level patterns, such as preferring word-final -ie over -y (ladie, prettie, ...), and our experiments show that adding these patterns to the compositor 'fingerprint' improves the quality of typesetter attribution.
- **Chapter 3** shifts the focus from the individual idiosyncrasies to what these user-specific styles have in common. In order for the content of the message to be understood by the reader, its written representation needs to be grounded in certain perceptions of representational similarity that the writer shares with the intended audience. In this chapter, we present our work on informal romanization, an idiosyncratic cipher-like process of

writing non-Latin-script languages in Latin alphabet, *e.g.* rendering Cyrillic хорошо as xorosho (Ryskina et al., 2020a). Informal romanization is mostly used on social media and does not have fixed rules, so character substitution choices vary between users, but there are higher-level regularities to this variation. In practice, most characters are either replaced with similar-looking ones (Cyrillic х → Latin x) or similar-'sounding' ones (Cyrillic ш → Latin sh). We propose an unsupervised model for converting Latin-encoded text into native orthography, and our experiments demonstrate that bootstrapping the model with the notions of type-level phonetic and visual similarity between characters provides a strong training signal in the absence of supervision. We also quantitatively and qualitatively compare our finite-state model with an unsupervised neural architecture and explore several combinations of the two model classes (Ryskina et al., 2021).

- **Chapter 4** views language change on a larger scale and considers the intra- and extra-linguistic forces that drive it. This chapter presents our work on tracking neology, or new word emergence, in a large diachronic corpus of American English (Ryskina et al., 2020b). We investigate the roles of two competing factors, which we interpret as *supply* and *demand*. The supply hypothesis suggests that new words are more likely to appear in sparser areas of the semantic space, where they are less likely to be blocked by existing synonyms. The demand hypothesis proposes that words emerge more often within domains of growing importance: as a particular area of discourse becomes more prominent, the community starts to both discuss it more and to invent more novel concepts—and, in turn, to come up with more novel words to express them. Operationalizing the supply and demand factors under the distributional semantics paradigm (as a neighborhood's density and its word frequency growth rate), we find them both predictive of word emergence, but the demand factor is revealed to be more significant.
- **Chapter 5** extends the methodology introduced in the previous chapter and applies it to social media, where the data allows for tracking linguistic innovation on a much finer-grained time scale. Testing the same supply- and demand-driven neology hypotheses on a new corpus collected from Twitter, we reproduce the general trends observed in our earlier study of literary corpora: neologisms tend to appear in the sparser ares of the semantic space and in areas where existing words grow in popularity more rapidly. However, we find that the relative importance of the two factors differs between the two data sources, suggesting that studying the richer computer-mediated communication data could help uncover tendencies in language change that might not have been visible in sparser historical data.
- In **Chapter 6**, we propose a lect-level study of generalization in morphological inflection, building on a classic user-level psycholinguistic experiment. In particular, we focus on inflection of novel words—a challenge that linguistic communities naturally face as new

lexical items enter their vocabularies. Productively using an emergent word in new contexts might require filling in the missing cells in its morphological paradigm: for example, to use the recently-emerged verb vax in the sentence I am fully ____ *[vaccinated]*, the user needs to come up with an appropriate past participle form. For computational models, the main prerequisite to human-like handling of neologisms is successfully generalizing to lemmas not seen in training; that itself has been shown to be challenging for the state-of-the-art morphological inflectors. We propose relying on analogy to guide inflection of novel lemmas, augmenting the input with an exemplar that is most similar to the input in terms of phonology, orthography, or semantics. We empirically compare the contributions of the different factors and find the phonologically and orthographically relevant exemplars most useful for generalization. Although we did not observe consistent improvements across the board from our proposed augmentation, we show that with better exemplar retrievers analogy-based approaches could generalize to new lemmas more successfully.

Part I

Non-Standard and Novel Orthographies

Chapter 2

Idiosyncratic Spellings as Typesetter Fingerprints

2.1 Introduction

Non-standardized orthography, where words can have multiple widely used spellings, poses challenges for automatic text processing because of its inherent variation. But the same variation also contains useful information that can provide clues about the provenance of the document—much like how in stylometry writing style is used to determine the author (Holmes, 1994; Hope, 1994; Juola, 2006; Koppel et al., 2009; Jockers and Witten, 2010), one can analyze the "spelling style" to identify the persons who had worked on the manuscript. In the study of historical printed documents which pre-date the current orthographic standards, individual spelling choices are a primary feature used for *compositor attribution*—clustering the printed pages by the individual (the *compositor*) who arranged the type on the printing press. These analyses, based on orthographic and visual clues, have traditionally been done by hand, but the efforts are painstaking due to the difficulty of manually recording these features.

In this chapter, we present an unsupervised model specifically designed for compositor attribution, incorporating both the textual and the visual sources of evidence traditionally used by bibliographers (Hinman, 1963; Taylor, 1981; Blayney, 1991). Our model jointly describes the patterns of variation both in orthography and in the whitespace between glyphs, allowing us to cluster pages by discovering patterns of similarity and difference. When applied to digital scans of historical printed documents, our model learns orthographic and whitespace preferences of

The work presented in this chapter was done in collaboration with Hannah Alpert-Abrams, Dan Garrette, and Taylor Berg-Kirkpatrick. We are grateful to Gabriel Egan for the remarks on the corresponding publication, which helped revise some content in this chapter.

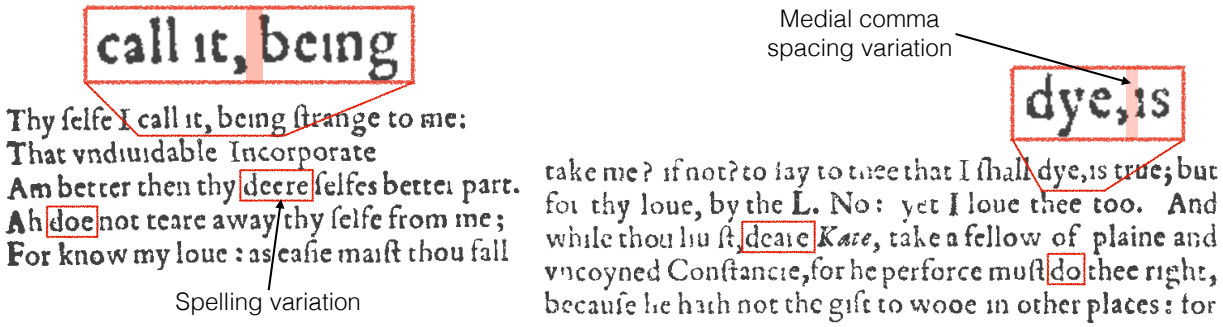


Figure 2.1: Two excerpts from the *First Folio*, taken from page 89 of *Comedies* (left) and page 93 of *Histories* (right). The compositor of the left page tended to use the spellings doe and deere, while the compositor for the right page used spellings do and deare, indicating that these pages were likely set by different people. The varying width of the medial comma whitespace also distinguishes the typesetters.

Note: the excerpt shown on the right (chosen for illustration only) is prose, which introduces an additional confound as the compositors may have adjusted spacing and spelling to help justify a line. In our experiments, we restrict the whitespace pattern analysis exclusively to short lines of text in order to avoid this confound.

individual compositors and predicts groupings of pages set by the same compositor.[1] This is, to our knowledge, the first attempt to perform compositor attribution automatically. Prior work has proposed automatic approaches to authorship attribution—which is typically viewed as the supervised problem of identifying a particular author given samples of their writing. In contrast, compositor attribution lacks supervision because compositors are unknown and, in addition, focuses on different linguistic patterns. We explain spellings of words conditioned on word choice, not the word choice itself.

To evaluate our approach, we fit our model to digital scans of Shakespeare's *First Folio* (1623)—a document with well-established manual judgments of compositor attribution. We find that even when relying on noisy OCR transcriptions of textual content, our model predicts compositor attributions that agree with manual annotations 87% of the time, outperforming several simpler baselines. We also find that a version of our model that does not make assumptions about the number of compositors ends up reproducing the scholars' conclusions drawn from manual analyses. Our approach opens new possibilities for considering patterns across a larger vocabulary of words and at a higher visual resolution than has been possible historically, and we hope that tools like ours will enable scalable first-pass analysis in understudied domains as a complement

[1] The validity of compositor attribution has sparked an ongoing and heated debate among bibliographers (McKenzie, 1969, 1984; Rizvi, 2016); while some reject parts or all of this approach, it continues to be cited in authoritative bibliographical texts (Gaskell, 2007; Blayney, 1996). Without taking a position in this debate, we seek only to automate the methods that remain in use by particular bibliographers (Blayney, 1996; Burrows, 2013).

to humanistic studies of composition.

2.2 Background

In this chapter, we focus on modeling the same types of observations made by scholars and demonstrate that our findings agree with authoritative attributions. We use compositor studies of Shakespeare's *First Folio* to inform our approach, drawing on the methods proposed by Hinman (1963), Howard-Hill (1973), and Taylor (1981). Hinman's (1963) landmark study split the pages of the *First Folio* between five different compositors based on variations in spelling among three common words. Figure 2.1, for example, shows portions of two pages from the *First Folio* with different spelling variants for the words dear and do: one compositor used deere and doe, while the other used deare and do. Hinman relied on the assumption that each compositor was consistent in their preferences for the sake of convenience in the typesetting process (Blayney, 1991).[2] Subsequent studies looked at larger sets of words and more general orthographic preferences (*e.g.* the preference to terminate words with -ie instead of -y), leading to modifications of Hinman's original analysis (Howard-Hill, 1973; Taylor, 1981). In this chapter, we propose a probabilistic model designed to capture both word-specific preferences and general orthographic patterns (§2.3). To separate the effect of the compositor from the choices made by the author or editor, we condition on a modernized (collated) version of Shakespeare's text (§2.4.1).

Visual features, including typeface usage and whitespace layout, also inform compositor attribution. For example, the highlighted spacing in Figure 2.1 shows different choices after medial commas (commas that occur before the end of the line). Bibliographers produced new hypotheses about how many compositors were involved in production based on the analysis of the use of spaces before and after punctuation (Howard-Hill, 1973; Taylor, 1981). We additionally incorporate this source of evidence into our automatic approach by modeling pixel-level whitespace distances (§2.3.2).

Bibliographers also use contextual information to inform their analyses, including copy text orthography, printing house records, collation, type case usage, and the use of type with cast-on spaces. In our model, we restrict our analysis to only those features that can be derived from the OCR output and simple visual analysis.

[2] We build on the same assumption in our analysis but acknowledge its limitations: since the typesetters copied the plays from manuscripts or earlier printed versions, often we cannot be sure whether the spelling was chosen by the compositor or simply reproduced from the source document.

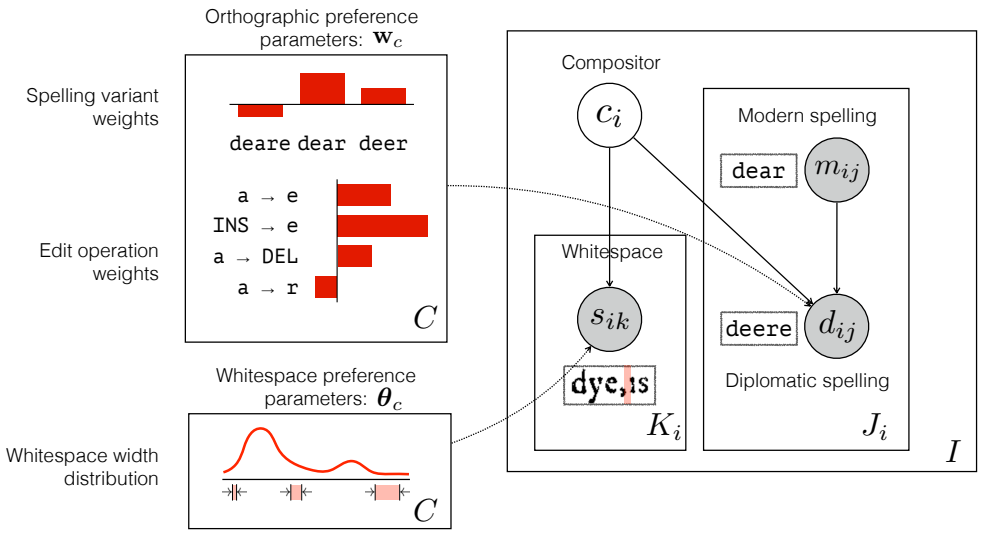


Figure 2.2: A visualization of our model's generative process (§2.3). A compositor c_i is sampled for the i-th page from a multinomial prior. Then, each word's diplomatic spelling, d_{ij}, is sampled conditioned on c_i and the corresponding modern spelling, m_{ij}, from a distribution parameterized by the weight vector $\mathbf{w}_c$. Finally, each medial comma spacing width (measured in pixels), s_{ik}, is sampled conditioning on c_i from a distribution parameterized by $\boldsymbol{\theta}_c$.

2.3 Model

Our computational approach to compositor attribution operates on the sources of evidence that have been considered by bibliographers. In particular, we focus on jointly modeling patterns of orthographic variation and spacing preferences across pages of a document, treating compositor assignments as latent variables in a generative model. We assume access to a diplomatic transcription of the document (a transcription faithful to the original orthography), which we automatically align with a modernized version.[3] We experiment with both manually and automatically (OCR) produced transcriptions and assume access to pixel-level spacing information on each page, which can be extracted using OCR as described in §2.4.2. As discussed in §2.4, we evaluate both parametric and non-parametric versions of the same model; for simplicity, this section describes only the most general parametric setting, and the non-parametric generative process is detailed in §2.4.3.

Figure 2.2 illustrates the generative process under our parametric model. Each of the I pages in the book is generated independently. The compositor assignment for the i-th page is represented by the variable $c_i \in \{1, \ldots, C\}$ and is sampled from a multinomial prior. Each word d_{ij} observed on page i is sampled conditioning on the corresponding modern spelling m_{ij} and

[3] Modern editions are common for many books that are of interest to bibliographers, though future work could consider how to cope with their absence.

the compositor who set the page, c_i. Finally, the model generates the pixel width of the space after each medial comma, s_{ik}, again conditioned on the compositor, c_i. The joint distribution for the data and the compositor assignment for page i, conditioned on the modern text, takes the following form:

$$p(\{d_{ij}\}, \{s_{ik}\}, c_i | \{m_{ij}\}) = \underbrace{p(c_i)}_{\text{Prior on compositors}} \cdot \underbrace{\prod_{j=1}^{J_i} p(d_{ij} | m_{ij}, c_i; \mathbf{w}_{c_i})}_{\text{Orthographic model}} \cdot \underbrace{\prod_{k=1}^{K_i} p(s_{ik} | c_i; \boldsymbol{\theta}_{c_i})}_{\text{Whitespace model}} \quad (2.1)$$

2.3.1 Orthographic Preference Model

We design the parameterization for the distribution of diplomatic spellings in order to capture two types of spelling preferences: (1) general preferences for certain character groups (such as -ie) and (2) preferences that only pertain to a particular word and do not indicate a larger pattern. Since it is unknown which of the two behaviors is dominant (*i.e.* whether the compositors memorized their preferred spellings as a whole or had a particular set of edit preferences determining spelling choices), we let the model describe both and learn to separate their effects. Using a log-linear parameterization:

$$p(d | m, c; \mathbf{w}) \propto \exp\Big(\mathbf{w}_c^\top \mathbf{f}(m, d)\Big),$$

we introduce features to capture both effects. Here, $\mathbf{f}(m, d)$ is a feature function defined on modern word m paired with diplomatic spelling d, while $\mathbf{w}_c$ is a weight vector corresponding to compositor c.

To capture word-specific preferences we add an indicator feature for each pair of modern word m and diplomatic spelling d. We refer to these as WORD features below. To capture general orthographic preferences, we introduce an additional set of features corresponding to the edit operations involved in the computation of the Levenshtein distance between m and d. In particular, each operation is added as a separate feature, both with and without local context (previous or next character in the modern word). We refer to this group as EDIT features. The weight vector for each compositor represents their unique biases, as shown in Figure 2.2.

2.3.2 Whitespace Preference Model

Manual analyses of spacing patterns have also revealed differences between pages. In particular, the choice of spaced or non-spaced punctuation marks is hypothesized by bibliographers to be indicative of compositor preferences and the specifics of the typecase. We add the whitespace width variable to our model to capture these observations. While bibliographers only made a coarse distinction between spaced or non-spaced commas, we explicitly model medial comma

spacing width in pixels s_{ik} to enable finer-grained analysis. We use a simple multinomial parameterization where each whitespace width is treated as a separate outcome up to a certain maximum allowed width:

$$s_{ik}|c_i \sim \text{Mult}\left(\boldsymbol{\theta}_{c_i}\right).$$

Here, $\boldsymbol{\theta}_c$ represents the vector of multinomial spacing parameters specific to compositor c. We choose this parameterization because it can capture non-unimodal whitespace preference distributions,[4] as depicted in Figure 2.2, and it makes learning simpler.

2.4 Experimental Evaluation

2.4.1 Data and Evaluation

To evaluate how well our model performs given perfectly transcribed historical text, we use the Bodleian diplomatic transcription of the *First Folio*.[5] To test whether our approach can also work with not yet transcribed books, we repeat some of the experiments using the Ocular OCR system (Berg-Kirkpatrick et al., 2013) output on the Bodleian facsimile images as an automatic diplomatic transcription. In both cases, we used Ocular's estimates of glyph bounding boxes on the complete *First Folio* images to extract spacing information. The modern text was taken from the MIT Complete Works of Shakespeare[6] and aligned with the diplomatic transcriptions via a word-level edit distance algorithm. Word alignments extracted by this method form the model's observed modern–diplomatic spelling pairs.

To compare the recovered attribution with one proposed by bibliographers, we evaluate against an authoritative attribution compiled by Peter Blayney (1996) which includes the work of various scholars (Hinman, 1963; Howard-Hill, 1973, 1976, 1980; Taylor, 1981; O'Connor, 1975; Werstine, 1982). We also evaluate our system against an earlier, highly influential attribution proposed by Hinman (1963), which we approximate by reverting certain compositor divisions in Blayney's attribution.[7]

Hinman's attribution posited five compositors, while Blayney's posited eight. As the parametric version of our model requires setting the maximum number of compositors in advance, we set it to $C = 5$ when reproducing Hinman's attribution, and use $C = 8$ with Blayney's. However,

[4] Preliminary experiments with a Poisson parameterization showed poorer performance.

[5] http://firstfolio.bodleian.ox.ac.uk/

[6] http://shakespeare.mit.edu/; sourced from the digital Moby Text, which reproduces an 1864 edition of the plays.

[7] Hinman's attribution is reconstructed by replacing Blayney's compositors *F*, *H/H2*, and *I* with compositor *A*, following the historical bibliography literature (Howard-Hill, 1973; Taylor, 1981; Blayney, 1991).

for the non-parametric model, which makes no initial assumption about the number of compositors, we can additionally investigate how its recovered number of clusters corresponds to the different scholars' proposals.

We evaluate the generated attributions by computing the one-to-one and many-to-one accuracies, mapping the recovered page groups to the gold compositor assignments in a way that maximizes accuracy, as is standard for many unsupervised clustering tasks such as POS induction (see Christodoulopoulos et al., 2010). We compute the one-to-one alignment between the recovered and the authoritative clusters via the Hungarian algorithm. In the non-parametric case, where computing one-to-one accuracy might not be possible because the number of compositors varies, we only use the many-to-one accuracy. Another metric used in a subset of our experiments is the pair-counting F-1 measure, which for every possible pair of pages checks whether they fall into the same cluster under both attributions.

2.4.2 Experiments with Parametric Models

BASIC **model variant** We evaluate a simple baseline model that generating diplomatic words under a multinomial parameterization $p(d|m, c; \mathbf{w}) = \text{Mult}(\gamma(\mathbf{w})) \triangleq p(d|m, c; \gamma)$ and does not incorporate subword orthographic features or spacing information. We experiment with two different options for selecting spelling variants to be considered by the model. First, we consider only the three words selected by Hinman: do, go and here (referred to as HINMAN). Second, we use a larger, automatically selected word list (referred to as AUTO). Here, we select all modern words that occur over 70 times, are not proper names, and exhibit sufficient variance in diplomatic spellings (most common diplomatic spelling occurs in less than 80% of aligned tokens). Infrequent spellings (occurring fewer than 3 times) are automatically discarded as typos or alignment errors. The resulting AUTO word list contains 162 words; we use it in all of our experiments with the full model, described in the following paragraph.

FEAT **model variant** We run experiments with several variants of our full parametric model, described in §2.3 (referred to as FEAT since they use a feature-based parameterization for the process of generating diplomatic word). We try ablating WORD and EDIT features, as well as model variants with and without the spacing generation component (referred to as SPACE). We refer to the full model that includes all three feature types as ALL.

Learning and inference The modern and diplomatic words and the spacing widths are observed, while the compositor assignments are latent. In order to fit the model to an input document, we estimate the orthographic preference parameters $\mathbf{w}_c$ and spacing preference parameters $\boldsymbol{\theta}_c$ for each compositor using the Expectation–Maximization algorithm (EM).

The log-likelihood of the observed data under the FEAT parametric model takes the following form:

$$\ell(\mathbf{w}, \boldsymbol{\theta}) = \sum_{i=1}^{I} \sum_{c_i=1}^{C} \left[\log p(c_i) + \sum_{j=1}^{J_i} \log p(d_{ij} | m_{ij}, c_i; \mathbf{w}_{c_i}) + \sum_{k=1}^{K_i} \log p(s_{ik} | c_i; \boldsymbol{\theta}_{c_i}) \right] \tag{2.2}$$

We estimate the feature weights by maximizing the expected log-likelihood in Equation 2.2. The E-step is accomplished via a tractable sum over the compositor assignments. The M-step for the spacing parameters $\boldsymbol{\theta}_c$ uses the standard multinomial update, but as the M-step for $\mathbf{w}_c$ has no closed-form solution, we use gradient ascent (Berg-Kirkpatrick et al., 2010) for optimizing the orthographic parameters. Finally, at inference time, we predict compositor assignments via an independent argmax over each c_i for $i \in \{1, \ldots, I\}$.

Hyperparameters For each model, we run 75 iterations of EM with 100 random restarts, choosing the learned parameters that correspond to the best model likelihood. We use a uniform initialization with a small random noise for all multinomial parameters and feature weights. To exclude lines of prose when extracting SPACE features, we consider only the lines where the text is at least 20 pixels shorter than the longest line of the page.

2.4.3 Experiments with Non-parametric Models

As mentioned in §2.4.1, the true number of compositors who worked on the *First Folio* is not known, and different historians had posited different numbers of compositors. Evaluating our attribution model in terms of how well it can estimate the number of compositor clusters is also an interesting avenue of analysis. We approach it in two ways, building on the parametric model described in §2.3: (1) training multiple instances of the model, each with a different value of C, and seeing which of them fits the data best (referred to as HOLDOUT); and (2) designing a non-parametric extension of the model, which abolishes the need for specifying the number of compositors altogether (BASIC-NP).

Our proposed non-parametric model is based on the BASIC model described in §2.4.2, where the diplomatic spellings are sampled from a per-compositor multinomial distribution $p(d|m, c; \gamma) = \text{Mult}(\gamma)$. It uses a Chinese Restaurant Process (CRP) prior on the compositor variable c_i:

$$P(c_i = k | c_{-i}; \beta) = \begin{cases} \frac{I_{-i}^{(k)}}{i+\beta-1}, & \text{if compositor } k \text{ has been seen before} \\ \frac{\beta}{i+\beta-1}, & \text{if compositor } k \text{ is new,} \end{cases} \tag{2.3}$$

where c_{-i} denotes the compositor assignments for all pages up to i, I is the total number of pages,

and $I_{-i}^{(k)}$ is the number of pages previously assigned to compositor k.

Likelihood estimation via Gibbs sampling We train the non-parametric model using a Gibbs sampler with an infinite mixture of multinomial components and a Dirichlet prior on the compositor parameters $\boldsymbol{\gamma}^{(c)}$. Each mixture component is a collection of multinomials, one per modern word type m, and the pages can be viewed as lists of collections of multinomial outcomes for each of those types. In this section, we first describe the non-marginalized version of the model where the compositor parameters are sampled explicitly (Eq. 2.4) and then introduce the collapsed version where we marginalize over the compositor parameters (Eq. 2.5).

Let us define the sampling process more rigorously. For each compositor $c \in \{1, \ldots, C\}$, we have a set of parameters $\boldsymbol{\gamma}^{(c)} = \left\{\boldsymbol{\gamma}_m^{(c)}\right\}_M$, where $\boldsymbol{\gamma}_m^{(c)}$ parameterize the multinomial distribution over the spellings of each modern word type $m \in M$. The collection of the diplomatic spellings of the modern word m occurring on page i is denoted by $\mathbf{d}_{m,i} = \left\{d_{m,i,1}, \ldots, d_{m,i,J_{m,i}}\right\}$. To avoid redundancy, we also write $p\left(d_{m,i,j} | m, c; \boldsymbol{\gamma}^{(c)}\right)$ as $p\left(d_{m,i,j}; \boldsymbol{\gamma}_m^{(c)}\right)$.

The joint distribution of all the observed spellings, underlying words, compositor assignments, and compositor-specific multinomial parameters then takes the following form:

$$p\left(\{c_i\}_I, \{\mathbf{d}_{m,i}\}_{M,I}, \left\{\boldsymbol{\gamma}_m^{(c)}\right\}_{M,C}; \beta\right) \propto p\left(\{c_i\}_I; \beta\right) \cdot \prod_{i=1}^{I} \prod_{m \in M} \prod_{j=1}^{J_{m,i}} p\left(d_{m,i,j}; \boldsymbol{\gamma}_m^{(c_i)}\right),$$

where $p\left(\{c_i\}_I; \beta\right)$ is defined by the CRP:

$$p(\{c_i\}_I; \beta) = \prod_{i=1}^{I} p(c_i | c_{-i}; \beta).$$

To sample the updated multinomial parameters and the compositor assignments from the posterior $p\left(\{c_i\}_I, \left\{\boldsymbol{\gamma}_m^{(c)}\right\}_{M,C} | \left\{\mathbf{d}_{m,i}\right\}_{M,I}; \beta\right)$, we use a sampling method described by Neal (2000). This method consists of iterating between two sampling steps: given the current Markov chain state $\left(\{c_i\}_I, \left\{\boldsymbol{\gamma}_m^{(c)}\right\}_{M,C}\right)$, we sequentially resample the values of c_i, then the values of the parameters, and so on.

These two steps are formalized as follows. Given an assignment of the pages to the compositors, we sample the next set of compositor parameters from the posterior. Using a conjugate prior, we get:

$$\boldsymbol{\gamma}_m^{(c)} \sim \text{Dir}\left(\boldsymbol{\alpha}_m^{(c)}\right)$$

$$\boldsymbol{\gamma}_m^{(c)} | \left\{\mathbf{d}_{m,i} : c_i = c\right\} \sim \text{Dir}\left(\boldsymbol{\alpha}_m^{(c)\prime}\right),$$

where $\boldsymbol{\alpha}_m^{(c)\prime} = \boldsymbol{\alpha}_m^{(c)} + \sum_{i=1}^{I} \sum_{j=1}^{J_{m,i}} \boldsymbol{\delta}_{m,i,j}^{(c)}$, and $\boldsymbol{\delta}_{m,i,j}^{(c)}$ is a one-hot vector (for each occurrence of the

spelling $d_{m,i,j}$ corresponding to a modern word m on a page set by compositor c, we add 1 to the corresponding α).

Now, given the updated parameters for each compositor, we reassign the pages to the compositors by sampling from a new distribution:

$$P\left(c_i = k | c_{-i}, \left\{\mathbf{d}_{m,i}\right\}_M ; \boldsymbol{\gamma}_m^{(k)}, \beta\right) \propto \begin{cases} \frac{I_{-i}^{(k)}}{i+\beta-1} \cdot \prod_{m \in M} \prod_{j=1}^{J_{m,i}} p\left(d_{m,i,j}; \boldsymbol{\gamma}_m^{(k)}\right), & \text{if } k \text{ seen before} \\ \frac{\beta}{i+\beta-1} \cdot \prod_{m \in M} \prod_{j=1}^{J_{m,i}} p\left(d_{m,i,j}\right), & \text{if } k \text{ new.} \end{cases} \tag{2.4}$$

Equation 2.4 describes the update for the non-marginalized version of our non-parametric model. In the latter case defined by this equation—when a new compositor is introduced—the model marginalizes over the compositor's multinomial parameters. We can apply the same marginalization to existing compositors as well, removing the need to calculate the values of $\boldsymbol{\gamma}_m^{(c)}$. In this marginalized version of the model, we directly use the posterior predictive distribution:

$$\mathbf{d}_{m,i} | \mathbf{d}_{m,-i}, c_i \sim \text{DirMult}\left(\mathbf{d}_{m,i} | \boldsymbol{\alpha}_m^{(c_i)\prime}\right).$$

The Gibbs sampler then performs the following update:[8]

$$P\left(c_i = k | c_{-i}, \left\{\mathbf{d}_{m,i}\right\}_M, \left\{\mathbf{d}_{m,-i}\right\}_M ; \beta\right) \propto \begin{cases} \frac{I_{-i}^{(k)}}{I+\beta-1} \cdot \prod_{m \in M} \prod_{j=1}^{J_{m,i}} p_k\left(d_{m,i,j} | \mathbf{d}_{m,-i}\right), & \text{if } k \text{ seen before} \\ \frac{\beta}{I+\beta-1} \cdot \prod_{m \in M} \prod_{j=1}^{J_{m,i}} p_k\left(d_{m,i,j}\right), & \text{if } k \text{ new,} \end{cases} \tag{2.5}$$

where $p_k\left(d | \mathbf{d}_{m,-i}\right)$ is the Dirichlet-multinomial posterior predictive distribution over the spellings of an underlying word m on pages set by compositor k, whose individual parameters are $\boldsymbol{\alpha}_m^{(k)\prime}$.

The compositor assignments sampled from Eq. 2.5 (or Eq. 2.4 for the non-marginalized model) are then used to compute a Monte Carlo approximation of the log-marginal likelihood, which we then use to select the model that fits the data best.

Model selection As specified in §2.4.2, when fitting parametric models to the data, we train them with many random restarts and then choose one with the highest log-marginal likelihood of all pages combined: that is the model whose predictions we compare against the scholars' attributions. However, Markov Chain Monte Carlo estimation required in the non-parametric setting is time-consuming. For faster evaluation, we hold out a random sample of 100 pages out of the total of 885 for model selection. We learn the compositor parameters from the remaining pages,

[8] On the first iteration, i is used instead of I.

Model Setup		Bodleian Transcription				Ocular OCR Transcription			
		Hinman Attr.		Blayney Attr.		Hinman Attr.		Blayney Attr.	
		1-to-1	M-to-1	1-to-1	M-to-1	1-to-1	M-to-1	1-to-1	M-to-1
RANDOM		22.5	49.6	16.7	49.6	22.5	49.6	16.7	49.6
BASIC	W/ HINMAN	67.9	71.8	60.4	67.3	66.6	70.5	47.1	63.8
	W/ AUTO	64.3	81.0	58.8	81.3	64.9	81.1	53.7	80.7
FEAT	W/ EDIT	75.3	79.1	77.1	83.1	76.8	77.4	76.1	76.0
	W/ EDIT + WORD	81.1	81.1	80.7	80.6	75.1	75.0	74.4	74.4
	W/ EDIT + SPACE	**87.6**	**87.5**	**87.3**	**87.2**	**86.7**	**86.6**	**85.9**	**85.8**
	W/ ALL	83.8	83.7	83.5	83.4	82.5	82.4	82.4	82.2

Table 2.1: One-to-one and many-to-one accuracies for all the setups of the parametric model on manual transcriptions and on OCR text. In the experiments with the BASIC model, we evaluate its performance with the short HINMAN word list and with the large, automatically filtered AUTO word list. We show the results for several variants of our full parametric model (FEAT), both with and without spacing features. A random baseline where compositors are sampled uniformly out of 5 or 8 (for the Hinman and Blayney attributions respectively) is included for comparison.

perform inference via independent maximum likelihood estimation for each held-out page, select the model with the best holdout set likelihood, and evaluate its clustering of all pages of the *First Folio*. It should be noted that this makes our method not truly non-parametric because at the validation/model selection step the model is restricted to the number of compositors it recovered in the training pages.

Parametric HOLDOUT baseline To enable learning the number of compositors under our parametric baseline, we train a set of parametric BASIC models varying C from 2 to 10. Here we again fit the models to the pages allocated for training and then measure the likelihood of the 100 held-out pages to see which value of C results in the best fit. For each value of C, we train the model with multiple random restarts and select the best one in terms of the training set likelihood.

Hyperparameters In all non-parametric experiments, we use $\boldsymbol{\alpha} = (0.1, 0.1, \ldots, 0.1)$ for the prior on each of the spelling multinomials. We set the default CRP strength parameter $\beta = 0.1$, although we experiment with other values of β in §2.5.2.

2.5 Results and Analysis

2.5.1 Parametric Models

Our experimental results for the different parametric models are presented in Table 2.1. The BASIC variant, modeled after Hinman's original procedure, substantially outperforms the random

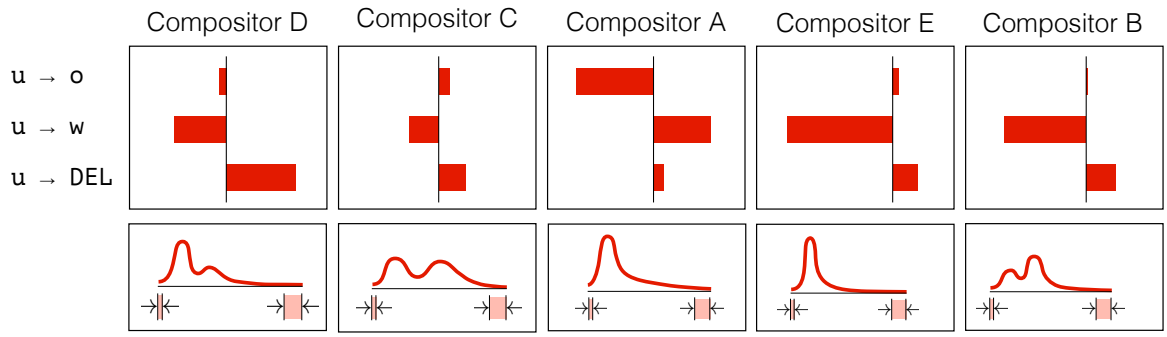


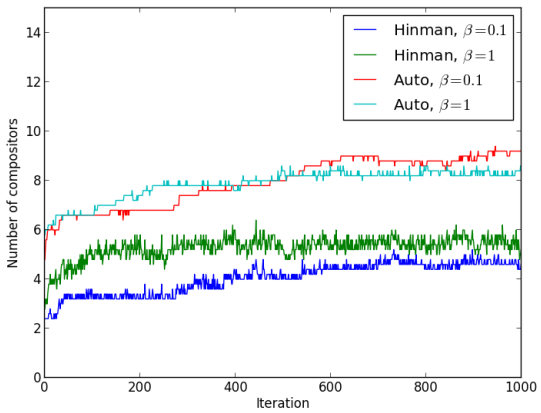
Figure 2.3: Behaviors of the *Folio* compositors learned by our parametric model. Our model only detected the presence of five compositors (ranked according to the number of pages assigned to the compositor by our model). Compositor D's habit of omitting u (yong vs. young) and compositor C's usage of spaced medial commas were also noticed by Taylor (1981).

baseline, with the HINMAN word list resulting in higher accuracy than the larger AUTO word list. However, the use of the larger word list with feature-based models yields large gains in all scenarios, even when evaluating against Hinman's original attributions and using OCR-generated diplomatic transcriptions. The best-performing model for both the manually transcribed and the OCR-produced text uses the EDIT features in conjunction with modeling the spacing and achieves an accuracy of up to 87%. Adding the WORD features on top of this leads to a slight drop in performance, perhaps as a result of a substantial increase in the number of free parameters. In the OCR scenario, using the EDIT and WORD features together decreases accuracy compared to EDIT only, while the same experiment on the manual transcription produces the opposite result. This could be explained by the word-level features being especially brittle to the OCR mistakes.

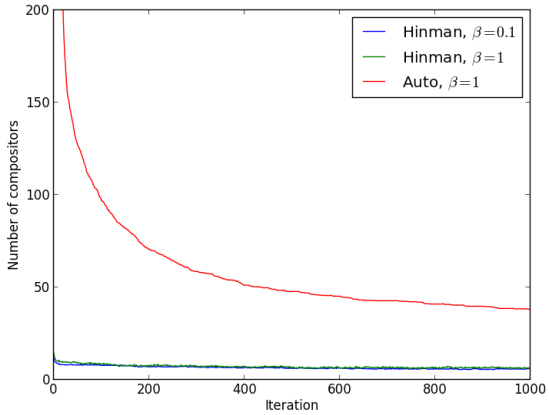
Particularly interesting is the result that modeling spacing, rarely a factor considered in NLP models, improves the accuracy significantly for our system when compared with EDIT features alone. Because pixel-level visual information and arbitrary orthographic patterns are also the most difficult features to measure manually, our results give strong evidence to the assertion that NLP-style models can aid bibliographers.

The results on the OCR transcription (character error rate for most plays $\approx 10-15\%$) are only marginally worse than those on the manual transcription, which shows that our approach can be used in the more common scenario when a manual diplomatic transcription is not available. For our experiments, we also chose a common modern edition of Shakespeare instead of a more carefully produced modernized transcription of the facsimile, our goal being to again show the generalizability of our approach, perhaps to documents where careful modernizations of the facsimile are not available. Together, these results suggest that our model may be sufficiently robust to aid bibliographers in their analysis of less studied texts.

Figure 2.3 shows an example of the feature weights and spacing parameters learned by the



(a) Number of clusters over time as recovered by a marginalized version of BASIC-NP. The bottom two lines correspond to using the HINMAN word list and the top two—to the 160-word long AUTO list. The higher the CRP parameter β is set, the more likely the model is to form new clusters.



(b) Number of clusters over time as recovered by a non-marginalized version of BASIC-NP. When the HINMAN word list is used, the number stays approximately the same. With the AUTO word list, the model starts by forming around 700 clusters and then gradually reduces the number.

Figure 2.4: Number of clusters (compositors) identified by the non-parametric BASIC-NP models vs. the number of Gibbs sampling iterations. Results are averaged over 5 random restarts.

FEAT W/ ALL model. Our statistical approach is able to successfully explain some of the scholars' observations. For example, Taylor (1981) notices that compositors C and D prefer to omit u in young while A does not. Our model reflects this by giving the feature u → DEL a high weight for D and a low weight for A. However, the weight of a single feature is difficult to interpret in isolation: this might be the reason why our model only moderately agrees in the case of compositor C. Another example can be seen in spacing patterns: according to Taylor (1981), compositor C uses spaced medial commas but compositors A and D do not, and our model learns the same behavior.

2.5.2 Non-parametric Models

First, we inspect the number of compositors that our models and baselines converged on. As mentioned in §2.4.1, we specify this number beforehand for the parametric versions to match the scholarly attribution we evaluate on. However, for Blayney's 8-compositor case, our parametric models only populate 5 clusters out of the maximum $C = 8$, leaving out the other 3 compositors. Fitting the HOLDOUT model by varying C from 2 to 10 produces a surprising result: the model yields the best likelihood with just 2 compositors when reconstructing either Hinman's or Blayney's attribution.

We also look at how the number of compositors found by our non-parametric models changes

Model Setup	C learned or fixed?	Hinman Attr.			Blayney Attr.		
		F-1	M-to-1	C	F-1	M-to-1	C
RANDOM	Fixed	19.5	49.6	5	28.4	49.6	8
BASIC	Fixed	54.0	73.2	5	54.4	81.4	8
FEAT W/ ALL	Fixed	**75.6**	**83.7**	5	69.8	**83.4**	8
HOLDOUT	Learned	71.1	70.3	2	68.9	64.2	2
BASIC-NP	Learned	57.8	67.1	6	66.0	74.0	36
BASIC-NP (marginalized)	Learned	59.7	66.6	4	**70.6**	77.6	7

Table 2.2: Empirical comparison of different parametric and non-parametric models' attribution quality. The metrics shown are the pair-counting F-1 measure (F-1) and the many-to-one accuracy of mapping the predicted clusters to the compositors in manual attribution (M-to-1). A random baseline is included for comparison. BASIC many-to-one accuracy differs slightly from one in Table 2.1 due to minor changes in implementation. For the non-parametric models and the HOLDOUT baseline, we also indicate the number of compositors C learned by the model; for all other models, C is set to the 'ground-truth' number posited by the author of the attribution.

over time (Figure 2.4). Here we refer to one cycle of resampling all variables as one iteration. As shown in Figure 2.4a, the predictions of the marginalized non-parametric BASIC model align very closely with the corresponding scholar's observations. When the model is only inferring the clustering from the spellings of the three words Hinman studied (HINMAN word list), it predicts 3–5 compositors where Hinman posited 5. With the larger AUTO word list, it predicts 7–9 compositors, where the current authoritative attribution posits 8. Increasing the strength parameter β makes the model more likely to create new clusters in the beginning, but the model eventually converges on roughly the same number of compositors even with different values of β. Figure 2.4b corresponds to the non-marginalized model, where all the compositor-specific parameters are sampled explicitly. The version trained on the HINMAN word list behaves similarly to the previous experiment, but using the AUTO word list results in the model with so many parameters that the clusters can overspecialize. As a result, the latter grossly overestimates the number of compositors, going from 600–700 at the start of the training to 35–40 at the end.

Finally, we evaluate the attribution accuracy of our non-parametric models (Table 2.2). Both non-parametric BASIC-NP models outperform the parametric BASIC in terms of F-1, but have lower many-to-one accuracy. However, it should be noted that the many-to-one accuracy can become artificially inflated as the number of clusters grows, so it may not be a valid metric for comparing clusterings of different sizes.

2.6 Conclusion

Our primary goal is to scale the methods of compositor attribution, including both textual and visual modes of evidence, for use across books and corpora. We demonstrate how the use of NLP-style modeling techniques can automate some of the manually intensive aspects of bibliographical studies. By using principled statistical techniques and considering the evidence at a larger scale, we offer a more robust approach to compositor identification than has previously been possible. The fact that our system works well on OCR texts means that we are not restricted to only those documents for which we have manually produced transcriptions, opening up the possibility for bibliographic study on a much larger class of texts. Though we are unable to incorporate the kinds of world knowledge used by bibliographers, our ability to include more information and track fine-grained features allows us to recreate their results. Having validated these techniques on the *First Folio*, where historical claims are well-established, we hope that future work can extend these methods and their application.

Chapter 3

Finding Patterns in Idiosyncratic Romanization

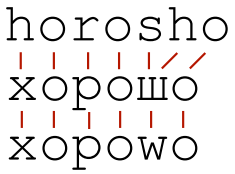
3.1 Introduction

Even when the orthography of a language is relatively standardized, less formal domains like social media can still present significant variation, which modern NLP systems are not yet equipped to process. One notable example of orthographic variation in computer-mediated communication is *informal romanization*[1]—speakers of languages usually written with non-Latin alphabets encoding their messages in Latin characters, for convenience or due to technical constraints (improper rendering of native script or keyboard layout incompatibility). An example of such a sentence can be found in Figure 3.2. Unlike named entity transliteration where the change of script represents the change of language, here Latin characters serve as an intermediate symbolic representation to be decoded by another speaker of the same source language, calling for a completely different transliteration mechanism: rather than expressing the pronunciation of the word according to the phonological rules of another language, informal transliteration is more akin to a substitution cipher, where each source character is replaced with a similar Latin character.

In this chapter, we focus on decoding informally romanized texts back into their original scripts. We view this task as a decipherment problem and propose an unsupervised approach, which allows us to save annotation effort since parallel data for informal transliteration does not occur naturally. We propose a weighted finite-state transducer (WFST) cascade model that learns to decode informal romanization without parallel text, relying only on transliterated data and a

The work presented in this chapter was done in collaboration with Eduard Hovy, Matthew R. Gormley, and Taylor Berg-Kirkpatrick.

[1] Our focus on *informal* transliteration excludes formal settings such as pinyin for Mandarin where transliteration conventions are well established.



[Phonetically romanized]
[Underlying Cyrillic]
[Visually romanized]

Figure 3.1: Example transliterations of a Russian word хорошо [horošo, 'good'] (middle) based on the phonetic (top) and visual (bottom) similarity, with character alignments displayed. The phonetic-visual dichotomy gives rise to one-to-many mappings such as ш /ʃ/ → sh/w.

language model over the original orthography. We test it on romanized texts in three languages, Egyptian Arabic, Kannada, and Russian, where for the latter we collect our own dataset of romanized text from a Russian social network website `vk.com`.

4to mowet bit' ly4we?	[Romanized]
Что может быть лучше?	[Latent Cyrillic]
Čto možet byt' lučše?	[Scientific]
/ʃto ˈmoʒɨt bɨtʲ ˈlut͡ʃʃɨ/	[IPA]
'What can be better?'	[Translated]

Figure 3.2: Example of an informally romanized sentence from the dataset presented in this chapter, containing a many-to-one mapping ж/ш → w. Scientific transliteration, broad phonetic transcription, and translation are not included in the dataset and are presented for illustration only.

Since informal transliteration is not standardized, converting romanized text back to its original orthography requires reasoning about the specific user's transliteration preferences and handling many-to-one (Figure 3.2) and one-to-many (Figure 3.1) character encodings, which is beyond traditional rule-based converters. Although user behaviors vary, there are two dominant patterns in informal romanization that have been observed independently across different languages, such as Russian (Paulsen, 2014), dialectal Arabic (Darwish, 2014) or Greek (Chalamandaris et al., 2006):

- **Phonetic similarity:** Users represent source characters with Latin characters or digraphs associated with similar phonemes (*e.g.* м /m/ → m, л /l/ → l in Figure 3.2). This substitution method requires implicitly tying the Latin characters to a phonetic system of an intermediate language (typically, English).
- **Visual similarity:** Users replace source characters with similar-looking symbols (*e.g.* ч /t͡ʃʲ/ → 4, y /u/ → y in Figure 3.2). Visual similarity choices often involve numerals, especially when the corresponding source language phoneme has no English equivalent (*e.g.* Arabic ع /ʕ/ → 3).

Taking this consistency across languages into account, we show that incorporating these style patterns into our model as priors on the emission parameters—also constructed from naturally occurring resources—improves the decoding accuracy on both romanized Arabic (*Arabizi*) and romanized Russian (*translit*). We compare our proposed unsupervised WFST model with a supervised version of the same model, an unsupervised neural architecture, and commercial decoders (§3.6.1), and find that our unsupervised WFST shows a lower character error rate than the unsupervised neural baseline on all three languages (Russian, Arabic, Kannada; §3.6.2). We also investigate how combining our unsupervised finite-state model with an unsupervised sequence-to-sequence one at decoding time affects the output quantitatively and qualitatively (§3.6.2).

3.2 Related Work

Prior work on informal transliteration uses supervised approaches with character substitution rules either manually defined or learned from automatically extracted character alignments (Darwish, 2014; Chalamandaris et al., 2004). Typically, such approaches are pipelined: they produce candidate transliterations and rerank them using modules encoding knowledge of the source language, such as morphological analyzers or word-level language models (Al-Badrashiny et al., 2014; Eskander et al., 2014). Supervised finite-state approaches have also been explored (Wolf-Sonkin et al., 2019; Hellsten et al., 2017); these WFST cascade models are similar to the one we propose, but they encode a different set of assumptions about the transliteration process due to being designed for abugida scripts (using consonant-vowel syllables as units) rather than alphabets. To our knowledge, there is no prior unsupervised work on this problem.

Named entity transliteration, a task closely related to ours, is better explored, but there is little unsupervised work on this task as well. In particular, Ravi and Knight (2009) propose a fully unsupervised version of the WFST approach introduced by Knight and Graehl (1998), reframing the task as a decipherment problem and learning cross-lingual phoneme mappings from monolingual data. We take a similar path, although it should be noted that named entity transliteration methods cannot be straightforwardly adapted to our task due to the different nature of the transliteration choices. The goal of the standard transliteration task is to communicate the pronunciation of a sequence in the source language (SL) to a speaker of the target language (TL) by rendering it appropriately in the TL alphabet; in contrast, informal romanization emerges in communication between SL speakers only, and TL is not specified. If we picked any specific Latin-script language to represent TL (*e.g.* English, which is often used to ground phonetic substitutions), many of the informally romanized sequences would still not conform to its pronunciation rules: the transliteration process is character-level rather than phoneme-level and does not take possible TL digraphs into account (*e.g.* Russian сх /cx/ → sh), and it often involves eclectic visual substitu-

tion choices such as numerals or punctuation (*e.g.* Arabic تحت [tHt, 'under'][2] → ta7t, Russian для [dlja, 'for'] → dl9|).

Finally, another relevant task is translating between closely related languages, possibly written in different scripts. An approach similar to ours is proposed by Pourdamghani and Knight (2017). They also take an unsupervised decipherment approach: the cipher model, parameterized as a WFST, is trained to encode the source language character sequences into the target language alphabet as part of a character-level noisy-channel model, and at decoding time it is composed with a word-level language model of the source language. Recently, the unsupervised neural architectures (Lample et al., 2018, 2019) have also been used for related language translation and similar decipherment tasks (He et al., 2020). We extend one of these neural models to our character-level setup to serve as a baseline and experiment with combining it with our proposed model at decoding time (§3.5).

3.3 Methods

We train a character-based noisy-channel model that transforms a character sequence o in the native alphabet of the language into a sequence of Latin characters l, and we use it to decode the romanized sequence l back into the original orthography. Our proposed model is composed of separate transition and emission components as discussed in §3.3.1, similarly to a Hidden Markov Model (HMM). However, an HMM assumes a one-to-one alignment between the characters of the observed and the latent sequences, which is not true for our task. One original script character can be aligned to two consecutive Latin characters or vice versa: for example, when a phoneme is represented with a single symbol on one side but with a digraph on the other (Figure 3.1), or when a character is omitted on one side but explicitly written on the other (*e.g.* short vowels not written in unvocalized Arabic but written in transliteration or the Russian soft sign ь representing palatalization being often omitted in the romanized version). To handle those alignments, we introduce insertions and deletions into the emission model and modify the emission transducer to limit the number of consecutive insertions and deletions. In our experiments with the proposed finite-state model, we compare the performance of the model with and without informative phonetic and visual similarity priors described in §3.3.2.

[2]The square brackets following a foreign word show its linguistic transliteration (using the scientific and the Buckwalter schemas for Russian and Arabic respectively) and its English translation.

3.3.1 Model

If we view the process of romanization as encoding a source sequence o into Latin characters, we can consider each observation l to have originated via o being sampled from a distribution $p(o)$ and then transformed to Latin script according to another distribution $p(l|o)$. We can write the probability of the observed Latin sequence as:

$$p(l) = \sum_{o} p(o;\gamma) \cdot p(l|o;\theta) \cdot p_{\text{prior}}(\theta;\alpha). \tag{3.1}$$

The first two terms in Equation 3.1 correspond to the probabilities under the transition model (the language model trained on the original orthography) and the emission model respectively. The third term represents the prior distribution on the emission model parameters through which we introduce human knowledge into the model. Our goal is to learn the parameters θ of the emission distribution with the transition parameters γ being fixed.

We parameterize the emission and transition distributions as weighted finite-state transducers (WFSTs):

Transition WFSA The weighted finite-state acceptor (WFSA) T represents a character-level n-gram language model of the language in the native script, producing the native alphabet character sequence o with the probability $p(o;\gamma)$. We use the parameterization of Allauzen et al. (2003), with the states encoding conditioning history, arcs weighted by n-gram probabilities, and the failure transitions representing backoffs. The role of T is to inform the model of what well-formed text in the original orthography looks like; its parameters γ are learned from a separate corpus and kept fixed during the rest of the training.

Emission WFST The emission WFST S transduces the original script sequence o to a Latin sequence l with the probability $p(l|o;\theta)$. Since there can be multiple paths through S that correspond to the input–output pair (o, l), this probability is summed over all such paths (*i.e.* is a marginal over all possible monotonic character alignments):

$$p(l|o;\theta) = \sum_{e} p(l, e|o;\theta). \tag{3.2}$$

We view each path e as a sequence of edit operations: substitutions of original characters with Latin ones ($c_o \rightarrow c_l$), insertions of Latin characters ($\epsilon \rightarrow c_l$), and deletions of original alphabet characters ($c_o \rightarrow \epsilon$). Each arc in S corresponds to one of the possible edit operations; an arc representing the edit $c_o \rightarrow c_l$ is characterized by the input label c_o, the output label c_l, and the

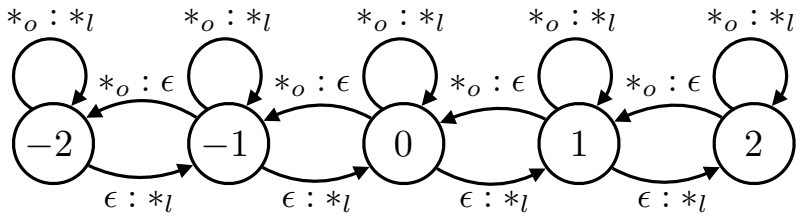


Figure 3.3: Schematic representation of the emission WFST with limited delay (here, up to 2). States are labeled by their delay values. $*_o$ and $*_l$ represent an arbitrary original or Latin symbol respectively. Weights of the arcs are omitted for clarity; weights of the arcs with the same input–output label pairs are tied.

weight $-\log p(c_l|c_o;\theta)$. The emission parameters θ are the multinomial conditional probabilities of the edit operations $p(c_l|c_o)$; we learn θ using the algorithm described in §3.3.3.

3.3.2 Phonetic and Visual Priors

To inform the model of which pairs of symbols are close in the phonetic or visual space, we introduce the priors on the emission parameters, increasing the probability of an original alphabet character being substituted by a similar Latin one. Rather than attempting to operationalize the notions of phonetic or visual similarity, we choose to read the likely mappings between symbols off human-compiled resources that use the same underlying principle: phonetic keyboard layouts and visually confusable symbol lists. Examples of mappings that we encode as priors can be found in Table 3.1.

Phonetic similarity Since we think of informal romanization as a cipher, we aim to capture the phonetic similarity between characters based on type-level association rather than on the actual grapheme-to-phoneme mappings in specific words. We approximate it using *phonetic keyboard layouts*, one-to-one mappings built to bring together "similar-sounding" characters in different alphabets. We take the character pairs from one or multiple layouts for each language: two for Arabic,[3] four for Russian,[4] and one for Kannada.[5] One drawback of using keyboard layouts is that they require every character to have a Latin counterpart, so some mappings will inevitably be arbitrary; for two out of three languages, we try to compensate for this by averaging over several layouts. We refer to priors constructed from such keyboard layouts as 'phonetic', although their character mappings might reflect transliteration conventions as much as they reflect perceptual

[3] https://thomasplagwitz.com/2013/01/06/imrans-phonetic-keyboard-for-arabic/, http://arabic.omaralzabir.com/

[4] http://winrus.com/kbd_e.htm

[5] http://kaulonline.com/uninagari/kannada/

similarity—which, in our case, is also an advantage.

Visual similarity The strongest example of visual character similarity would be *homoglyphs*—symbols from different alphabets represented by the same glyph, such as Cyrillic а and Latin a. The fact that homoglyph pairs can be made indistinguishable in certain fonts has been exploited in phishing attacks, *e.g.* when Latin characters are replaced by virtually identical Cyrillic ones (Gabrilovich and Gontmakher, 2002). This led the Unicode Consortium to publish a list of symbols and symbol combinations similar enough to be potentially confusing to the human eye (referred to as *confusables*).[6] This list contains not only exact homoglyphs but also strongly homoglyphic pairs such as Cyrillic Ю and Latin IO.

We construct a visual prior for the Russian model from all Cyrillic–Latin symbol pairs in the Unicode confusables list.[7] Although this list does not cover more complex visual associations used in informal romanization, such as partial similarity (Arabic Alif with Hamza أ → 2 due to Hamza ء resembling an inverted 2) or similarity conditioned on a transformation such as reflection (Russian л → v), it makes a sensible starting point. However, this restrictive definition of visual similarity does not allow us to create a visual prior for Arabic or Kannada—their scripts are dissimilar enough from Latin that the confusables list contains very few Arabic–Latin or Kannada–Latin character pairs. Proposing a more nuanced definition of visual similarity for Arabic and Kannada and constructing the associated priors is left for future work.

We incorporate these mappings into the model as Dirichlet priors on the emission parameters: $\theta \sim \mathrm{Dir}(\alpha)$, where each dimension of the parameter α corresponds to a character pair (c_o, c_l), and the corresponding element of α is set to the number of times these symbols are mapped to each other in the predefined mapping set.

3.3.3 Learning

We learn the emission WFST parameters in an unsupervised fashion, observing only the Latin side of the training instances. The marginal likelihood of a romanized sequence l can be computed by summing over the weights of all paths through the lattice obtained by composing $T \circ S \circ A(l)$. Here $A(l)$ is an unweighted acceptor of l, which, when composed with a lattice, constrains all paths through the lattice to produce l as the output sequence. The Expectation–Maximization (EM) algorithm is commonly used to maximize marginal likelihood; however, the size of the

[6] https://www.unicode.org/Public/security/latest/confusables.txt

[7] In our parameterization, we cannot introduce a mapping from one to multiple symbols or vice versa, so we map all possible pairs instead: (ю, IO) → (ю, I), (ю, O).

Original	Latin	
	‘Phonetic’	Visual
р /r/	r	p
б /b/	b	b, 6
в /v/	v, w	b
ҙ /w, u:, o:/	w, u	—
ҁ /x/	k, x	—

Table 3.1: Example Cyrillic–Latin and Arabic–Latin mappings encoded in the visual and ‘phonetic’ priors respectively.

lattice would make the computation prohibitively slow. We combine online learning (Liang and Klein, 2009) and curriculum learning (Bengio et al., 2009) to achieve faster convergence.

Unsupervised Learning

We use a version of the stepwise EM algorithm described by Liang and Klein (2009), reminiscent of the stochastic gradient descent in the space defined by the sufficient statistics. Training data is split into mini-batches, and after processing each mini-batch we update the overall vector of the sufficient statistics μ and re-estimate the parameters based on the updated vector. The update is performed by interpolating between the current value of the overall vector and the vector of sufficient statistics s_k collected from the k-th mini-batch: $\mu^{(k+1)} \leftarrow (1 - \eta_k)\mu^{(k)} + \eta_k s_k$. The step size is gradually decreased, causing the model to make smaller changes to the parameters as the learning stabilizes. Following Liang and Klein (2009), we set it to $\eta_k = (k + 2)^{-\beta}$.

However, if the mini-batch contains long sequences, summing over all paths in the corresponding lattices could still take a long time. As we know, the character substitutions are not arbitrary: each original alphabet symbol is likely to be mapped to only a handful of Latin characters across the entire corpus, which means that most of the paths through the lattice would have very low probabilities. We prune the improbable arcs in the emission WFST while training on batches of shorter sentences. Doing this eliminates up to 66% and up to 76% of the emission arcs for Arabic and Russian respectively.

We discourage excessive use of insertions and deletions by keeping the corresponding probabilities low at the early stages of training: during the first several updates, we freeze the deletion probabilities at a small initial value and disable insertions completely to keep the model locally normalized. We also iteratively increase the language model order as the learning progresses. Once most of the emission WFST arcs have been pruned, we can afford to compose it with a larger language model WFST without the size of the resulting lattice rendering the computation impractical. The two steps of the EM algorithm are performed as follows:

- **E-step:** At the E-step, we compute the sufficient statistics for updating θ, which in our case would be the expected number of traversals of each of the emission WFST arcs. For ease of bookkeeping, we compute those expectations using finite-state methods in the expectation semiring (Eisner, 2002). Summing over all paths in the lattice is usually performed via shortest distance computation in the log semiring; in the expectation semiring, we augment the weight of each arc with a basis vector, where the only non-zero element corresponds to the index of the emission edit operation associated with the arc (*i.e.* the input-output label pair). This way the shortest distance algorithm yields not only the marginal likelihood but also the vector of the sufficient statistics for the input sequence.

  To speed up the shortest distance computation, we shrink the lattice by limiting the delay of all paths through the emission WFST. The delay of a path is defined as the difference between the number of epsilon labels on the input and output sides of the path. Figure 3.3 shows the schematic representation of the emission WFST with limited delay. Substitutions are performed without a state change, and each deletion or insertion arc transitions to the next or previous state respectively. When the first (last) state is reached, further deletions (insertions) are no longer allowed.

- **M-step:** The M-step then corresponds to simply re-estimating θ by appropriately normalizing the obtained expected counts.

Supervised Learning

We also compare the performance of our model with the same model trained in a supervised way, using the annotated portion of the data that consists of pairs of parallel native-script and Latin-script sequences (o, l). In the supervised case, we can additionally constrain the lattice with an acceptor of the original-orthography sequence: $A(o) \circ T \circ S \circ A(l)$. However, the alignment between the symbols in o and l is still latent. To optimize this marginal likelihood we still employ the EM algorithm; however, as this constrained lattice is much smaller, we can run the standard EM without the modifications discussed above.

3.3.4 Decoding

Inference at test time is also performed using finite-state methods and closely resembles the E-step of the unsupervised training algorithm: given a Latin sequence l, we construct the machine $T \circ S \circ A(l)$ in the tropical semiring and run the shortest path algorithm to obtain the most probable path $\hat{e}$; the source sequence $\hat{o}$ is then read off the obtained path.

	Train (source)		Train (target)		Validation		Test	
	Sent.	Char.	Sent.	Char.	Sent.	Char.	Sent.	Char.
Arabic	5K	104K	49K	935K	301	8K	1K	20K
Russian	5K	319K	307K	111M	227	15K	1K	72K
Kannada	10K	1M	679K	64M	100	11K	100	10K

Table 3.2: Dataset splits for each language. The source and target train data are 'monolingual' (native- or Latin-script sequences only), while the validation and test sentences are parallel. The source and target sides correspond to the Latin and the original script respectively. All Arabic and Kannada data comes from the LDC BOLT Phase 2 and Dakshina corpora respectively, with all sentences annotated with their transliteration into the native script. For the experiments on Russian, the language model is trained on a section of the Taiga corpus, and the train, validation, and test portions are collected by the authors; only the validation and test sentences are annotated.

3.4 Datasets

We conduct experiments with romanized data in three languages, Arabic, Russian, and Kannada, all from different language families. They also span three major types of writing systems—abjad, alphabetic, and abugida, respectively—which allows us to empirically compare how well-suited the assumptions built into our character-level model are for learning their various alignment patterns. Table 3.2 shows the sizes of train, validation, and test sets for each of the three languages.

Romanization for South Asian languages or for Arabic has been explored in prior computational work, but Russian romanization has not, as its use online has declined in recent years. Since a dataset of informally romanized Russian was not available, we collect and partially annotate our own dataset from the Russian social network `vk.com` (§3.4.2).

3.4.1 Arabic

We use the Arabizi portion of the LDC BOLT Phase 2 SMS/Chat dataset (Bies et al., 2014; Song et al., 2014), a collection of written informal conversations in romanized Egyptian Arabic annotated with their Arabic script representation. To prevent the annotators from introducing orthographic variation inherent to dialectal Arabic, compliance with the Conventional orthography for dialectal Arabic (CODA; Habash et al., 2012) is ensured. However, the effects of some of the normalization choices (*e.g.* expanding frequent abbreviations or adjusting word boundaries; see Figure 3.4) result in discrepancies between the source and target sides, which pose difficulties to our model.

To obtain a subset of the data better suited for our task, we discard any instances which are not originally romanized (5% of all data), ones where the Arabic annotation contains Latin characters

Source:	de el menu:)
Filtered:	de el menu<...>
Target:	<...>دي ألنه
Gloss:	'This is the menu'

Figure 3.4: A parallel example from the LDC BOLT Arabizi dataset, written in Latin script (source) and converted to Arabic (target) semi-manually. Some source-side segments (shown in red) are removed by the annotators; we use the version without such segments (filtered) for our task. The annotators also standardize word boundaries on the target side, which results in differences with the source (shown in blue).

(4%), or where emoji/emoticon normalization was performed (12%). The information about the splits is provided in Table 3.2. Most of the data is allocated to the language model training set in order to give the unsupervised model enough signal from the native script side. We choose to train the transition model on the annotations from the same corpus to make the language model specific to both the informal domain and the CODA orthography.

3.4.2 Russian

This section describes the data collection and annotation process for the new corpus of informally romanized Russian introduced in this chapter.

Scraping transliterated data We collect our romanized Russian data from the social network website `vk.com`, adopting an approach similar to the one described by Darwish (2014). We take a list of the 50 most frequent Russian lemmas (Lyashevskaya and Sharov, 2009), filtering out those shorter than 3 characters, and produce a set of candidate romanizations for each of them to use as queries to the `vk.com` API. In order to encourage diversity of romanization styles in our dataset, we generate the queries by defining all plausible visual and phonetic mappings for each Cyrillic character and applying all possible combinations of those substitutions to the underlying Russian word, yielding 270 candidate transliterations of 26 Russian words to use as queries. However, many of the produced combinations are highly unlikely and yield no results, and some happen to share the spelling with words in other languages (most often other Slavic languages that use Latin script, such as Polish). We scrape public posts on the user and group pages, retaining only the information about which posts were authored by the same user, and manually go over the collected set to filter out coincidental results.

Preprocessing and splitting We additionally preprocess the collected data by lowercasing, normalizing punctuation, and removing non-ASCII characters and emoji. We also shorten all sub-

Annotated	
Source:	proishodit s prirodoy 4to to very very bad
Filtered:	proishodit s prirodoy 4to to <...>
Target:	происходит с природой что-то <...>
Gloss:	'Something very very bad is happening to the environment'
'Monolingual'	
Source:	—
Target:	это видеоролики со съезда партии "Единая Россия"
Gloss:	'These are the videos from the "United Russia" party congress'

Figure 3.5: **Top:** A parallel example from the romanized Russian dataset. We use the filtered version of the romanized (source) sequences, removing the segments the annotators were unable to convert to Cyrillic, such as code-switched phrases (shown in red). The annotators also standardize minor spelling variation such as hyphenation (shown in blue). **Bottom:** a 'monolingual' Cyrillic example from the `vk.com` portion of the Taiga corpus, which mostly consists of comments in political discussion groups.

strings of the same character repeated more than twice to only two repetitions (as suggested by Darwish et al., 2012) since these repetitions are more likely to be a written expression of emotion than to be explained by the underlying Russian sentence. We later apply the same preprocessing to the native-script side of the data as well.

Our dataset consists of 1,796 wall posts from 1,681 users and communities. Since the posts are quite long on average (248 characters, with longest ones up to 15K), we split them into sentences using the NLTK sentence tokenizer, with manual correction when needed. The obtained sentences are used as data points, split into training, validation, and test as reported in Table 3.2. The average length of an obtained sentence is 65 characters, which is 3 times longer than an average Arabizi sentence; we believe this is due to the different nature of the data (social media posts vs. SMS). Sentences collected from the same user are distributed across different splits so that we observe a diverse set of romanization preferences in both training and testing. Each sentence in the validation and test sets is annotated by one of the two native speaker annotators, following guidelines similar to those designed for the Arabizi BOLT data (Bies et al., 2014).

Annotation While transliterating, annotators perform orthographic normalization wherever possible, correcting typos and errors in word boundaries (Figure 3.5, top, shown in blue); grammatical errors are not corrected. Tokens that do not require transliteration (foreign words, emoticons) or ones that the annotator fails to identify (proper names, words misspelled beyond recognition) are removed from the romanized sentence and not transliterated (Figure 3.5, top, shown in red). Although it means that some of the test set sentences will not exactly represent the original roman-

ized sequence, it will help us ensure that we are only testing our model's ability to transliterate rather than make word-by-word normalization decisions.

In addition, 200 of the validation sequences are dually annotated to measure the inter-annotator agreement. We evaluate it using character error rate, the same metric we use to evaluate the model's performance (§3.5.3). In this case, since neither of the annotations is the ground truth, we compute CER in both directions and average. Despite the discrepancies caused by the annotators deleting unknown words at their discretion, the average CER is only 0.014, which indicates a very high level of agreement.

'Monolingual' Cyrillic data Since we do not have enough annotations to train the Russian language model on the same corpus, we use a separate dataset, collected from the same social network `vk.com`. We use the relevant portion of the Taiga corpus (Shavrina and Shapovalova, 2017), containing 307K comments from public groups. An example sentence from this dataset is shown in Figure 3.5 (bottom). It should be noted that although both sides were scraped from the same online platform, the Taiga data is collected primarily from political discussion groups, so there is still a substantial domain mismatch between the source and target sides of the data. We apply the same preprocessing steps here as we did in the romanized data collection process.

3.4.3 Kannada

Our Kannada data (Figure 3.6) is taken from the Dakshina dataset (Roark et al., 2020), a large collection of native-script text from Wikipedia for twelve South Asian languages. Unlike the Russian and Arabic data, the romanized portion of Dakshina is not scraped directly from online communication but instead elicited from native speakers given the native-script sequences. Because of this, all romanized sentences in the data are parallel; we allocate most of them to the source side training data, discarding their original script counterparts, and split the remaining annotated ones between validation and test (Table 3.2).

Target:	ಮೂಲ ಸಾಕೆಟ್‌ನಲ್ಲಿ DDR3ಯನ್ನು ಬಳಸಲು
Source:	moola saaketnalli ddr3yannu balasalu
Gloss:	'to use DDR3 in the source circuit'

Figure 3.6: A parallel example from the Kannada portion of the Dakshina dataset. The Kannada-script data (target) is scraped from Wikipedia and manually converted to Latin script (source) by human annotators. Foreign target-side characters (shown in red) get preserved in the annotation but our preprocessing replaces them with an UNKsymbol on the target side.

3.5 Experiments

3.5.1 Models

We compare the performance of four different model classes: our proposed finite-state model (§3.3.1) with and without informative priors (§3.3.2); an unsupervised neural model which encodes no assumptions about the underlying process at all; various combinations of our finite-state model and the neural sequence-to-sequence one; and commercial handcrafted decoders which directly represent the human knowledge about the transliteration process.

WFST models We evaluate the performance of our finite-state model trained in three different setups: unsupervised with a uniform prior on the emission parameters, unsupervised with informative phonetic and visual priors, and supervised. We train the unsupervised models with the stepwise EM algorithm as described in §3.3.3, performing stochastic updates and making only one pass over the entire training set. The supervised models are trained on the validation set with five iterations of EM and a six-gram transition model. It should be noted that only a subset of the validation data is actually used in the supervised training: if the absolute value of the delay of the emission WFST paths is limited by n, we will not be able to compose a lattice for any data points where the input and output sequences differ in length by more than n (those constitute 22% of the Arabic validation data and 33% of the Russian validation data for $n = 5$ and $n = 2$ respectively). Since all of the Arabic data comes annotated, we can perform the same experiment using the full training set; surprisingly, the performance of the supervised model does not improve (see Table 3.3).

Neural baseline Our sequence-to-sequence (seq2seq) baseline is the unsupervised neural machine translation (UNMT) model of Lample et al. (2018). We follow the implementation by He et al. (2020) with one important change: since the romanization process is known to be strictly character-level, we tokenize the text into characters rather than words. We also explore several ways to combine an independently-trained UNMT model with our WFST, described below.

WFST and neural model combinations The simplest way to combine two independently trained models is reranking: using one model to produce a list of candidates and rescoring them according to another model. We apply this process in both directions: using WFST to generate candidates and UNMT to rerank them, and vice versa. To generate candidates with a WFST, we apply the n–shortest paths algorithm (Mohri and Riley, 2002). It should be noted that the n–best list might contain duplicates since each path represents a specific source–target character alignment. The length constraints encoded in the WFST also restrict its capacity as a reranker: beam

search under the UNMT model may produce hypotheses too short or long to have a non-zero probability under the WFST.

Our second approach is a product-of-experts-style joint decoding strategy (Hinton, 2002): we perform beam search on the WFST lattice, reweighting the arcs with the output distribution of the seq2seq decoder at the corresponding timestep. For each partial hypothesis, we keep track of the WFST state s and the partial input and output sequences $x_{1:k}$ and $y_{1:t}$.[8] When traversing an arc with an input label $i \in \{x_{k+1}, \epsilon\}$ and an output label o, we multiply the arc weight by the probability of the neural model outputting o as the next character: $p_{\text{seq2seq}}(y_{t+1} = o | x, y_{1:t})$. Transitions with $o = \epsilon$ (*i.e.* deletions) are not rescored by the seq2seq. We group the partial hypotheses by their consumed input length k and select n best extensions at each timestep.

Handcrafted decoders For Russian and Arabic, we also use online transliteration decoding systems as baselines: `translit.net` (Russian) and Yamli[9] (Arabic). The Russian decoder is rule-based, but the algorithm used in the Arabic decoder is not disclosed.

3.5.2 Implementation Details

We use the OpenFst library (Allauzen et al., 2007) for the implementation of the finite-state models, in conjunction with the OpenGrm NGram library (Roark et al., 2012) for training the transition model specifically. We train the character-level n-gram models of orders from two to six with Witten–Bell smoothing (Witten and Bell, 1991). Since the WFSTs encoding full higher-order models become very large (for example, the Russian six-gram model has 3M states and 13M arcs), we shrink all the models except for the bigram one using relative entropy pruning (Stolcke, 1998). However, since pruning decreases the quality of the language model, we observe the most improvement in accuracy while training with the unpruned bigram model, and the subsequent order increases lead to relatively minor gains.

We optimize the delay limit for each language separately, obtaining the best results with 2 for Russian and 5 for Arabic and Kannada. To approximate the monotonic word-level alignment between the original and Latin sequences, we restrict the operations on the space character to only three: insertion, deletion, and substitution with itself. We apply the same to the punctuation marks (with specialized substitutions for certain Arabic symbols, such as ? → ؟). This substantially reduces the number of arcs in the emission WFST, as punctuation marks make up over half of each of the alphabets.

Our joint seq2seq–WFST decoding implementations rely on PyTorch and the Pynini finite-state library (Gorman, 2016). In reranking, we rescore $n = 5$ best hypotheses produced via

[8] Due to insertions and deletions in the emission model, k and t might differ; epsilon symbols are not counted.

[9] `https://www.yamli.com/`

beam search and the n–shortest path algorithm for the UNMT and WFST models respectively. Product-of-experts decoding is also performed with beam size 5.

Further implementation details and hyperparameter settings for all models can be found in the Appendix (§A.1).

3.5.3 Evaluation

We use character error rate (CER) as our main evaluation metric. We compute CER as the character-level edit distance between the predicted original-script sequence and the human annotation (reference) divided by the length of the reference sequence in characters. For some of the experiments, we also report the word error rate (WER; computed the same as CER except the edit distance is word-level), and the BLEU-4 score (Papineni et al., 2002).[10] For both BLEU and WER, we split sentences into words using the Moses tokenizer (Koehn et al., 2007).

3.6 Results and Analysis

3.6.1 Varying Levels of Supervision

Our first series of experiments, focusing on Russian and Arabic, aims to determine how much information relevant for our task is contained in the character similarity mappings, and how it compares to the amount of information encoded in the human annotations. We compare them by evaluating the effect of the informative priors (described in §3.3.2) on the performance of the unsupervised model and comparing it to the performance of the supervised model. We also evaluate the performance of an unsupervised neural baseline.

The CER values for the models we compare are presented in Table 3.3. One trend we notice is that the error rate is lower for Russian than for Arabic in all the experiments, including the uniform prior setting, which suggests that decoding Arabizi is an inherently harder task. Some of the errors of the Arabic commercial system could be explained by the decoder predictions being plausible but not matching the CODA orthography of the reference.

Effect of priors The unsupervised model without an informative prior performs poorly for either language, which means that there is not enough signal in the language model alone under the training constraints we enforce. Possibly, the algorithm could have converged to a better local optimum if we did not use the online algorithm or prune both the language model and the

[10] Measured using the Moses toolkit script: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

	Arabic	Russian
Unsupervised: uniform prior	0.735	0.660
Unsupervised: 'phonetic' prior	0.377	0.222
Unsupervised: visual prior	—	0.372
Unsupervised: combined prior	—	0.212
Supervised	0.225*	0.140
UNMT	0.791	0.242
Commercial	0.206	0.137

Table 3.3: Character error rates obtained in different experimental setups. For each language, we compare the unsupervised models with and without informative priors with the supervised model (trained on the validation data) and a commercial online system. We do not have a visual prior for Arabic due to the Arabic–Latin visual character similarity not being captured by the restrictive confusables list that defines the prior (see §3.3.2). Each of the supervised and unsupervised experiments is performed with 5 random restarts. *The Arabic supervised model result is reported for the model trained on the validation set; training on the 5K training set yields 0.226.

Original	Latin
p /r/	r (.93), p (.05)
б /b/	b (.95), 6 (.02)
в /v/	v (.87), 8 (.05), w (.05)
ҩ /w, uː, oː/	w (.48), o (.33), u (.06)
ҳ /x/	5 (.76), k (.24)

Table 3.4: Emission probabilities learned by the supervised model for a subset of native-script characters (compare to Table 3.1). For each native-script character, all substitutions with a probability greater than 0.01 are shown.

emission model; however, that experiment would be infeasibly slow. Incorporating the 'phonetic' prior reduces the error rate by 0.36 and 0.44 for Arabic and Russian respectively, providing a substantial improvement while maintaining the efficiency advantage. The visual prior for Russian appears to be slightly less helpful, improving CER by 0.29. We attribute the better performance of the model with the 'phonetic' prior to the sparsity and restrictiveness of the visually confusable symbol mappings, or it could be due to the phonetic substitutions being more popular with users. Finally, combining the two priors for Russian leads to a slight additional improvement in accuracy over the 'phonetic' prior only.

To additionally verify that the phonetic and visual similarity-based substitutions are prominent in informal romanization, we inspect the emission parameters learned by the supervised model with a uniform prior (Table 3.4). We observe that: (a) the highest-probability substitutions can

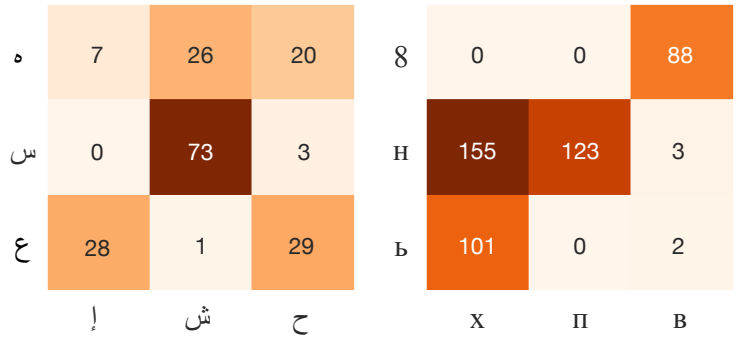


Figure 3.7: Fragments of the confusion matrix comparing the test-time predictions of the best-performing unsupervised models for Arabic (left) and Russian (right) to the human annotations. Each number represents the count of the corresponding substitution in the best alignment (edit distance path) between the predicted and the gold sequences, aggregated over the test set. Rows stand for predictions, columns correspond to ground truth.

be explained by either phonetic or visual similarity, and (b) the external mappings we use for our priors are indeed appropriate since the supervised model recovers the same mappings in the annotated data.

Error analysis Figure 3.7 shows some of the elements of the confusion matrices for the test-time predictions of the best-performing unsupervised models in both languages. Many of the frequent errors are caused by the model failing to disambiguate between two plausible decodings of a Latin character, either based on two different types of similarity (н /n/ [phonetic] → n ← [visual] п, н [visual] → h ← [phonetic] х /x/) or on the same one (visual 8 → 8 ← в, phonetic ه /h/ → h ← ح /ħ/); such cases could be ambiguous for humans to decode as well.

Other errors in Figure 3.7 illustrate the limitations of our parameterization and the resources we rely on. Our model does not allow one-to-many alignments, which leads to digraph interpretation errors such as س /s/ + ه /h/ → sh ← ش /ʃ/. Some artifacts of the resources our priors are based on also pollute the results: for example, the confusion between ь and x in Russian is explained by the Russian soft sign ь, which has no English phonetic equivalent, being arbitrarily mapped to the Latin x in one of the phonetic keyboard layouts.

Comparison to UNMT The unsupervised neural model trained on Russian performs only marginally worse than the unsupervised WFST model with an informative prior, demonstrating that with a sufficient amount of data the neural architecture is powerful enough to learn the character substitution rules without the need for the inductive bias. However, we cannot say the same about Arabic—with a smaller training set (see Table 3.2), the UNMT model is outperformed by the unsupervised WFST even without an informative prior (although it shows better results with

	Arabic			Russian			Kannada		
	CER	WER	BLEU	CER	WER	BLEU	CER	WER	BLEU
WFST	.405	.86	2.3	.202	.58	14.8	.359	**.71**	12.5
Seq2Seq	.571	.85	4.0	.229	**.38**	**48.3**	.559	.79	11.3
Reranked WFST	**.398**	.85	2.8	.195	.57	16.1	**.358**	**.71**	12.5
Reranked Seq2Seq	.538	**.82**	**4.6**	.216	.39	45.6	.545	.78	**12.6**
Product of experts	.470	.88	2.5	**.178**	.50	22.9	.543	.93	7.0

Table 3.5: Character and word error rates (lower is better) and BLEU scores (higher is better) for the finite-state and the neural model and their combinations. **Bold** indicates the best result per column. Model combinations mostly interpolate between the base models' scores, although reranking yields minor improvements in the character-level and the word-level metrics for the WFST and seq2seq respectively. **Note:** base model results are not intended as a direct comparison between the WFST and the seq2seq, since they are trained on different amounts of data.

a slightly different preprocessing method; see Table 3.5 and §A.2). The main difference between the unsupervised finite-state and sequence-to-sequence models comes down to the trade-off between structure and power: although the neural architecture captures long-range dependencies better due to having a stronger language model, it does not provide an easy way of enforcing character-level constraints on the decoding process, which the WFST model encodes by design. As a result, we observe that while the UNMT model can recover whole words more successfully (lower BLEU and WER; see Table 3.5), it also tends to arbitrarily insert or repeat words in the output, which leads to higher CER.

3.6.2 Comparing Finite-State and Neural Models

Since the finite-state and the neural architectures fall at the opposite ends of the structure–power spectrum, we perform further comparative analysis to investigate how these typological properties affect their performance on the romanization decipherment task. This section details our experiments comparing the two unsupervised base models: the UNMT sequence-to-sequence model and the WFST trained with the prior that resulted in the best performance (combined 'phonetic'+visual for Russian, 'phonetic'-only for Arabic and Kannada). We also explore whether the disparate strengths of these base architectures can be successfully harnessed by combining them at decoding time via reranking and product of experts (§3.5.1).

Table 3.5 presents the character and word error rates and BLEU scores for all models and their combinations on all three languages. The CER for the base WFST and UNMT on Russian and Arabic differs slightly from Table 3.3, due to minor changes in hyperparameter settings (§A.1) and preprocessing (§A.2). The results for the base models support what we show later in this

Input	kongress ne odobril biudjet dlya osuchestvleniye “bor’bi s kommunizmom” v yuzhniy amerike.	
Ground truth	конгресс не одобрил бюджет для осуществления “борьбы с коммунизмом” в южной америке.	kongress ne odobril bjudžet dlja osuščestvlenija “bor’by s kommunizmom” v južnoj amerike.
WFST	конгресс не одобрил виудет дла осусчествлениые “бор#би с коммунизмом” в уузнани америке.	kongress ne odobril viudet dla osuščestvleniye “bor#bi s kommunizmom” v uuznani amerike.
Reranked WFST	конгресс не одобрил видет дела осусчествлениые “бор#би с коммунизмом” в уузнани америке.	kongress ne odobril videt dela osuščestvleniye “bor#bi s kommunizmom” v uuznani amerike.
Seq2Seq	конгресс не одобрил бы удивительно с коммунизмом” в южный америке.	kongress ne odobril by udivitel’no s kommunizmom” v južnyj amerike.
Reranked Seq2Seq	конгресс не одобрил бюджет для осуществление “борьбы с коммунизмом” в южный америке.	kongress ne odobril bjudžet dlja osuščestvlenie “bor’by s kommunizmom” v južnyj amerike.
Product of experts	конгресс не одобрил бидет для а осуществлениые “борьбы с коммунизмом” в уузник амери	kongress ne odobril bidet dlja a osuščestvleniye “bor’by s kommunizmom” v uuznik ameri

Table 3.6: Decipherment outputs generated by different models for a Russian transliteration example (left—Cyrillic, right—scientific romanization). Prediction errors are shown in red. Correctly transliterated segments that do not match the reference because of the annotators’ spelling standardization decisions are shown in yellow. # stands for UNK.

section: the sequence-to-sequence model is more likely to recover words correctly (higher BLEU, lower WER) while the WFST is more faithful on character level and avoids word-level deletion or insertion errors (lower CER). Table 3.6 shows example outputs produced by all models and model combinations for a romanized Russian sentence.[11] These examples showcase the general error patterns: the WFST errors are character-level and scattered across the sentence, while the neural model is prone to hallucinations, although a WFST reranker can keep it from deleting and inserting words freely. Our further qualitative and quantitative findings are summarized in the following high-level takeaways:

1. Model combinations still suffer from search issues.

We would expect the decoding-time combinations of the finite-state and the neural model to discourage all errors common under one model but not the other, improving the performance

[11] Examples for Arabic and Kannada can be found in the Appendix (Table A.1, Table A.2).

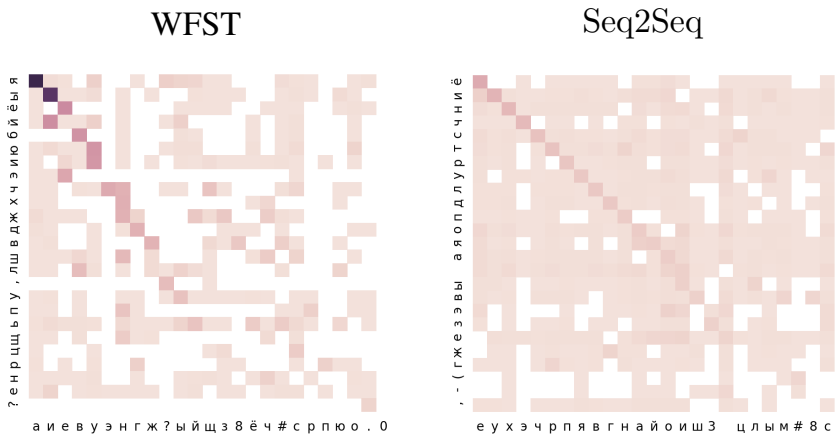


Figure 3.8: Highest-density submatrices of the two base models' character confusion matrices, computed for the Russian romanization task. Color intensity matches the value in the corresponding cell, and white cells represent zeros. The WFST confusion matrix (left) is noticeably sparser than the sequence-to-sequence one (right), indicating more repetitive errors. # stands for UNK.

by leveraging the strengths of both model classes. However, as Table 3.5 shows, they instead mostly interpolate between the scores of the two base models. In the reranking experiments, we find that this is often due to the same base model error (*e.g.* the sequence-to-sequence model hallucinating a word mid-sentence) repeating across all the hypotheses in the final beam. This suggests that successful reranking would require a much larger beam size or a diversity-promoting search mechanism.

Interestingly, we observe that although adding a reranker on top of a decoder does improve performance slightly, the gain is only in terms of the metrics that the base decoder is already strong at—character-level for the reranked WFST and word-level for the reranked seq2seq—at the expense of the other scores. Overall, none of our decoding strategies is a clear leader and no model combination substantially improves over both base models in any of the metrics.

2. The finite-state model makes more repetitive errors.

Although two of our evaluation metrics, CER and WER, are based on edit distance, they do not distinguish between the different types of edits (substitutions, insertions, and deletions). Breaking them down by the edit type, we find that while both models favor substitutions on both word and character levels, insertions and deletions are more frequent under the neural model (43% vs. 30% of all edits on the Russian romanization task). We also find that the character substitution choices of the neural model are more context-dependent: while the total counts of the substitution errors made by the two models are comparable, the WFST is more likely to repeatedly make the same few substitution errors per character type. This is illustrated by Figure 3.8, which visualizes the

most populated submatrices of the confusion matrices for the same task as heatmaps. The WFST confusion matrix is noticeably more sparse, with the same few substitutions occurring much more frequently than others: for example, WFST often mistakes я for a and rarely for other characters, while the neural model's substitutions of я are distributed closer to uniform. This suggests that the WFST errors might be easier to correct with rule-based postprocessing. Interestingly, we did not observe the same effect for the translation task, likely due to a more constrained nature of the orthography conversion.

3. The neural model is more sensitive to data distribution shifts.

The language model aiming to replicate its training data distribution could cause the output to deviate from the input significantly. This could be an artifact of a domain shift, like in the Russian romanization task, where the LM training data came from political discussion forums: the sequence-to-sequence model frequently predicts unrelated domain-specific proper names in place of very common Russian words, *e.g.* жизнь [žizn, 'life'] → Зюганов [Zjuganov, 'Zyuganov (politician's last name)'] or это [èto, 'this'] → Единая Россия [Edinaja Rossija, 'United Russia (political party)'], presumably distracted by the shared first character in the romanized version. To quantify the effect of a mismatch between the Russian train and test data distributions, we inspect the most common word-level substitutions under each decoding strategy, looking at all substitution errors covered by the 1,000 most frequent substitution 'types' (ground truth–prediction word pairs) under the respective decoder. We find that 25% of the seq2seq substitution errors fall into this category, as compared to merely 3% for the WFST—notable given the relative proportion of in-vocabulary words in the models' outputs (89% for UNMT vs. 65% for WFST).

3.7 Conclusion

This chapter tackles the problem of decoding non-standardized informal romanization used in social media into the source orthography of the language without parallel text. We propose a WFST noisy-channel model to decode romanized Egyptian Arabic and Russian to their original scripts and train it using the stepwise EM algorithm combined with curriculum learning. We then empirically demonstrate that while the unsupervised model by itself performs poorly, introducing an informative prior that encodes the notion of phonetic or visual similarity between characters brings its performance substantially closer to that of the supervised model.

The informative priors used in our experiments are constructed using sets of character mappings compiled for other purposes but using the same underlying principle (phonetic keyboard layouts and the Unicode confusable symbol list). While these mappings provide a convenient way to avoid formalizing the complex notions of phonetic and visual similarity, they are restric-

tive and do not capture all the diverse aspects of similarity that idiosyncratic romanization uses, so designing more suitable priors via operationalizing the concept of character similarity could be a promising direction for future work. Future research could experiment with creating truly phonetic priors (*e.g.* by using grapheme-to-phoneme conversion to map characters to each other via IPA; Lee et al., 2020; Mortensen et al., 2018) or even phonological ones where character similarity is measured as proximity between their representations in the articulatory feature space (Mortensen et al., 2016; Moran and McCloy, 2019). For finer-grained visual priors, similarities can be computed directly from renderings of glyphs (see Bedrick et al., 2012), possibly conditioned on various affine transforms.

We also analyze the types of errors made by our proposed WFST model with informative priors, an unsupervised neural MT model, and their decoding-time combinations, all trained to decode romanization in Russian, Arabic, and Kannada. We find that the two model types tend towards different errors: sequence-to-sequence models are more prone to word-level errors caused by distributional shifts while finite-state models produce more character-level noise despite the hard alignment constraints. Although none of our simple decoding-time combinations substantially outperform the base models, we believe that combining neural and finite-state models to harness their complementary advantages is a promising research direction. For example, such combinations might involve biasing sequence-to-sequence models towards WFST-like behavior via pretraining or directly encoding constraints such as hard alignment or monotonicity into their parameterization (Wu et al., 2018; Wu and Cotterell, 2019). We hope that our analysis provides further insight into leveraging the strengths of the two approaches for modeling character-level phenomena in the absence of parallel data.

Part II

Non-Standard and Novel Lexemes

Chapter 4

Modeling Word Emergence in Semantic Space

4.1 Introduction

The previous chapters discussed language variation as a result of the idiosyncratic behaviors of the users. Some of these behaviors, initially non-standard, eventually become adopted by larger communities of users in the constant process of *language change* (Aitchison, 2001). Perhaps the most obvious type of change is the introduction of new lexical items, or *neologisms* (a process called *neology*). Neologisms have various sources. They are occasionally coined out of whole cloth (grok). More frequently, they are loanwords from another language (tahini), derived words (unfriend), or existing words that have acquired new senses (as when web came to mean 'World Wide Web' and then 'the Internet'). While neology has long been of interest to linguists (§4.2), there have been relatively few attempts to study it as a global, systemic phenomenon. Computational modeling and analysis of neology are the focus of our work.

What are the factors that predict neology? Certainly, social context plays a role. Close interaction between two cultures, for example, may result in increased borrowing (Appel and Muysken, 2006). We hypothesize, though, that there are other factors involved—factors that can be modeled more directly. These factors can be understood in terms of *supply* and *demand*.

Bréal (1904) introduced the idea that the distribution of words in the semantic space tends towards uniformity. This framework predicts that new words would emerge where they would repair uniformity—where there was space not occupied by a word. We refer to this mechanism as supply-driven neology; although we define supply in terms of the layout of the semantic space, that is merely a statistical effect of the underlying cause—the selective pressures on the language

The work presented in this chapter was done in collaboration with Ella Rabinovich, Taylor Berg-Kirkpatrick, David R. Mortensen, and Yulia Tsvetkov.

to be efficient and expressive (Xu and Kemp, 2015).

Next, demand plays a role as well as supply (Campbell, 2013); in particular, we posit that new words emerge in "stylish" neighborhoods, corresponding to domains of discourse that are increasing in importance (reflected by the increasing frequency of the words in those neighborhoods). Such frequency effects are a reflection of the language users' communicative need (Kemp et al., 2018; Karjus, 2021): word emergence and survival within a linguistic community "must to a certain extent depend upon the chief interests of a people" (quoting Boas, 1911).

We operationalize these ideas using distributional semantics (Lenci, 2018). To formalize the hypothesis of supply-driven neology for computational analysis, we measure *sparsity of areas in the word embedding space* where neologisms would later emerge. The demand-driven view of neology motivates our second hypothesis: *neighborhoods in the embedding space containing words rapidly growing in frequency* are more likely to produce neologisms. Both hypotheses are defined more formally in §4.3.

Having formalized our hypotheses in terms of word embeddings, we test them by comparing the distributions of the corresponding metrics for a set of automatically identified neologisms and a control word set. The methodology of the word selection and hypothesis testing is detailed in §4.4. We discuss the results in §4.5, demonstrating evidence for both hypotheses, although the demand-driven hypothesis has more significant support.

4.2 Background

Neology Specific sources of neologisms have been studied: lexical borrowing (Taylor and Grant, 2014; Daulton, 2012), morphological derivation (Lieber, 2017), blends or portmanteaus (Cook, 2012; Renner et al., 2012), clippings, acronyms, analogical or arbitrary coinages. However, these studies have tended to look at neologisms atomistically or to explicate the social conditions under which a new word entered a language rather than looking at neologisms in a systemic context.

To address this deficit, we look back to the seminal work of Michel Bréal, who introduced the idea that words exist in a semantic space. His work implies that, other things being equal, the semantic distribution of words tends toward uniformity (Bréal, 1904). This is most explicit in his law of differentiation, which states that near-synonyms move apart in semantic space, but has other implications as well. For example, the same principle would predict that new words are more likely to emerge in areas where they would increase uniformity. This could be viewed as supply-driven neology—new words appear to fill gaps in semantic space (to express concepts that are not currently lexicalized).

In the linguistic literature, neology is often associated with new concepts or domains of in-

creasing importance (Campbell, 2013). Just as there are factors that predict where houses are built other than the availability of land, there are factors that predict where new words emerge other than the availability of semantic space. Demand, we hypothesize, plays a role in neology as well as supply.

Most existing computational research on the mechanisms of neology focuses on discovering sociolinguistic factors that predict acceptance of emerging words into the mainstream language and growth of their usage, typically in online social communities (Del Tredici and Fernández, 2018). These variables can include geography (Eisenstein, 2017), user demographics (Eisenstein et al., 2012, 2014), diversity of linguistic contexts (Stewart and Eisenstein, 2018), or word form (Kershaw et al., 2016). This chapter presents one of the first studies focused on discovering factors predictive of the emergence of new words rather than modeling their lifecycle: concurrently with our work, Hofmann et al. (2020) explored the social and linguistic factors correlated with the emergence of novel morphological derivatives, and Karjus et al. (2020) empirically showed that trending topics produce more new words.

Distributional semantics and language change Word embeddings have been successfully used for different applications of the diachronic analysis of language (Tahmasebi et al., 2021). The closest task to ours is analyzing meaning shift (tracking changes in word sense or the emergence of new senses) by comparing word embedding spaces across time periods (Kulkarni et al., 2015; Xu and Kemp, 2015; Hamilton et al., 2016; Kutuzov et al., 2018). Typically, embeddings are learned for discrete time periods and then aligned (but see Bamler and Mandt, 2017; Dubossarsky et al., 2019). There has also been work on revising the existing methodology, specifically accounting for frequency effects in embeddings when modeling semantic shift (Dubossarsky et al., 2017).

Other related questions where distributional semantics proved useful were exploring the evolution of bias (Garg et al., 2018) and the degradation of age- and gender-predictive language models (Jaidka et al., 2018).

4.3 Hypotheses

This section outlines the two hypotheses we introduced earlier from the linguistic perspective, formalized in terms of distributional semantics.

Hypothesis 1 *Neologisms are more likely to emerge in sparser areas of the semantic space.* This corresponds to the supply-driven neology hypothesis: we assume that areas of the space that contain fewer semantically related words are likely to give birth to new ones so as to fill in the

'semantic gaps'. Word embeddings give us a natural way of formalizing this: since semantically related words have been shown to populate the same regions in the embeddings space, we can translate semantic sparsity into geometric sparsity and measure the density of the word's semantic neighborhood as the number of word vectors within a certain distance of the word's embedding.

Hypothesis 2 *Neologisms are more likely to emerge in semantic neighborhoods of growing popularity.* Here we formalize our demand-driven view of neology, which assumes that the growing frequency of words in a semantic area is a reflection of its growing importance in discourse and that the latter is in turn correlated with the emergence of neologisms in that area. In terms of word embeddings, we again consider nearest word vectors as the word's semantic neighbors and quantify the rate at which their frequencies grow over decades (formally defined in §4.4.4).

4.4 Methodology

Our analysis is based on comparing the embedding space neighborhoods of neologism vectors against the neighborhoods of the embeddings of words from an alternative control set. Our method for automatically identifying neologisms is described in §4.4.2, and in §4.4.4 we detail the factors we control for when selecting the alternative word set. In §4.4.1, we describe the datasets used in our experiments. Our data is split into two large corpora, HISTORICAL and MODERN; we additionally require the HISTORICAL corpus to be split into smaller time periods so that we can estimate the word frequency change rate. Embedding models are trained on each of the two corpora, as described in §4.4.3. We compare the neighborhoods in the HISTORICAL embedding space, but due to the nature of our neologism selection process, many neologisms might not exist in the HISTORICAL vocabulary. To locate their neighborhoods, we adopt an approach from prior work on diachronic distributional semantics: we learn an orthogonal projection between the HISTORICAL and MODERN embeddings to align the two spaces in order to make them comparable (see Hamilton et al., 2016) and then use vector projections to represent neologisms in the HISTORICAL space (§4.4.3). Finally, §4.4.5 describes the hypothesis testing setup: which statistics we choose to quantify our two hypotheses and how their distributions are compared.

4.4.1 Datasets

We use the Corpus of Historical American English (COHA; Davies, 2002) and the Corpus of Contemporary American English (COCA; Davies, 2008), large diachronic corpora balanced by genre to reflect the variation in word usage. The COHA data is split into decades; we group the COHA documents from 18 decades (1810–1989) to represent the HISTORICAL English collection

and use the full COCA corpus (1990–2012) as the MODERN corpus. The obtained HISTORICAL split contains 405M tokens of 2M types, and MODERN contains 547M tokens of 3M types.[1]

4.4.2 Neologism Selection

We rely on a usage-based approach to extract the set of neologisms for our analysis, choosing the words based on their patterns of occurrence in our datasets. It can be seen as an approximation of selecting words by their earliest recorded use dates, as these dates are also reconstructed based on the words' usage in historical corpora. This analogy is supported by the qualitative analysis of the obtained set of neologisms (§4.6).

We limit our analysis to nouns, an open-class lexical category. We detect nouns in our corpora using a part-of-speech dictionary: for each word in our vocabulary, we retrieve its POS label distribution in a POS-tagged corpus of English Wikipedia data (Wikicorpus; Reese et al., 2010). If the word's most frequent label is 'NN', we classify it as a noun. We additionally filter the candidate neologisms to exclude words that occur more frequently in capitalized than lowercase form; this heuristic helps us remove proper nouns missed by the POS tagger.

We select a set of neologisms by identifying words that are substantially more frequent in the MODERN corpus than in the HISTORICAL one. It is important to note that while we use the term "neologism," implying a word at the early stages of emergence, this method selects words that have entered mainstream vocabulary in MODERN time but might have been coined prior to that. We consider a word w to be a neologism if its ratio $f^{(M)}(w)/f^{(H)}(w)$ is greater than a certain threshold; here $f^{(M)}(\cdot)$ and $f^{(H)}(\cdot)$ denote word frequencies (normalized counts) in the MODERN and HISTORICAL data respectively. Empirically we set the frequency ratio threshold to 20.

We rank the words satisfying these criteria by their frequency in the MODERN corpus and select the first 1,000 words to be our neologism set; this is to ensure that we only analyze words that subsequently become mainstream and not misspellings or other artifacts of the data.

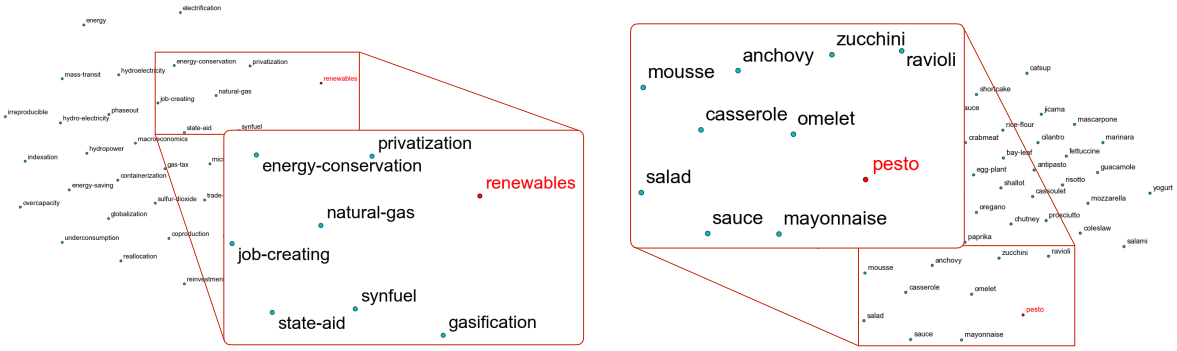
4.4.3 Embeddings

Our hypothesis testing process involves inspecting semantic neighborhoods of the neologisms in the HISTORICAL embedding space. However, many neologisms are very infrequent or nonexistent in the HISTORICAL data, so we approximate their vectors in the HISTORICAL space by projecting their MODERN embeddings into the same coordinate axes.

We learn Word2Vec Skip-Gram embeddings[2] (Mikolov et al., 2013) of the two corpora and

[1] Statistics accompanying the corpora state that the entire COHA dataset contains 385M words, and COCA contains 440M words; we assume the discrepancy is explained by tokenization differences.

[2] Hyperparameters: vector dimension 300, window size 5, minimum count 5.



(a) Semantic neighborhood of the word renewables.

(b) Semantic neighborhood of the word pesto.

Figure 4.1: Neighborhoods of the projected MODERN embeddings of two neologisms (shown in red), renewables and pesto, in the HISTORICAL embedding space, visualized using t-SNE (Maaten and Hinton, 2008). Figure 4.1a shows an example of a neighborhood exhibiting frequency growth: words like synfuel or privatization have been used more often towards the end of the HISTORICAL period. The neighborhood also includes natural-gas that can be seen as representing a concept to be replaced by renewables. The word pesto (Figure 4.1b) is projected into a neighborhood of other food-related words, most of which are also loanwords, several from the same language; it also has its hypernym sauce as one of its neighbors.

use the orthogonal Procrustes method to learn the aligning transformation:

$$\mathbf{R} = \arg\min_{\mathbf{\Omega}} |\mathbf{\Omega}\mathbf{W}^{(M)} - \mathbf{W}^{(H)}|,$$

where $\mathbf{W}^{(H)}, \mathbf{W}^{(M)} \in \mathbb{R}^{|V| \times d}$ are the word embedding matrices learned from the HISTORICAL and MODERN data respectively, restricted to the intersection of the vocabularies of the two corpora (*i.e.* embeddings of all words present in both spaces are used as anchors). To project the MODERN word embeddings into the HISTORICAL space, we multiply them by the obtained rotation matrix $\mathbf{R}$.

4.4.4 Control Set Selection

To test our hypotheses, we collect an alternative set of words and analyze how certain statistical properties of their neighbors differ from those of the neighbors of neologisms. At this stage, it is important to control for non-semantic confounding factors that might affect how words are distributed in the semantic space. One such factor is word frequency: it has been shown that embeddings of words of similar frequency tend to be closer in the embedding space (Schnabel et al., 2015; Faruqui et al., 2016), which results in very dense clusters, or hubs, of words with high cosine similarity (Radovanović et al., 2010; Dinu et al., 2014). We choose to also restrict

our control set to only include words that did not substantially grow or decline in frequency over the HISTORICAL period in order to prevent selecting counterparts that only share similar frequency in the MODERN subcorpus (*e.g.* due to recent topical relevance), but exhibit significant fluctuation prior to that period. In particular, we refrain from selecting words that emerged in language right before our HISTORICAL–MODERN split.

We create this alternative set by pairing each neologism with a non-neologism counterpart that exhibits a stable frequency pattern, additionally controlling for word frequency and word length in characters. Length is chosen as an easily accessible correlate to other factors for which one should control, such as morphological complexity, concreteness, and nativeness. We perform the pairing only to ensure that the distribution of those properties across the two sets is comparable, but once the selection process is complete we treat control words as a set rather than considering them in pairs with neologisms.

Following Stewart and Eisenstein (2018), we formalize the frequency growth rate as the Spearman correlation coefficient between timesteps $\{1, \ldots, T\}$ and frequency series $f^{(H)}_{\{1:T\}}(w)$ of word w. In our setup, timesteps $\{1, \ldots, 18\}$ enumerate decades from 1810s to 1980s, and $f^{(H)}_t(\cdot)$ denotes word frequency in the corresponding t-th decade of the HISTORICAL data.

Formally, for each neologism w_n we select a counterpart w_c satisfying the following constraints:

- Frequencies of the two words in the corresponding corpora are comparable: $3/4 < f^{(M)}(w_n)/f^{(H)}(w_c) < 4/3$;
- The length of the two words is identical up to 2 characters;
- The Spearman correlation coefficient r_s between decades $\{1, \ldots, 18\}$ and the control word frequency series $f^{(H)}_{\{1:18\}}(w_c)$ is small: $|r_s\left(\{1:18\}, f^{(H)}_{\{1:18\}}(w_c)\right)| \leq 0.1$

These words, which we will refer to as *stable*, make up our default and most restricted control set. We will also compare neologisms to a *relaxed* control set, omitting the stability constraint on the frequency change rate but still controlling for length and overall frequency, to see how neologisms differ from non-neologisms in a broader perspective.

4.4.5 Experimental Setup

We evaluate our hypotheses by inspecting neighborhoods of neologisms and their stable control counterparts in the HISTORICAL embedding space, viewing them as proxies for neighborhoods in the underlying semantic space. Since many neologisms are very infrequent or nonexistent in the HISTORICAL data, we approximate their vectors in the HISTORICAL space with their MODERN embeddings projected using the transformation described in §4.4.3. The neighborhood of a word w is defined as the set of HISTORICAL words for which cosine similarity between their HISTORICAL

embeddings and v_w exceeds the given threshold τ; v_w denotes a projected MODERN embedding if w is a neologism or a HISTORICAL embedding if it is a control word.[3]

The two factors we need to formalize are the semantic sparsity of the neighborhoods and the increase in popularity of the topic that the neighborhood represents. We use the sparsity in the embedding space as a proxy for semantic sparsity and approximate growth of interest in a topic with the frequency growth of the words belonging to it (*i.e.* embedded into the corresponding neighborhood). For the neighborhood of each word w, we compute the following statistics, corresponding to our two hypotheses:

1. *Density of a neighborhood* $d(w, \tau)$: number of words that fall into this neighborhood $d(w, \tau) = |\{u : \text{cosine}(v_w, v_u) \geq \tau, u \neq w\}|$
2. *Average frequency growth rate of a neighborhood* $r(w, \tau)$: as defined in the previous subsection, we compute the Spearman correlation coefficient between the timesteps and the frequency series for each word in the neighborhood and then take their mean:

$$r(w, \tau) = \frac{1}{d(w, \tau)} \sum_{u:\text{cosine}(v_w, v_u) \geq \tau, u \neq w} r_s\left(\{1 : 18\}, f_{(1:18)}(u)\right)$$

In our tests, we compare the values of those metrics for the neighborhoods of neologisms and the neighborhoods of control words and estimate the significance of each of the two factors for a range of neighborhood sizes defined by the threshold τ. We test whether the means of the distributions of those statistics for the neologism and the control set differ and whether each of the two is significant for classifying words into neologisms and controls.

As mentioned in §4.4.2, our vocabulary is restricted to nouns only, and we also only consider noun neighbors when evaluating the statistics.[4] Since all neologism word vectors are projected from the MODERN to the HISTORICAL embedding space, regardless of whether they occur in the HISTORICAL corpus or not, we might find a HISTORICAL vector of the neologism itself among the neighbors of its projection; we exclude such neighbors from our analysis. We cap the number of nearest neighbors to consider at 5,000, to avoid estimating statistics on overly large sets of possibly less relevant neighbors.

[3] Cosine similarity is chosen as our distance metric since it is traditionally used for word similarity tasks in distributional semantics (Lenci, 2018). We have also observed the same results when repeating the experiments with the Euclidean distance metric.

[4] Here we refer to the vocabulary of words participating in our analysis, not the embedding model vocabulary; embeddings are trained on the entire corpora.

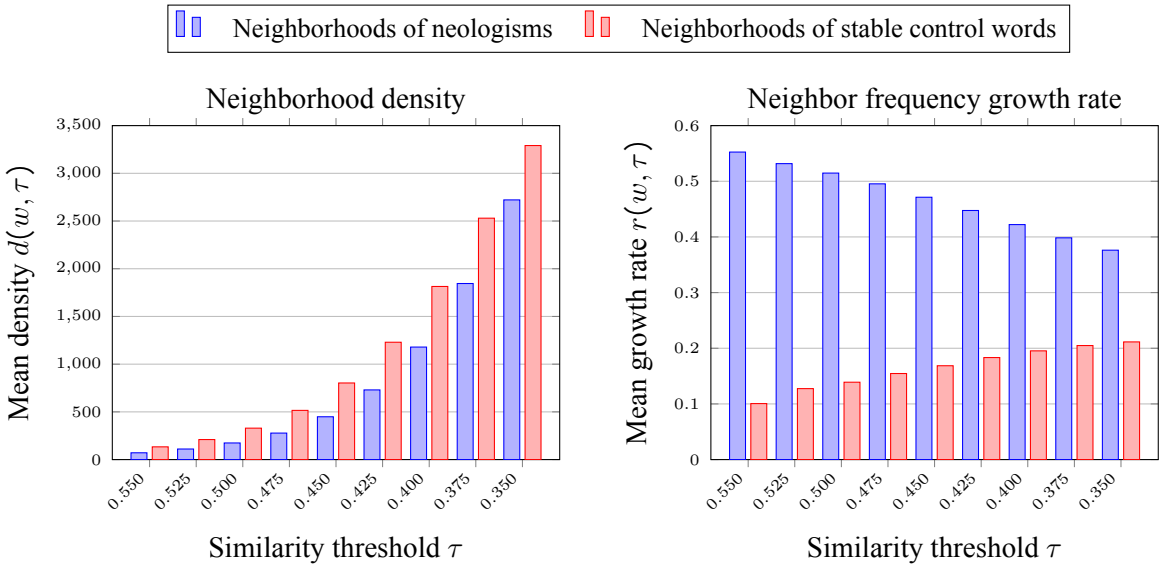


Figure 4.2: **Left:** Number of HISTORICAL word vectors within the cosine distance $1 - \tau$ of the vector of a given word, averaged across the neologism (blue) and the stable control word (red) sets. Projected neologism vectors appear in lower-density neighborhoods compared to the control words. **Right:** Average frequency growth rate (represented by the Spearman correlation coefficient) of those HISTORICAL neighbor words, averaged across the neologism (blue) and the stable control word (red) sets. Neighbors of the projected neologisms exhibit a stronger growth trend than those of the control words, especially in smaller neighborhoods.

4.5 Results

Following the experimental setup described in §4.4.5, we estimate the contribution of each of the hypothesized factors using the strictly constrained and relaxed control sets. We start by analyzing how the distributions of those statistics differ for neologisms and stable controls, both by comparing their sample means and by more rigorous statistical testing. We also evaluate the significance of the factors using generalized linear models for both stable and relaxed control sets.

4.5.1 Comparison to Stable Control Set

First, we test our hypotheses on the 720 neologism-stable control word pairs (not all words are paired in the stable control setting due to its restrictiveness).

Figure 4.2 demonstrates the density and frequency growth rate values for a range of neighborhood sizes, averaged over the neologism and the control sets. Both results conform with our hypotheses: the chart on the left shows that, on average, the projected neologism has fewer neighbors than its stable counterpart, especially for larger neighborhoods; the chart on the right shows that, on average, the frequencies of the neighbors of a projected neologism grow at a faster rate

Neighborhood size	Stable control set				Relaxed control set			
	Density		Growth		Density		Growth	
	$\beta_d^{(\tau)} \times 10^4$	p-value	$\beta_r^{(\tau)} \times 10$	p-value	$\beta_d^{(\tau)} \times 10^4$	p-value	$\beta_r^{(\tau)}$	p-value
Large ($\tau = 0.35$)	1.98	8.25×10^{-5}	1.84	2.35×10^{-80}	-1.07	5.63×10^{-4}	0.61	2.83×10^{-34}
Medium ($\tau = 0.45$)	0.20	8.29×10^{-1}	1.16	2.92×10^{-80}	-3.67	4.00×10^{-10}	0.46	6.19×10^{-46}
Small ($\tau = 0.55$)	6.90	2.90×10^{-2}	0.70	1.61×10^{-68}	-8.92	4.01×10^{-5}	0.28	1.19×10^{-36}

Table 4.1: Values of the GLM coefficients and their corresponding p-values for the different neighborhood sizes, defined by the cosine similarity threshold τ. $\beta_d^{(\tau)}$ and $\beta_r^{(\tau)}$ denote the coefficients for density and average frequency growth respectively for the neighborhoods with threshold τ. Comparing the results for the stable and relaxed control sets, we find that for the stable controls, density is only significant in larger neighborhoods, but without the stability constraint, both factors are significant for all neighborhood sizes.

than those of its counterpart. Interestingly, we find that the neighbors of the stable control words still tend to exhibit a small positive growth rate. We attribute it to the general pattern that we observed: about 70% of words in our vocabulary have a positive frequency growth rate. We believe this might be explained by the imbalance in the amount of data between decades (*e.g.* the 1980s subcorpus has 20 times more tokens than the 1810s): some words might not occur until later in the corpus because of the relative sparsity of data in the early decades.

As we can see from Figure 4.2 (left), neighborhoods of larger sizes (corresponding to lower values of the similarity threshold) may contain thousands of words, so the statistics obtained from those neighborhoods might be less relevant; we might only want to consider the immediate neighborhoods, as those words are more likely to be semantically related to the central word. It is notable that the difference in the growth trends of the neighbors is substantially more prominent for smaller neighborhoods (Figure 4.2, right): the average correlation coefficient of the immediate neighbors of the stable words also falls into the stable range as we defined it, while immediate neighbors of the neologisms exhibit rapid growth.

4.5.2 Statistical Significance

To estimate the significance and the relative contribution of the two factors, we fit a generalized linear model (GLM) with a logistic link function to the corresponding features of the neighborhoods of the neologism and the control words:[5]

$$y(w) \sim \Big[1 + \exp\Big(-\beta_0^{(\tau)} - \beta_d^{(\tau)} \cdot d(w,\tau) - \beta_r^{(\tau)} \cdot r(w,\tau)\Big)\Big]^{-1}$$

[5] We use the implementation provided in the MATLAB Statistics and Machine Learning Toolbox.

where y is a Bernoulli variable indicating whether the word w belongs to the neologism set (1) or the control set (0), and τ is the cosine similarity threshold defining the neighborhood size.

Table 4.1 shows how the coefficients and p-values for the two statistics change with the neighborhood size. We found that when comparing with the stable control set, the average frequency growth rate of the neighborhood was significant for all sizes, but neighborhood density was significant at level $p < 0.01$ only for the largest ones.[6] We attribute this to the effect discussed in the previous section: the difference in the average frequency growth rate between the neighbors of neologisms and stable words shrinks as we include more remote neighbors (Figure 4.2, right), so for large neighborhoods frequency growth rate by itself is no longer predictive enough.

We also evaluate the significance of the same features for the relaxed control set (without the stability constraint) on 1,000 neologism–control pairs. We have repeated the experiment with 5 different randomly sampled relaxed control sets (results for one shown in Table 4.1). For medium-sized neighborhoods ($0.4 \leq \tau \leq 0.5$), the density variable was always significant at $p < 0.01$, but for the largest and the smallest neighborhoods, density was rejected in several runs. With more variance in the control set, the differences in the neighborhood frequency growth rate between neologisms and controls are less prominent than in the stable setting, so density plays a more important role in prediction.[7]

Growth feature weights $\beta_r^{(\tau)}$ were always positive and density feature weights $\beta_d^{(\tau)}$ were negative in the relaxed setting (where density is significant). This matches our intuition that the neighborhood's frequency growth and its sparsity are predictive of neology.

Comparing sample means of the density and the growth rate between the neologism set and each of the 5 randomly selected relaxed control sets (as we did for the stable control set in Figure 4.2) demonstrated that neologisms still appear in sparser neighborhoods than their control counterparts. The difference in the frequency growth rate between the neologism and the control word neighborhoods is also observed for all control sets (although it varies noticeably between sets), but it no longer exhibits an inverse correlation with the neighborhood size.

4.6 Discussion

We have demonstrated that our two hypotheses hold for the set of words we automatically labeled as neologisms. To establish the validity of our results, we qualitatively examine the obtained word list to see if the words are in fact recent additions to the language. We randomly sample 100 words

[6] Applying the Wilcoxon signed-rank test to series of neighborhood density and frequency growth values for the neologism and stable control sets showed the same results.

[7] Detailed results of the regression analysis and collinearity tests are included in the repository for the corresponding publication (Ryskina et al., 2020b). No evidence of collinearity was found in any of the experiments.

hotline	legroom	twentysomething	camcorder	knockoff
nachos	halogen	hypertext	cross-sex	monofilament
virtual-reality	weeknight	youre	rulemaking	switchgrass
oxytocin	gelato	acupuncturist	bioethics	roadie
belowground	impactor	biofeedback	early-onset	overrepresentation
filmmaker	off-task	interobserver	hummus	counter-terrorism
generalizability	software	connectivity	enhancer	pathogen
giftedness	porcini	bulgur	workfare	pro-democracy
sunblock	focuser	all-mountain	fictionality	self-assessment
waveform	childcare	otolaryngologist	biotech	reconfiguration
defibrillator	biofilter	musics	couscous	whiteboard
singletrack	reader-response	derailleur	blackgum	lemongrass
sunroof	countertop	waypoint	inculturation	subsidiarity
collider	frizz	uptick	home-schooling	tendinitis
big-picture	soundtrack	listserv	sobre	preschooler
blogger	encephalopathy	workspace	audiotape	prion
biotechnology	reauthorization	discriminant	goodness-of-fit	mouthwatering
radicchio	governance	bycatch	spyware	biodiversity
globalization	ethnicity	aquifer	sarcoidosis	midrange
homeschooling	ppg	neuropathy	cook-off	forensics

Table 4.2: A random sample of 10% of the neologisms selected by our method (§4.4.2). According to the Oxford English Dictionary, 82% of them had their first recorded use in their most recent nominal sense after the beginning of the HISTORICAL time period, and 58% of them were first recorded in the twentieth century.

(shown in Table 4.2) out of the 1,000 selected neologisms and look up their earliest recorded use in the Oxford English Dictionary Online (OED, 2018). Of those 100 words, eight are not defined in the dictionary: they only appear in quotations for other entries (bycatch (quotation from 1995), twentysomething (1997), cross-sex (1958), *etc.*) or do not occur at all (all-mountain, interobserver, off-task). Of the remaining 92 words, 78 have been first recorded after the year 1810 (*i.e.* since the beginning of the HISTORICAL timeframe), 44 have been first recorded in the twentieth century, and 21 words have first occurred after 1950. However, some of the words which date back to before the nineteenth century have only been recorded in their earlier, possibly obsolete sense: for example, while there is evidence of the word software being used in the eighteenth century, this usage corresponds to its obsolete meaning of 'textiles, fabrics', while the first recorded use in its currently dominant sense of 'programs essential to the operation of a computer system' is dated 1958. To account for such semantic neologisms, we can track the first recorded use of the newest sense of the word; that gives us 82, 58, and 31 words appearing since 1810, 1900, and

1950 respectively.[8] This leads us to assume that most words selected for our analysis have indeed been neologisms sometime over the course of the HISTORICAL time.

We would also like to note that the results of this examination may be skewed due to factors for which lexicography may not account: for example, many words identified as neologisms are compound nouns like countertop or soundtrack that have been written as two separate words or joined with a hyphen in earlier use. There is also considerable spelling variation in loanwords, *e.g.* cuscusu, cooscoosoos, kesksoo were used interchangeably before the form couscous was accepted as the standard spelling. Specific word forms might also have different life cycles: while the word music existed in Middle English, the plural form musics in the particular sense of 'genres, styles of music' is much more recent.

A qualitative examination of the neologism set reveals that new words tend to appear in the same topics; for example, many words in our set were related to food, technology, or medicine. This indirectly supports our second hypothesis: rapid change in these spheres makes it likely for related terms to substantially grow in frequency over a short period of time. One example of such a neighborhood is shown in Figure 4.1a: the neologism renewables appeared in a cluster of words related to energy sources—a topic that has been more discussed recently. There is also some correlation between the topic and how new words are formed in it: most food neologisms are so-called cultural borrowings (Weinreich, 2010), when the name gets loaned from another culture together with the concept itself (*e.g.* pesto, salsa, masala), while many technology neologisms are compounds of existing English morphemes (*e.g.* cyber+space, cell+phone, data+base).

We also inspect the nearest neighbors (HISTORICAL words with the highest cosine similarity) of the neologisms to ensure that the neologisms are being projected into the appropriate parts of the embedding space. Examples of such nearest neighbors are shown in Table 4.3. We found several different patterns of how the concept represented by a neologism relates to concepts represented by its neighbors. For example, some terms for new concepts appear next to related concepts they succeeded and possibly made obsolete: *e.g.* email:letter, e-book:paperback, database:card-index. Other neologisms emerge in clusters of related concepts they still equally coexist with: hip-hop:jazz, hoodie:turtleneck; most cultural borrowings fall under this type (see the neighborhood of the word pesto in Figure 4.1b). Both of these patterns can be viewed as examples of a more general trend: one concept taking place of another related one, whether it means fully replacing the older concept or just taking its place as the dominant one.

Other interesting effects we observed include lexical replacement (a new word form replacing an old one without a change in meaning, *e.g.* vibe:ambience), tendency to abbreviate terms as they become mainstream (biotech:biotechnology, chemo:chemotherapy), and the previously

[8] For all words that have one or more senses marked as a noun, we only consider the newest of those senses. Out of the 92 listed words, only three do not have nominal senses, and for two more nominal use is marked to be rare.

Neologism	Nearest HISTORICAL neighbors	
email	telegram	letter
pager	beeper	phone
blogger	journalist	columnist
sitcom	comedy	movie
spokeswoman	spokesman	director
sushi	caviar	risotto
rehab	detoxification	aftercare

Table 4.3: Nearest HISTORICAL neighbors of the projected MODERN embeddings for a sample of emerging words. We can see that the neologisms get projected into semantically relevant neighborhoods and that the nearest neighbors can even be useful for observing the evolution of concepts (*e.g.* pager:beeper).

mentioned changes in spellings of compounds (lifestyle:life-style, daycare:day-care).

4.7 Conclusion

We have shown that our two hypothesized factors, semantic neighborhood sparsity and its average frequency growth rate, play a role in determining in what semantic neighborhoods new words are likely to emerge. Our analyses provide more support for the latter, conforming with the prior linguistic intuition of how language-external factors (which frequency growth implicitly represents) affect language change. We also discovered evidence for the former, although sparsity was found less significant.

Our contributions are manifold. From a computational perspective, we extend prior research on meaning change to a new task of analyzing word emergence, proposing another way to obtain linguistic insights from distributional semantics. From the point of view of linguistics, we approach an important question of whether language change is affected by not only language-external factors but language-internal factors as well. We show that internal factors—semantic sparsity, specifically—contribute to where in the semantic space neologisms emerge. To the best of our knowledge, our work is the first to use word embeddings as a way of quantifying semantic sparsity. We have also been able to operationalize one kind of external factor, technological and cultural change, as something that can be measured in corpora and word embeddings, paving the way to similar work with other kinds of language-external factors in language change.

An admittable limitation of our analysis lies in its restricted ability to account for polysemy, which is a pervasive issue in distributional semantics studies (Faruqui et al., 2016). As such, semantic neologisms (existing words taking on a novel sense) were not a subject of this study, but they introduce a potential future direction. Exploring further properties of semantic neighbor-

hoods and their influence on language change is also a promising direction; for example, Francis et al. (2021) extend our methodology to identify the language-internal factors predictive of word decline, considering a wider array of semantic, distributional, and phonological factors. Finally, the analysis presented in this chapter is limited to only American English: future work could also test if our conclusions hold across multiple languages.

Chapter 5

Studying Neology on a Smaller Time Scale

5.1 Introduction

Hypotheses about language evolution traditionally arise from observing historical linguistic data over large periods of time. However, analyses based on historical data are inevitably subject to a certain "survival bias": linguistic innovation typically ends up recorded in historical sources such as printed books only by the time it gains widespread acceptance in the language of the community. Working with such sources, we do not get to observe the underlying evolutionary process of language change, where individual language users continuously modify the language, and then some of these modifications survive and propagate across communities (Bowern, 2019).

The rise of computer-mediated communication has given researchers access to unprecedented amounts of linguistic data and opened up new avenues for studying language change in real time (McCulloch, 2020). We can now study linguistic innovation in much more fine-grained ways, both in terms of the time scale and the size of linguistic communities. For example, social media lets us track lexical changes over the course of weeks (Eisenstein et al., 2014), compared to decades (Hamilton et al., 2016) or centuries with historical data. The availability of written online communication, as well as its diversity, lets us make inferences about the laws of language change that might not have been possible with sparser historical corpora.

Motivated by this, we propose to extend the methodology developed in Chapter 4 from printed literature to Twitter postings. In this chapter, we investigate how new words *(neologisms)* are introduced on Twitter and what semantic factors, both language-internal and language-external, influence their emergence. The contribution of this analysis is two-fold. First, it lets us test the robustness of the findings presented in Chapter 4 by empirically testing whether they hold for a new, richer dataset. Second, analyzing Twitter metadata such as precise timestamps or users'

The work presented in this chapter was done in collaboration with Vivek Kulkarni, Taylor Berg-Kirkpatrick, and other researchers at Twitter.

social connections can help shed more light on the extralinguistic facets of the process by which new words emerge.

Our study of American English literature and press from the nineteenth, twentieth, and twenty-first centuries showed that new words are more likely to emerge in the areas of the semantic space that are sparser or are growing in popularity faster (Chapter 4). In this chapter, we test the same hypotheses (§5.2) using a large corpus of English tweets (§5.3.1). Building on our prior work (Ryskina et al., 2020b), we operationalize the two hypotheses under the distributional semantics paradigm (§5.3.5) and again find evidence for both factors being correlated with word emergence (§5.4). The overall agreement between the experimental findings demonstrates the robustness of our conclusions, while a deeper analysis of their lower-level disagreements (§5.5) brings forward the contrasts between the larger-scale and smaller-scale models of language change.

5.2 Question and Hypotheses

For the analysis presented in this chapter, we adapt the supply-driven and demand-driven neology hypotheses of Ryskina et al. (2020b) (§4.3). Both hypotheses rely on the concept of a semantic space—a manifold of meanings where certain points correspond to words—and reason about *where* in such a space new words are likely to emerge, *i.e. what kinds of meanings* are likely to be expressed in new word forms. We operationalize the hypotheses under the distributional semantics paradigm (Lenci, 2018), using the word embedding space learned from co-occurrence statistics within our corpus as a proxy for the underlying semantic space.

Supply hypothesis This hypothesis proposes that neologisms are more likely to emerge in *sparser areas of the semantic space.* This hypothesis is derived from Bréal's (1904) law of differentiation, which states that the semantic space tends towards uniformity. It predicts that if two existing words are too close in meaning, they will either diverge or one of them will fall out of use. By extension, Ryskina et al. (2020b) posit that the existence of gaps in the semantic space could create pressure on the language to repair uniformity by populating them with neologisms.

Demand hypothesis The demand hypothesis states that neologisms are more likely to emerge in *semantic neighborhoods of growing popularity.* Growing popularity of a certain semantic cluster—*i.e.* growing frequency of use for the words that make it up—can be viewed as a reflection of the increasing importance of the corresponding domain of discourse. Rapidly developing domains such as technology might produce novel concepts faster, and the need for words to refer to these new concepts could also be a driving factor of neology.

5.3 Methodology

In this chapter, we apply the methods described in §4.4 to a new corpus, collected from the Twitter social media platform. This section details the creation of the dataset and the modifications we make to the original methodology of Ryskina et al. (2020b).

5.3.1 Data

We create our corpus by randomly sampling 100K tweets per day of posting from the Twitter database, spanning the period from January 1, 2007, to December 15, 2021. We rely on Twitter's internal classifiers to filter out non-English, code-switched, or machine-generated tweets: we exclude any tweets not labeled as English, any tweets from the accounts classified as bots, and any retweets. There were much fewer tweets posted in the first few years after the platform's launch in 2007, so the tweet distribution in our corpus is skewed towards more recent years: the number of tweets per year grows from around 110K in 2007–2008 to 32–35M in 2010–2021. The full corpus contains around 437M tweets.

Following §4.4.1, we split our corpus into two non-overlapping subsets, HISTORICAL (2007–2010) and MODERN (2011–2021). One major difference with the previous study is in the relative size of the subcorpora: while in §4.4.1 the HISTORICAL subset was much larger than the MODERN one, here our choice of splitting at the start of the new decade results in most of the data (396M out of 437M tweets) being allocated to the MODERN subset.

5.3.2 Neologism Selection

As we did in §4.4.2, we identify neologisms among the corpus vocabulary based on the patterns of their usage rather than relying on an external word list. However, the original heuristic of selecting words that are substantially more frequent in the MODERN subcorpus than in the HISTORICAL one produces many false positives when applied to the Twitter corpus. This could be explained by the corpora being different along many dimensions: tweets in our dataset exhibit greater dialectal diversity, are often highly informal, contain more typos and variation in spelling and capitalization, and are subject to external constraints that could incentivize creative uses of language (*e.g.* using more abbreviations and acronyms to fit under the 140/280 character limit).

Instead, we use the method introduced by Kulkarni et al. (2018), which for a given word estimates the year when it came into popular usage. It computes the cumulative usage count of the word w throughout the entire corpus $c_{total}(w)$ and then finds the first year in which the use of the word exceeded a specific threshold: $\arg\min_t[c_t(w) > \alpha \cdot c_{total}(w)]$; here, t is the timestep (*i.e.* year), and $c_t(w)$ represents the number of times the word w was used in the year t. Words that

came into popular use in 2011 or later are selected as neologisms. We tune the hyperparameter α by optimizing the recall of our selection procedure with respect to a list of recent neologisms identified by Würschinger (2021).[1] The final value of α is empirically set to 0.01.

Unlike Ryskina et al. (2020b), we do not limit our analysis to a particular part of speech, but we still use part-of-speech tagging to exclude non-generic proper names (*e.g.* Trump). We use the Flair English POS tagger[2] (Akbik et al., 2018); since tagging is time-consuming, we run it only on a random 1% sample of the HISTORICAL tweets. We discard any words for which the most frequent tag was 'NNP' or 'NNPS', and also all tokens shorter than 3 characters, all tokens containing emoji, numbers, or punctuation marks except for hyphens and apostrophes, and all hashtags. To remove rare variants, we rank the remaining tokens by their frequency in the entire corpus and choose the first 10,000 words to represent our final set of neologisms.

5.3.3 Embeddings

We use Word2Vec SkipGram embeddings (Mikolov et al., 2013) learned from the Twitter data to identify which words form semantic neighborhoods. Following §4.4.3, we learn separate embeddings for the HISTORICAL and the MODERN subcorpora and align them using an orthogonal Procrustes transformation. The alignment step is necessary for finding the neighborhoods in the HISTORICAL space where neologisms eventually appear: as most neologisms are not in the vocabulary of the HISTORICAL Word2Vec model, we approximate their positions in the HISTORICAL space by projecting their MODERN vectors into the same axes using the aligning transformation.

We use the embedding hyperparameters as specified in §4.4.3, with one difference: instead of specifying the minimum occurrence count, we automatically adjust it based on the desired vocabulary size (100K words for either model).

5.3.4 Control Set Selection

As we did in the previous chapter, we select a set of control words to compare against the neologisms. Similar to §4.4.4, we pair each neologism with a control word while controlling for word length and frequency, which are known confounding factors in distributional semantics. In addition, we also ensure a tighter semantic correspondence within each neologism–control pair, pairing only words with high enough similarity between their embedding vectors.

Formally, for each neologism w_n we select a counterpart w_c satisfying the following constraints:

[1] https://github.com/wuqui/sna/blob/master/out/df_comp.csv

[2] https://huggingface.co/flair/pos-english

- Frequency ranks of the two words in the corresponding corpora are in the same percentile: $|z_m(w_n) - z_h(w_c)| \leq 1000$. Here $1 \leq z_h(\cdot), z_m(\cdot) \leq 100000$ are ranks of the words in the vocabularies of the HISTORICAL and MODERN models respectively, sorted by frequency;
- The length of the two words is identical up to 3 characters;
- The cosine similarity between the neologism and its control counterpart in the HISTORICAL embedding space is above a certain threshold: $\text{cosine}\left(v_{w_n}, v_{w_c}\right) \geq 0.4$. Here v_w denotes a projected MODERN embedding if w is a neologism or a HISTORICAL embedding if w is a control word.

Under these strict constraints, we are able to find control counterparts only for 1,147 of 10,000 neologisms. Table 5.1 shows ten of the resulting neologism–control pairs. We find that the addition of the cosine similarity constraint often gives the resulting word pairs an extra semantic or syntactic connection: conceptual similarity (snapchat:youtube, tinder:msn), matching part of speech and inflection (accs:dms, stanning:unfollowing), or even morphological relatedness (relatable:relate).

Neologism	Control
snapchat	youtube
coronavirus	government
mutuals	tumblr
selfie	twitpic
moots	fellow
tinder	msn
accs	dms
relatable	relate
notifs	dm's
stanning	unfollowing

Table 5.1: Control words (right) paired with the ten most frequent neologisms (left; excluding the neologisms that could not be paired with controls).

Following §4.4.4, we perform the pairing process only to ensure that the selected neologisms and control words are similarly distributed. We do not perform pairwise comparisons but instead contrast the entire set of neologisms with the entire set of control words.

5.3.5 Experimental Setup

Building on the experimental setup outlined in §4.4.5, we verify the hypotheses by comparing certain statistics of the neighborhoods of the neologisms in our sample with those of the neighborhoods of the corresponding control words. The neighborhood of a word w in the HISTORICAL

embedding space is defined as the set of the HISTORICAL words for which the cosine similarity between their HISTORICAL embeddings and the word vector v_w exceeds the given threshold τ; v_w is a HISTORICAL embedding if w is a control word and a projected MODERN embedding if w is a neologism.

The neighborhood properties we are interested in are sparsity (linked to supply) and increase in popularity (linked to demand). Following §4.4.5, we quantify them in terms of the neighborhood's density and the average frequency growth rate of the words in the neighborhood. Formally, for a neighborhood of a word w (as defined by the cosine similarity threshold τ):

1. *Density of a neighborhood* $d(w, \tau)$: number of words that fall into this neighborhood: $d(w, \tau) = |\{u : \text{cosine}(v_w, v_u) \geq \tau, u \neq w\}|$
2. *Average frequency growth rate of a neighborhood* $r(w, \tau)$: mean linear regression slope for each neighbor word's yearly occurrence counts $c_t(\cdot)$ in the HISTORICAL subcorpus vs. the year $t \in \{2007, 2008, 2009, 2010\}$:

$$r(w,\tau) = \frac{1}{d(w,\tau)} \sum_{u:\text{cosine}(v_w,v_u)\geq\tau} \left[\frac{\sum_{t=2007}^{2010} \left(t - \frac{1}{4}\sum_{t=2007}^{2010} t\right)\left(c_t(u) - \frac{1}{4}\sum_{t=2007}^{2010} c_t(u)\right)}{\sum_{t=2007}^{2010} \left(t - \frac{1}{4}\sum_{t=2007}^{2010} t\right)^2} \right]$$
$$= \frac{1}{d(w,\tau)} \sum_{u:\text{cosine}(v_w,v_u)\geq\tau, u\neq w} 0.1\left[-3c_{2007}(u) - c_{2008}(u) + c_{2009}(u) + 3c_{2010}(u)\right]$$

The definition of $r(w, \tau)$ used here is different from the one used in §4.4.5: with the HISTORICAL time period spanning only 2007–2010 in these experiments, the sample size of four is no longer enough to reliably compute Spearman correlation coefficient. For a fairer comparison between how new words appear on Twitter vs. in historical published literature, we also repeat the experiments of Chapter 4 using the updated methodology (§5.4.2).

In the experiments described in this chapter, we compute the mean values of these two metrics for the neighborhoods of neologisms and the neighborhoods of control words over a range of neighborhood sizes defined by the threshold τ. As in §4.4.5, we cap the neighborhood size at 5,000 and exclude the neologisms and controls themselves from their neighborhoods. Some neologisms end up being excluded from their neighborhoods twice: as their projected MODERN vector, which we use to define the neighborhood, and as their HISTORICAL vector if it exists and falls within the neighborhood.

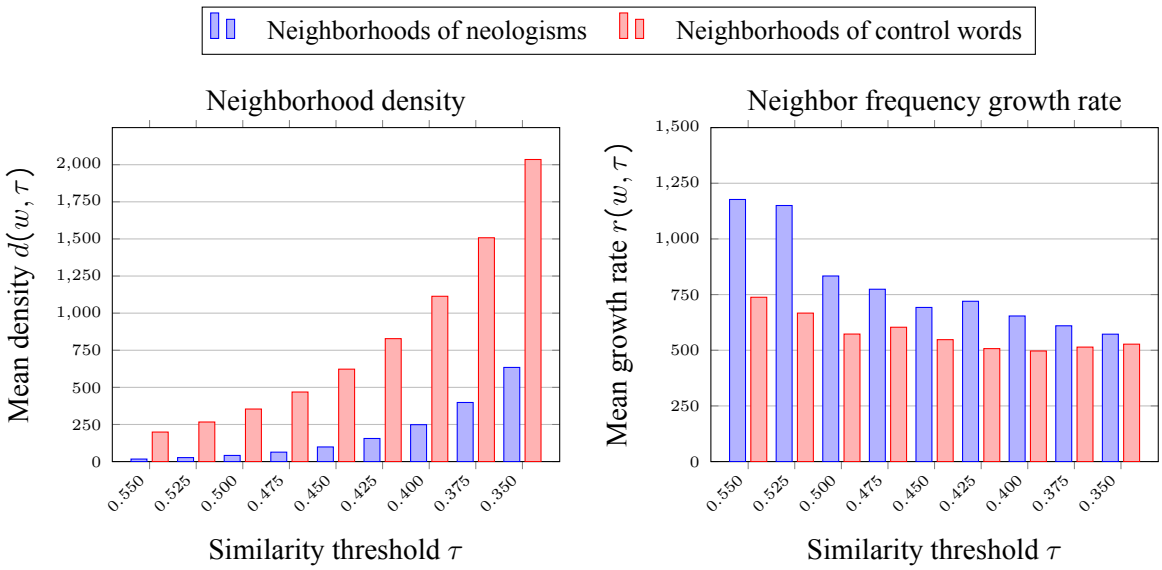


Figure 5.1: Experimental results for the Twitter corpus. **Left:** Number of HISTORICAL word vectors within the cosine distance $1 - \tau$ of a given word's vector, averaged across the neologism (blue) and the control word (red) sets. Neighborhoods of the projected neologism vectors are sparser on average than those of the control word vectors. **Right:** Average use growth rate (represented by the mean linear regression slope) of those HISTORICAL words, averaged across the neologism (blue) and the control word (red) sets. Neighbors of the projected neologism vectors exhibit higher rates of growth than the neighbors of the control word vectors.

5.4 Results

5.4.1 Twitter Data

Figure 5.1 shows how the mean neighborhood density (left) and the mean neighbor frequency growth rate (right) differ between neologisms and controls over a range of neighborhood sizes. The overall trends we observe are similar to the findings of Chapter 4: neighborhoods in which neologisms appear tend to be sparser than the neighborhoods of control words (lower density), but have higher frequency growth rates. However, the relative contribution of the two factors is now reversed: the disparity in density between the neologisms and the controls is more noticeable than the disparity in frequency growth (compare to Figure 4.2). To verify that this is in fact explained by the differences between the corpora and not by the changes made to the hypothesis formalization or minor experimental details, we run the same experiment on the data from Chapter 4, as described in the next section.

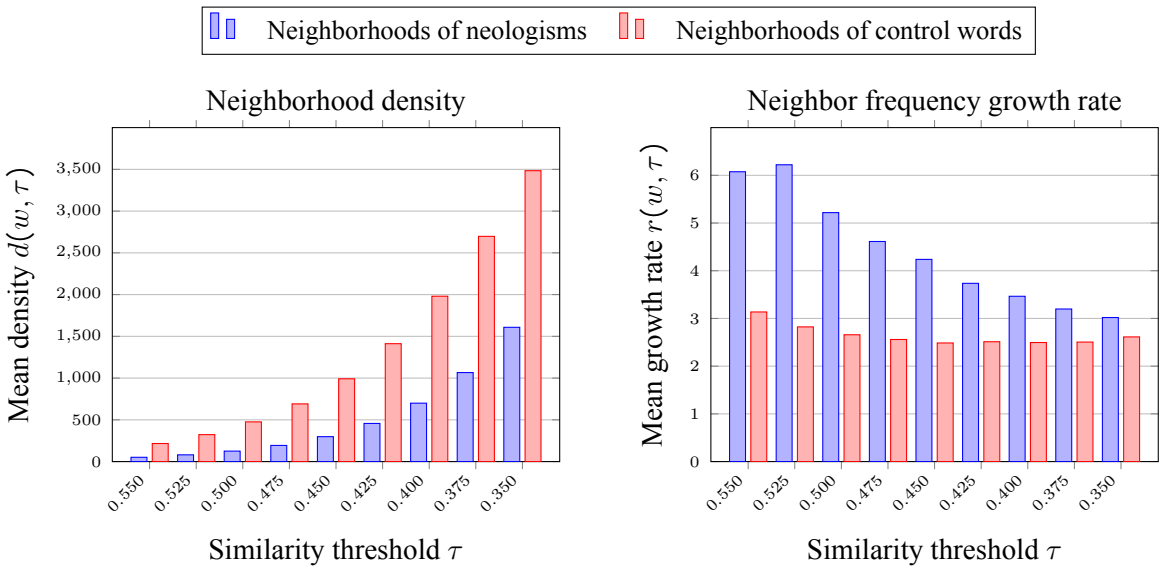


Figure 5.2: Experimental results for the COHA/COCA reproduction experiment, confirming the findings of Chapter 4: neighborhoods of the projected neologisms are sparser and exhibit a higher growth rate. **Left:** Number of HISTORICAL word vectors within the cosine distance $1 - \tau$ of a given word's vector, averaged across the neologism (blue) and the control word (red) sets. **Right:** Average use growth rate (represented by the mean linear regression slope) of those HISTORICAL words, averaged across the neologism (blue) and the control word (red) sets.

5.4.2 COCA/COHA Data

To test if the results of Chapter 4 hold under out updated methodology, we also run our analysis with the COHA corpus (Davies, 2002) as HISTORICAL data and the COCA corpus (Davies, 2008) as MODERN. In this experiment, we use the list of the 1,000 neologisms identified by comparing their frequencies in COHA and COCA (§4.4.2). All of these neologisms are nouns since the analysis of Chapter 4 was restricted to nouns only, but in this experiment, we impose no such restriction on the control words or the semantic neighbors. We newly pair them with the control words using the updated constraints (§5.3.4) and compute the density and the frequency growth rate for the resulting 774 neologism–control neighborhood pairs under the updated definitions (§5.3.5).

The results of this reproduction study are shown in Figure 5.2 (compare to Figure 4.2). Again, we find evidence for both the demand and the supply hypotheses: neologisms have fewer neighbors than the control words but their neighbors grow in frequency faster. We also reproduce the dynamics we saw in our previous experiment: neologisms and control neighborhoods are closer in terms of growth rate than in terms of density, the opposite of what we observed in §4.5. The potential reasons for this change in behavior are outlined in the section below.

5.5 Discussion

Both experiments in this chapter confirm what we observed in Chapter 4: we again find evidence for both the supply hypothesis (sparsity playing a role in neology) and the demand hypothesis (growing popularity being predictive of neology). This shows the robustness of the conclusions presented in the previous chapter: our observations stay consistent when we extend our analysis to a new dataset or make changes to the methodology, *e.g.* altering the operationalization of the hypotheses or modifying the experimental setup. The fact that we see similar trends in historical published data (§5.4.2) and on social media (§5.4.1) suggests that they may be a reflection of more general laws of neology and language change.

Although the neighborhood densities are of comparable magnitude for the embedding spaces learned from Twitter (Figure 5.1, left) and from the historical published literature (Figure 5.2, left), their frequency growth rates are very different: the mean slope of the linear regression fit to word occurrence counts is 2–6 for COHA (Figure 5.2, right) and 500–1,200 for Twitter (Figure 5.1, right). This is mainly an artifact of the rapid growth in Twitter usage throughout our chosen HISTORICAL timeline, but also a possible indicator of the increased speed of language change on social media. Platforms like Twitter allow for much faster and wider dissemination of written linguistic innovation than was possible previously, and the lowered barrier to entry to the corpus (*i.e.* the accessibility of publishing via social media) makes the users' linguistic expression less restricted and lets us capture its diversity more fully.

Comparing Figure 5.2 and Figure 4.2, we can see that in the reproduced results the difference in density is more substantial than the difference in frequency growth, while in the original experiment we observed the opposite. We attribute this change to three methodological differences: (1) how the embedding spaces models were trained (capping the vocabulary size might create a greater number of sparser neighborhoods), (2) allowing all parts of speech vs. only nouns in the vocabulary (nouns might behave differently in terms of both their distribution in the space and the dynamics of their use), and (3) the relaxation of the stability constraint, which ensured that the control words exhibit no significant growth or decline (as shown in Table 4.1, both frequency growth and density were found to be significant for a relaxed control set). Surprisingly, the addition of the cosine similarity constraint to the neologism–control pairing process does not reduce the gap in neighborhood densities between neologisms and controls, even though it guarantees an overlap between the paired neighborhoods at $\tau \leq 0.4$. We also observe that the frequency growth differs more substantially for COHA than for Twitter (Figure 5.2, right vs. Figure 5.1, right), while density differs more substantially for Twitter than for COHA (Figure 5.2, left vs. Figure 5.1, left): this also aligns with the findings of Chapter 4 and shows another potential axis of difference between social media and historical printed corpora.

5.6 Conclusion

In this chapter, we extend the methods introduced in Chapter 4 to a dataset of tweets and reproduce the findings of Ryskina et al. (2020b) on both the new data and the data used in the original study. Our experimental results show the robustness of our supply-and-demand framing of neology and let us explore the similarities and the differences between how new words emerge in literature vs. on social media. Future work on neology using Twitter data could incorporate more social science variables into the model, such as the user's social status or network connections: this would allow us to study neology on a much smaller scale (*e.g.* new words becoming established in a vocabulary of a small group of friends rather than the entire community of users) as well as gain a more nuanced understanding of the role that extra-linguistic factors (currently represented by frequency growth only) play in neology.

Part III

Applying Morphology to Novel Lexemes

Chapter 6

Nearest-Neighbor Morphological Inflection for New Lemmas

6.1 Introduction

As discussed in Chapter 4, new word emergence is a major component of language change. As language users acquire novel lexical items and start using them in new contexts, they often have to modify the word form so that it manifests the desired grammatical features. For example, as the verb google entered the English language, speakers came up with the corresponding past tense form googled by applying the regular English past formation rule. In producing such forms despite never having been exposed to them before, users generalize known morphological patterns to fill in the unknown cells in the new word's *inflectional paradigm*. Over the span of a neologism's life cycle, from emergence to becoming established in mainstream vocabulary, it typically acquires a dominant, majority-preferred inflected form for each slot in the paradigm, although language variation is still present.[1]

Neology also presents a challenge for modern natural language processing models. As we process the text from domains such as social media, where language change happens rapidly, we want our models of morphology to seamlessly generalize to the novel words that they might encounter, both in comprehension (morphological tagging) and production (reinflection needed for language generation). However, recent work shows that modern morphological inflection models still generalize poorly to lemmas not seen in training (Goldman et al., 2022). The first

The work presented in this chapter was done in collaboration with Matthew R. Gormley and Taylor Berg-Kirkpatrick.

[1] This includes both free variation (*e.g.* Finnish omena 'apple' + GEN;PL → equally acceptable omenoiden, omenoitten, omenojen, omenien, omenain; Gorman et al., 2019) and non-prescriptive inflections (*e.g.* Polish podnieść się 'rise' + PL;3;FUT → standard podniosą się and colloquial podniesą się; Pimentel et al., 2021).

step towards enabling the state-of-the-art inflectors to handle neologisms in human-like ways is to ensure they can successfully generalize in a 'pseudo-wug test' scenario, where they are tested on lemmas that already have agreed-upon paradigms.

In this chapter, we propose augmenting the morphological inflection models with a retriever component, which for every test data point selects a relevant exemplar from the training set. Building on recent work on analogy-guided inflection (Liu and Hulden, 2020), we hypothesize that appending an exemplar from the correct inflection class to the input would bias the model towards applying the appropriate inflection rule and generating the correct output. We explore several methods for exemplar retrieval, relying on orthographic, semantic, or phonological similarity, and find phonology- and orthography-based analogies most helpful for generalization. Although none of our retrievers outperform the simpler data augmentation baseline across the board, we show that pairing each test instance with a perfect exemplar does increase inflection accuracy, suggesting that analogy-based approaches are a promising direction for improving morphological generalization to new lemmas.

6.2 Background

Morphological inflection In this chapter, we focus on the default inflection task, in which morphological features (*e.g.* past tense of a verb, or V;PST) are applied to the base uninflected form (also called a *lemma*; *e.g.* sing) to produce the corresponding inflected form (sang). All of the lemma's inflected forms form a *paradigm*, where each *cell* corresponds to a set of morphological features. In the computational community, the annual SIGMORPHON shared tasks on morphological inflection and reinflection (a version of the task where the contents of a cell are predicted from another cell in the same paradigm) (Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019; Vylomova et al., 2020; Pimentel et al., 2021) and the UniMorph database (Kirov et al., 2018; McCarthy et al., 2020; Batsuren et al., 2022) provided convenient benchmarks for monitoring the evolution of the state of the art. Over the years, finite-state inflectors (Gorman and Sproat, 2021; Beemer et al., 2020) had been surpassed by RNN encoder–decoder models (Kann and Schütze, 2016a,b; Bergmanis et al., 2017), RNN hybrids with FST-style alignment (Aharoni and Goldberg, 2017; Makarov et al., 2017; Makarov and Clematide, 2018; Wu et al., 2018; Wu and Cotterell, 2019), and eventually Transformers (Wu et al., 2021; Canby et al., 2020). With the SOTA reaching near-perfect accuracy on the standard splits, the research focus within the community has been shifting towards creating more challenging benchmarks to test the models' generalization capabilities.

Morphological generalization in humans Humans' ability to generalize their knowledge of morphological rules to novel words has long been of interest to psychologists. In the famous wug test, Berko (1958) tested the ability of English-speaking children to perform nonce word inflection, such as forming the plural form of the made-up noun wug. It has been shown that children can use their implicit knowledge of morphophonology to select the correct morpheme (*e.g.* the suffix -s for English plural noun forms) and allomorph (wug + /z/ rather than *wug + /s/) even at a very young age (Berko, 1958). There have since been many psycholinguistic studies probing human subjects' ability to inflect nonce words across a variety of languages, mostly focusing on investigating regular rule acquisition in children (*e.g.* Dąbrowska and Szczerbiński, 2006) or probing the cognitive plausibility criteria with deliberately ambiguous stimuli, such as the English past tense debate (spling + V;PST → splinged/splung; Albright and Hayes, 2003) or the German plural formation for nouns (Marcus et al., 1995). Those tests assess only the *plausibility* of produced inflections: in order to avoid memorization effects, such experiments use made-up stimuli which the subjects could not have encountered before in their linguistic interactions, so by design, there is no prescriptive or majority-preferred inflectional paradigm for these nonce words. The absence of a dominant inflection that can be treated as ground truth makes it difficult to extend wug testing to computational models, although there are studies on comparing the distributions of nonce word inflections between models and human subjects (Corkery et al., 2019; McCurdy et al., 2020).

Morphological generalization in models Because evaluating the inflections of actual nonce words is challenging, in NLP wug tests are often simulated by testing on held-out lemmas for which the correct answer is assumed to be known (Liu and Hulden, 2022). Generalizing to lemmas never seen in training is a known challenge for morphological inflection models: although the state-of-the-art systems have demonstrated impressive performance on most languages in the recent shared tasks (Vylomova et al., 2020; Pimentel et al., 2021), it has been shown that resplitting the data to have no lemma overlap between the train and test sets leads to a drop in accuracy, most notably for languages with fewer training examples available (Goldman et al., 2022). Some of the errors of these top-performing systems are human-like (*e.g.* predicting an inflection that is plausible but not attested or attested but not listed as ground truth), but often enough the errors are not interpretable in terms of the morphology of the language (like outputting membled as a past tense of the verb mail),[2] showing that the models still struggle even with fairly regular rules.

The 2022 iteration of the SIGMORPHON shared task (Kodner et al., 2022) focused specif-

[2]The cited error is actually produced by an older system by Rumelhart and McClelland (1986) and pointed out by Pinker and Prince (1988), but Gorman et al. (2019) show that SOTA models still make similar mistakes in languages other than English.

ically on testing how well models generalize to unseen lemmas and feature sets, which again proved to be challenging for both neural and non-neural models. The winning system used data hallucination and student-forcing (Yang et al., 2022), while Elsner and Court (2022) proposed an exemplar-augmented model similar to ours.

Retrieval-augmented models in NLP Recent work on text generation tasks such as language modeling or translation has explored retrieving relevant exemplars from a datastore for more explicit reliance on memorization (Khandelwal et al., 2019, 2020). Augmenting inputs with items nearest to them in a certain space has been shown to improve performance on these tasks. In morphological inflection, analogy-guided models have been proposed, where exemplars are chosen uniformly from the training paradigms (Liu and Hulden, 2020); recent work also experimented with adding feature labels to inform the model how well the exemplar is likely to match the test instance (Elsner and Court, 2022). In this chapter, we propose using nearest neighbors as exemplars for morphological inflection, with the distance metric reflecting lemma similarity in terms of phonology, orthography, or semantics.

6.3 Proposed Method

In the traditional supervised setup, we expect the inflection model to infer the morphological rules from the training data and apply them to the new inputs at test time. The fact that these models perform successfully on form-split data but struggle to generalize to entirely new lemmas suggests that the difficulty of the lemma-split test case might be in placing the lemma into the correct inflection class. For example, if paradigms like (ring, rang, rung) and (wing, winged, winged) occur in the training set, is it is much easier for the model to predict that a new verb sing is irregular, *i.e.* that its past tense is sang rather than singed, if it has already seen that sing + V.PTCP;PST → sung. To overcome this hurdle, we propose using analogy to guide the model: for each test instance, we append an exemplar from the training set to the input, such as sing + V;PST | [ring + V;PST = rang] → sang.

Our goal is to select an exemplar that falls into the same inflection class as the test lemma. We assume that lemmas that fall into the same class are similar in some way, so we define a range of similarity metrics and select exemplars that are closest to the test lemmas in terms of the chosen metric. We consider the following dimensions of similarity:

- **Phonology:** the word's phonological form has been shown to be correlated with its syntactic categories like part of speech or grammatical gender (Kelly, 1992; Monaghan et al., 2005), which in turn affect the word's inflectional paradigm. Since we only have access to the orthographic forms of the lemmas, we use automatic grapheme-to-phoneme (G2P)

conversion (Epitran; Mortensen et al., 2018) to obtain their IPA representations and then measure the Levenshtein distance between these IPA strings to determine the nearest neighbor for each test time instance.

- **Orthography:** orthographic form can considered a proxy for phonological form (Williams et al., 2020), although the connection is less reliable in languages with less transparent orthographies. However, one advantage of selecting exemplars by orthographic form directly is that it is less brittle: if errors are introduced at the G2P conversion step, they might affect the nearest neighbor choice and hurt the downstream inflection accuracy. We explore two ways to formalize orthographic similarity between a pair of lemmas: negative Levenshtein distance and the length (in characters) of their longest common suffix. The latter is tailored to languages whose inflectional morphology favors suffixing; 55% languages listed in the World Atlas of Language Structures (WALS; Dryer, 2013) exhibit this property.
- **Semantics:** although the relationship between semantics and inflectional morphology is still an open question in linguistics, recent computational work has found a correlation between meaning and inflection class (Williams et al., 2020) and has shown that training on semantically relevant instances improves the performance of morphological inflectors on certain non-Indo-European languages (Goldman and Tsarfaty, 2021). In conjunction with orthography, meaning could also shed light on morphological relatedness: for example, if we know that mishave—an actual English lemma listed in UniMorph—is semantically closer to have than to shave, it is easier to guess that its past tense is mishad rather than mishaved. We use pretrained fastText models (Bojanowski et al., 2017; Grave et al., 2018) to generate lemma representations in the semantic space and formalize their semantic similarity as the cosine similarity between their embeddings.

6.4 Experiments

6.4.1 Data

Benchmark dataset We are working with a supervised framing of the morphological inflection task, with the data provided in the format of triplets consisting of the lemma, morphological tags, and the corresponding inflected form (*e.g.* ring + V;PST → rang). Following Goldman et al. (2022), we use the data from the SIGMORPHON 2020 shared task dataset (Vylomova et al., 2020). Out of the 90 languages used in the shared task, we select 12 for which Goldman et al. (2022) observed a noticeable discrepancy between the form-split and lemma-split inflection accuracy: ben, ceb, hin, kaz, kir, mlt, orm, sna, swa, tgk, tgl, zul. Three of these languages are from the Niger–Congo family (sna, swa, zul) and all except for Hindi (hin) are low-resource

(*i.e.* have <10,000 training triplets; Tajik (tgk) has as few as 53), the two categories observed to have the biggest drop in accuracy in the lemma-split setting (Goldman et al., 2022).

Data augmentation For all languages, we also generate additional training instances using the hallucination method of Anastasopoulos and Neubig (2019). Their approach computes character alignment between the lemma and the inflection to identify the stem, and then randomly samples replacements for inner characters of the stem to generate new lemma–inflection pairs. We synthesize 10,000 additional training instances based on the original training set, ensuring no repetition or overlap with the test set.

We also use lemma copying to generate additional data: for each lemma l in the training set, including the hallucinated ones, we add a $l + \textsf{COPY} \rightarrow l$ triplet to the training data (Anastasopoulos and Neubig, 2019; Liu and Hulden, 2022). This is designed to strengthen the copying bias in the inflector. Concurrently with our work, Yang et al. (2022) showed that augmenting with both hallucination and lemma copying improves the inflection accuracy on new lemmas.

6.4.2 Models

Base models Our first inflection baseline is the character-level Transformer of Wu et al. (2021), which was one of the top-performing systems in the SIGMORPHON 2020 and 2021 shared tasks on morphological reinflection (Vylomova et al., 2020; Pimentel et al., 2021).[3] Following Goldman et al. (2022), we also use a character-level LSTM encoder–decoder baseline, with a one-layer bidirectional encoder and a one-layer decoder.[4] The LSTM takes in a concatenation of the lemma characters and the tags, separated by a special character, and is trained to output the inflected form character by character: s i n g $ V PST → s a n g.

Exemplar augmentation As described in §6.3, our proposed approach is to append an exemplar filling the same paradigm cell to the input at both training and test time. We simply concatenate both the exemplar lemma and its inflected form to the input, separated by special characters; for the sing + V;PST | [ring + V;PST = rang] → sang example discussed above, the input and output would take the following form:

s i n g $ V PST & r i n g # r a n g → s a n g

To ensure that the model learns to rely on the exemplar and transform the input lemma analogously, at training time we generate artificial exemplars that are guaranteed to belong to the same

[3] We use the authors' implementation: https://github.com/shijie-wu/neural-transducer. All hyperparameters and architecture details follow Wu et al. (2021).

[4] We use the authors' implementation: https://github.com/OnlpLab/LemmaSplitting. All hyperparameters and architecture details follow Goldman et al. (2022).

inflection class. We synthesize an exemplar from each training input–output pair using the data hallucination method of Anastasopoulos and Neubig (2019) (§6.4.1). Our hallucination generator substitutes each stem character in both the lemma and the inflected form with probability 0.25, and we make sure that the exemplar differs from its source instance in at least one character.

At test time, we rank all candidates (training set lemmas for which the target paradigm cell is filled) according to the chosen similarity metric and select the top candidate as the exemplar. As discussed in §6.3, our base similarity metrics include: (1) negative character-level Levenshtein distance between the test lemma and the candidate (orthographic); (2) length of the longest common suffix shared between the lemma and the candidate (orthographic); (3) cosine similarity between the fastText embedding of the lemma and the candidate (semantic); and (4) negative Levenshtein distance between the Epitran-generated IPA string representations of the lemma and the candidate (phonological). We also experiment with combining orthographic and semantic criteria via reranking: for each test lemma, we select top-10 candidates under the Levenshtein distance metric and then choose the one that is closest to the lemma in the embedding space. For the orthographic and phonological metrics, exemplars can be selected from either the original training instances or the hallucinated ones (§6.4.1); for the embedding-based retriever, we limit the candidates to real (non-hallucinated) words only. In the case of the reranker, the initial top candidates produced by the Levenshtein retriever could include hallucinated lemmas: while we cannot make any claims about what such randomly generated character strings mean, we can still obtain their embeddings under the fastText model based on their n-gram makeup and rerank them by the embedding similarity. For every metric, if the initial set of candidates is empty (*i.e.* this combination of morphological features was never seen in training), we use the lemma itself as both the input and output parts of the exemplar.

Oracle baseline We also test the 'oracle' versions of the augmented models to evaluate whether exemplar augmentation would help if we had a perfect exemplar for each test instance. In this experiment, we synthesize exemplars via data hallucination not only at training time, but at test time as well. We refer to this approach as 'oracle' because it uses the ground-truth inflected forms, which we normally do not have access to at test time.

6.5 Results and Analysis

Tables 6.1 and 6.2 show the performance of the LSTM and the Transformer inflectors respectively, with and without retriever augmentation. The second column in either table (Lemma-split) shows the baseline accuracy of the non-augmented model trained on the lemma split of Goldman et al. (2022). The third column (+Hall.+Copy) shows the same models trained on the larger train-

Lang.	Base accuracy		Nearest-neighbor (Δ)					
	Lemma-split	+Hall.+Copy	Lev.	Suffix	Embed.	Rerank.	Epitran	Oracle
ben	23.96	77.34	−6.52	−16.96	−15.89	−11.62	−4.27	+0.36
ceb	17.50	78.33	−12.50	−15.00	−10.00	−0.83	−3.33	+14.17
hin	32.66	77.88	−3.91	−10.12	−12.45	−8.96	−4.95	−2.31
kaz	24.65	43.71	−5.26	**+14.55**	−11.60	−6.95	−0.52	+13.52
kir	9.73	71.16	−3.66	−10.18	−33.57	−9.73	−14.91	+7.59
mlt	11.90	54.96	−5.95	−3.40	−18.13	−9.07	−3.69	−3.12
orm	5.30	89.90	−26.77	−5.81	—	—	−5.81	+4.29
sna	30.98	73.73	**+8.62**	−5.30	—	—	**+14.11**	+18.03
swa	37.24	97.65	−8.57	−1.32	−6.83	−2.14	**+0.51**	−6.02
tgk	18.75	87.50	−6.25	0.00	0.00	0.00	−6.25	0.00
tgl	17.05	38.95	−12.00	−22.32	−18.74	−14.95	−9.48	+26.73
zul	17.78	68.89	**+3.33**	**+1.11**	—	—	**+2.22**	+10.00
Avg.	20.63	71.03	−6.62	−6.23	−14.13	−7.14	−3.03	+6.94

Table 6.1: Experimental results for the LSTM-based inflection models. Columns 2 and 3 display inflection accuracy when training on the original lemma-split data or on the training set augmented with hallucination and copying respectively. The following columns show how accuracy increases or decreases compared to column 3 when each input is augmented with an exemplar (nearest neighbor by the Levenshtein distance, longest common suffix, embedding distance, embedding-distance reranking of the top candidates by the Levenshtein distance, and Levenshtein distance in the IPA space respectively). **Bold** marks the cases where the retriever-augmented models outperform the hallucination and copying baseline; "—" indicates languages for which pretrained embeddings were not available. The final column shows the skyline performance achieved by the oracle augmentation, where the exemplar is synthesized based on the gold output.

ing set augmented with hallucination and copying: for both the LSTM and the Transformer, this increases the average accuracy by about 50 points, echoing the findings of Yang et al. (2022). This suggests that simply increasing the size of the training set and strengthening the copying bias is enough to substantially boost performance on low-resource languages. Notably, some languages are harder than others for both the LSTM and the Transformer: for example, Tagalog (tgl) reduplication is difficult for sequence-to-sequence models to learn, and augmenting the training set with synthetic data does not address this problem.

The following group of columns shows the change in accuracy (as compared to column 3) resulting from the addition of the different exemplar augmentation mechanisms. On average, our proposed retrievers decrease the inflection quality, with the semantic one being the weakest for both base models. However, individual languages can benefit from orthographic or phonological exemplar augmentation: performance improves the most consistently for Zulu (zul) under the

Lang.	Base accuracy		Nearest-neighbor (Δ)					
	Lemma-split	+Hall.+Copy	Lev.	Suffix	Embed.	Rerank.	Epitran	Oracle
ben	32.03	82.21	−8.42	−19.8	−21.95	−12.69	−8.19	−1.07
ceb	24.17	82.50	−3.33	−11.67	−4.17	−2.50	−5.00	+10.83
hin	59.96	88.02	−0.81	−14.91	−3.73	−2.35	**+0.88**	+5.58
kaz	16.20	72.02	−17.98	−31.22	−35.26	−23.52	−15.92	+10.19
kir	26.52	81.34	−28.04	−30.71	−57.32	−31.61	−21.07	+9.46
mlt	29.18	58.92	−3.12	−3.68	−15.58	−0.57	−5.67	+3.40
orm	22.22	93.18	−0.50	−7.32	—	—	−4.29	+5.56
sna	30.20	86.86	**+6.08**	**+0.20**	—	—	**+2.03**	+7.25
swa	33.47	98.06	−1.33	−1.43	−0.71	−0.41	−5.10	−5.71
tgk	25.00	87.50	0.00	0.00	0.00	0.00	0.00	0.00
tgl	39.58	44.63	−14.11	−22.95	−24.63	−17.47	−12.21	+39.37
zul	31.11	78.89	−18.89	−12.22	—	—	−16.67	+12.22
Avg.	30.80	79.51	−7.54	−12.98	−18.15	−10.12	−7.60	+8.09

Table 6.2: Experimental results for the Transformer-based inflection models. The column order is the same as in Table 6.1: "Lemma-split" shows the baseline accuracy, "+Hall.+Copy" corresponds to the hallucination and copying-augmented base model, and the following columns show gains or drops in accuracy compared to "+Hall.+Copy" yielded by various exemplar retrievers. An oracle exemplar model is included to show skyline accuracy.

LSTM base model and for Shona (sna) under either LSTM or Transformer. For the LSTM, the phonological retriever (Epitran) outperforms its orthographic and semantic counterparts, and for the Transformer, it is tied with the Levenshtein retriever (Lev.); this demonstrates that, as expected, for our selection of languages the phonological form (direct or approximated through orthography) is more correlated with inflection class than the meaning of the word.

It may seem surprising that the embedding-based exemplars did not improve performance for Swahili (swa), a Bantu language that has a semantic system of noun classes.[5] This can be explained by the fact that the Swahili data includes paradigms only for verbs, which do not exhibit the same regularity in how meanings are paired with forms.

The final column in both tables shows the gains from using the oracle exemplar augmentation: compared to the data augmentation baseline, it yields additional 6.94 and 8.09 accuracy points for the LSTM and the Transformer respectively. This shows that although our proposed retrievers did not manage to reliably increase accuracy, exemplar augmentation could in principle improve generalization to novel lemmas, so future work could focus on designing retrievers that are more suited for this task.

[5] Pretrained fastText embeddings were not available for the other two Bantu languages in our set, sna and zul.

6.5.1 English Error Analysis

In order to better understand what kinds of errors modern inflection models make in the lemma-split scenario, we run the same experiment on the English data from the same shared task, split by lemma according to Goldman et al. (2022). However, the English training set is large enough (~80K instances) for the Transformer model to achieve an accuracy of 96.91 even on the lemma-split data. For a more sensible approximation of the model's behavior for our selection of languages, we emulate a low-resource scenario for English by randomly sampling 350 paradigms from the lemma-split training set. This yields a set of 1,758 instances, which is the median training set size across our twelve-language sample.

The Transformer model trained on this smaller subset still achieves an accuracy of 84.32, which is substantially higher than any of our previous per-language results (Table 6.2, column 2). We attribute this to the relative simplicity of the English verbal morphology: an average paradigm in our English data contains 5 cells, while the mean paradigm size across our twelve test languages is 43 cells (ranging from 1 in the tgk data to 201 in hin).

Certain phonological effects at the morpheme boundaries pose challenges for the English inflector: relevant predictions include cruel + V.PTCP;PRS → *crueling instead of cruelling or tie-dye + V;SG;3;PRS → *tie-dies instead of tie-dyes. Under the error taxonomy of Gorman et al. (2019), these could be considered allomorphy errors, where the model misapplies an existing inflection rule. A human L2 English learner could plausibly make the same mistakes as well, especially in cases with highly unusual inflection mechanisms such as magic + V;PST → magicked (predicted form *magiced).

Another frequent type of allomorphy error stems from the model incorrectly classifying verbs as regular or irregular. The Transformer errs a fair amount in either direction, either predicting irregular past forms (tend + V;PST → *tent) instead of regular ones (tended) or vice versa (break + V.PTCP;PST → *breaked instead of broken).

Most other incorrect predictions fall into the "silly" error category per Gorman et al. (2019). Specifically, the inflector frequently modifies the stem of the verb, deleting repeated consonants (dampproof + V;NFIN → *damproof) or other stem segments (transdifferentiate + V.PTCP;PRS → *transdifernating). Stem-changing errors most commonly occur in lemmas that are long or otherwise improbable under the language model (*e.g.* containing rare characters: fœderate).

The remaining instances of the prediction not matching the reference can be attributed to the artifacts of the data itself (target errors). As Gorman et al. (2019) point out, the UniMorph data often includes only one of the multiple attested inflected forms: *e.g.* for greet + V.PTCP;PST the given gold inflection is grat, and our model is penalized for outputting greeted. In other cases, multiple inflections are included in the test set, *e.g.* scandal + V.PTCP;PRS is listed as both scandaling and scandalling; however, since these are two different test data points, any

deterministic model will inevitably lose a point for getting at least one of them wrong. Finally, some reference inflections are mismatched with their tagset: uptake + V;PST → *uptaking.

Augmenting the small training set with the hallucinated examples and copying improves the English Transformer's accuracy by around 5 points (84.32 → 89.02). This is mostly due to the model becoming more robust and less prone to "silly" errors; however, allomorphy errors still remain tricky. Analogy-guided inflection could help resolve such ambiguous cases, but only if the exemplar is guaranteed to apply the same inflection rule and not a different plausible one. For example, to fix the cruel → *crueling error, the exemplar must double the final consonant, but our orthographic nearest-neighbor retriever proposes either [soul → souling] (by Levenshtein distance) or [entrammel → entrammeling] (by common suffix), neither of which exhibits the desired property. Choosing perfect exemplars from the training set is difficult for two reasons: (1) orthographic (or phonological) similarity does not always translate to identical inflection, and (2) the sparse low-resource training set might not include a perfect analogy for every test instance.

Similarly to most languages in Table 6.2, the English nearest-neighbor augmentation also decreases the inflection accuracy. Besides the challenges of the exemplar selection, we also find that the analogy-guided inflection setup introduces its own difficulties. Concatenating the exemplar to the test lemma produces much longer inputs, increasing the likelihood of "silly" errors: for example, for lute + V;PST augmented with [self-pollute → self-polluted], the model gets distracted by the exemplar and generates *luteedslf-ped. The retrieved test-time exemplars might also be substantially longer or shorter than the training ones, which were synthesized to have the same length as the input lemma (§6.4.2); this mismatch could also pose a challenge.

While not all of the mentioned sources of error are equally likely to apply to truly low-resource languages, much of our analysis on English illustrates the main challenges of the task and provides insights into what causes the proposed nearest-neighbor inflection method to underperform.

6.6 Conclusion

In this chapter, we frame the morphological inflection of lemmas not seen in training as an exemplar-guided analogy task and propose exemplar retrieval methods designed to select the most orthographically, semantically, or phonologically relevant exemplars. We show that the phonological and orthographic nearest neighbor exemplars contribute more to the inflection accuracy than the semantic ones, but none of the proposed exemplar-augmented methods consistently outperform the simpler non-retrieval baseline based on augmenting the training set.

Our oracle experiment shows that, in principle, better exemplar selection models could further improve the morphological inflector performance on new lemmas. This highlights what future work on morphological generalization might focus on: defining similarity metrics based on the

factors more closely correlated with the word's inflection class, or proposing more sophisticated retriever components than the simple nearest-neighbor selector (*e.g.* training the retriever jointly with the exemplar-augmented inflector).

Chapter 7

Conclusion

In this thesis, we computationally model the different facets of non-standard written language, considering it on different linguistic scales (orthographic, morphological, lexical) and at different levels of social granularity (user, group, or lect). Using a range of machine learning techniques, including finite-state, latent-variable, and sequence-to-sequence approaches, we propose models of language variation in data sources from historical printed documents to social media. Our work aims to both uncover new linguistic knowledge using computational analyses and to use linguistic inductive biases to improve the handling of linguistic variation in natural language processing applications. The key contributions of this thesis are as follows:

1. Chapter 2 proposes an unsupervised probabilistic model for automatic compositor attribution, the task of clustering pages of a historical printed document according to the person who set the type. The model infers the compositors' identities by tracking their idiosyncratic orthographic choices, and it shows a high level of agreement with the manual scholarly attributions which rely on the same features, both in terms of the predicted number of compositors and their page assignments. Our tool could help bibliographers by serving as an initial step in performing compositor attribution on less studied texts.
2. Chapter 3 introduces an unsupervised finite-state approach for converting idiosyncratically romanized text in multiple languages into the conventional orthography of the corresponding language. Informative priors that encode similarity between character shapes or pronunciations bring our model's performance close to its supervised counterpart and make it competitive with the unsupervised neural architectures. We also present a new dataset of informally romanized Russian, explore ways to combine the finite-state and neural approaches, and empirically analyze their relative strengths and weaknesses.
3. Chapter 4 statistically tests two hypotheses about the factors driving the emergence of new word forms in the language. In a distributional semantics study on a diachronic corpus

of historical published literature, we show that neologisms are more likely to emerge in sparser regions of the semantic space and in the regions where the existing words' usage grows more rapidly. Although we find both of the factors predictive, the frequency growth of the semantic neighbors is found to be more significant in this study.

4. Chapter 5 extends the methodology introduced in the previous chapter to test the same two hypotheses on a new corpus collected from Twitter. We find that the neighborhood sparsity and the rate at which the in-neighborhood word use increases are predictive of neology even on a smaller time scale, which we were able to track with social media data. By reproducing the experimental findings on both the new dataset and the corpora from the previous chapter, we demonstrate the robustness of our conclusions.
5. Chapter 6 proposes a retrieval-augmented model for morphological inflection of novel lemmas. Assuming that similar lemmas would inflect similarly, we append one most similar instance from the training set to the input at test time. We explore several similarity metrics encoding the lemmas' orthographic, semantic, and phonological proximity and show that exemplars relevant in terms of phonology and orthography yield better performance than the ones selected by semantic similarity. We also show that a better retrieval model could improve generalization to novel lemmas, highlighting a gap that future work could fill.

Multi-perspective view of language variation Our work seeks insights into non-standard language and linguistic variation from both the natural language processing perspective (application-oriented) and the computational linguistics one (knowledge discovery-oriented). The goals of the two subfields may seem disparate: one focuses on building efficient and robust applications (prioritizing invariance to variation) and the other on learning about language processing in humans (prioritizing the variation itself). However, NLP applications should be built with users in mind, and they need to be able to generalize to the whole spectrum of possible human linguistic inputs and understand how the language variation contextualizes their meaning. Further work on non-standard language and other dimensions of linguistic variation could merge the two perspectives into a mutually beneficial cycle: NLP techniques can be used to gain linguistic insights, and these new insights can be used to improve NLP models.

Multi-scale modeling of language Studying non-standard language requires attending to variation that many NLP systems abstract away from—but nevertheless, some level of abstraction is necessary for drawing generalizable conclusions. This thesis considers linguistic phenomena at different levels of linguistic, temporal, and social granularity, and future work could focus on finding the optimal scale and level of abstraction: for example, finding the middle ground between meaning representations that are too coarse to capture the differences between word senses

(*e.g.* static embeddings which assign a single vector to each word type) and the ones that are too fine-grained (contextualized embeddings which treat each occurrence of a word in a different context differently). Another avenue of future research could consider linguistic phenomena on multiple levels of granularity at once: for example, our work on neology in social media (Chapter 5) could be extended in order to model how linguistic innovation spreads from individual speakers to whole populations, grounded in the evolutionary theories of language change.

Appendix

Appendix A

Supplemental Material for Chapter 3

A.1 Hyperparameter Settings

WFST hyperparameters The Witten–Bell smoothing parameter for the language model is set to 10, and the relative entropy pruning threshold is 10^{-5} for the trigram model and $2 \cdot 10^{-5}$ for higher-order models. Unsupervised training is performed in batches of size 10 and the language model order is increased every 100 batches. While training with the bigram model, we disallow insertions and freeze all the deletion probabilities at e^{-100}. The EM stepsize decay rate is $\beta = 0.9$. The emission arc pruning threshold is gradually decreased from 5 to 4.5 (in the negative log probability space). We perform multiple random restarts for each experiment, initializing the emission distribution to uniform plus random noise.

UNMT hyperparameters We use the PyTorch UNMT implementation of He et al. (2020)[1] which incorporates improvements introduced by Lample et al. (2019) such as the addition of a max-pooling layer. We use a single-layer LSTM with hidden state size 512 for both the encoder and the decoder. The embedding dimension is set to 128. For the denoising autoencoding loss, we adopt the default noise model and hyperparameters as described by Lample et al. (2018). The autoencoding loss is annealed over the first 3 epochs. We predict the output using greedy decoding and set patience for early stopping to 10.

In the experiments of §3.6.1, we set the maximum output length to be equal to the length of the input sequence. In §3.6.2, we instead tune the maximum training sequence length (controlling how much training data is used) and the maximum allowed output length by optimizing the validation set CER. In our case, the maximum output length is important because the evaluation metric penalizes the discrepancy in length between the prediction and the reference; we observe

[1] https://github.com/cindyxinyiwang/deep-latent-sequence-model

the best results when setting it to 40 characters for Arabic and 180 for Russian. At training time, we filter out sequences longer than 100 characters for either language, which constitute 1% of the available Arabic training data (both the Arabic-only LM training set and the Latin-only training set combined) but almost 70% of the Russian data. Surprisingly, the Russian model trained on the remaining 30% achieves better results than the one trained on the full data; we hypothesize that the improvement comes from having a more balanced training set, since the full data is heavily skewed towards the Cyrillic side (LM training set) otherwise (see Table 3.2).

A.2 Additional Preprocessing

This section describes the additional preprocessing steps added in the experiments reported in §3.6.2. As before, we lowercase and segment all sequences into characters as defined by Unicode codepoints, so diacritics and non-printing characters like ZWJ are also treated as separate vocabulary items. To filter out foreign or archaic characters and rare diacritics, we also restrict the alphabets to characters that cover 99% of the monolingual training data. After that, we add any standard alphabetical characters and numerals that have been filtered out back into the source and target alphabets. All remaining filtered characters are replaced with a special UNK symbol in all splits except for the target-side test.

Input	ana h3dyy 3lek bokra 3la 8 kda	
Ground truth	انا حأعدي عليك بكرة على 8 كده	AnA H>Edy Elyk bkrp ElY 8 kdh
WFST	انا حد بي لك بكر لأ 8 كده	AnA Hd yy lk bkr l> 8 kdh
Reranked WFST	انا حد بي لك بكر لأ 8 كده	AnA Hd yy lk bkr l> 8 kdh
Seq2Seq	انا بأدي أخلك حر أول 1 كده	AnA b>dy >xlk Hr >wl 1 kdh
Reranked Seq2Seq	انا بأدي أخلك حر أول 1 كده	AnA b>dy >xlk Hr >wl 1 kdh
Product of experts	انا دي لك ب كرا ألا 8 كده	AnA dy lk b krA > lA 8 kdh

Table A.1: Different model outputs for an Arabizi transliteration example (left column—Arabic, right—Buckwalter transliteration). Prediction errors are highlighted in red in the romanized versions. Correctly transliterated segments that do not match the ground truth because of spelling standardization during annotation are highlighted in yellow.

Input	kshullaka baalina avala horaatavannu adu vivarisuttade.	
Ground truth	ಕ್ಷುಲ್ಲಕ ಬಾಳಿನ ಅವಳ ಹೋರಾಟವನ್ನು ಅದು ವಿವರಿಸುತ್ತದೆ.	kṣullaka bāḷina avaḷa hōrāṭavannu adu vivarisuttade.
WFST	ಕುಹೂಲ್ಲಾಖೆ ಬಾಲಿನು ವಾಳ ಹೊರತಾವನ್ನು ಆದು ವಿವರಿಸುತ್ತದೆ.	kuhūllākhe bālinu vāḷa horatāvannu ādu vivarisuttade.
Reranked WFST	ಕುಹೂಲ್ಲಾಖೆ ಬಾಲಿನ ವಾಳು ಹೊರತಾವನ್ನು ಆದು ವಿವರಿಸುತ್ತದೆ.	kuhūllākhe bālina vāḷu horatāvannu ādu vivarisuttade.
Seq2Seq	ಕಳುಹುಳ್ಳ ಬಾವಿಂಗ್ ಇಲ್ಲವೇ ಹೋರಾಟವನ್ನು ಇದು ವಿವರಿಸುತ್ತದ	kaḷuhuḷḷa bāviṁg illavē hōrāṭavannu idu vivarisuttade.
Reranked Seq2Seq	ಕಳುಹುಳ್ಳ ಬಾವಿಂತ ಇಲ್ಲವೇ ಹೋರಾಟವನ್ನು ಇದು ವಿವರಿಸುತ್ತದೆ	kaḷuhuḷḷa bāviṁta illavē hōrāṭavannu idu vivarisuttade.
Product of experts	ಕಳ್ಳ ಬಾಕಲಿನ್ನ ವಾಲಾ ಹೋರಾಟತ್ವಾನ್ನು ದು ವಿವಾರಿಸುತ್ತದ	kaḷḷa bākalinna vālā hōrāṭatvānnu du vivārisuttada

Table A.2: Different model outputs for a Kannada transliteration example (left column—Kannada, right—ISO 15919 transliterations). The ISO romanization is generated using the Nisaba library (Johny et al., 2021). Prediction errors are highlighted in red in the romanized versions.

Bibliography

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada. Association for Computational Linguistics. 6.2

Jean Aitchison. 2001. *Language Change: Progress Or Decay?* Cambridge University Press. 4.1

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics. 5.3.2

Mohamed Al-Badrashiny, Ramy Eskander, Nizar Habash, and Owen Rambow. 2014. Automatic transliteration of Romanized dialectal Arabic. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 30–38, Ann Arbor, Michigan. Association for Computational Linguistics. 3.2

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, 90(2):119–161. 6.2

Cyril Allauzen, Mehryar Mohri, and Brian Roark. 2003. Generalized algorithms for constructing statistical language models. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 40–47, Sapporo, Japan. Association for Computational Linguistics. 3.3.1

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer. `http://www.openfst.org`. 3.5.2

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language*

Processing (EMNLP-IJCNLP), pages 984–996, Hong Kong, China. Association for Computational Linguistics. 6.4.1, 6.4.2

René Appel and Pieter Muysken. 2006. *Language contact and bilingualism*. Amsterdam University Press. 4.1

Robert Bamler and Stephan Mandt. 2017. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 380–389. PMLR. 4.2

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina J. Mielke, Charbel El-Khaissi Elena Budianskaya, Tiago Pimentel, William Lane Michael Gasser, Mohit Raj, Matt Coler, Jaime Rafael Montoya Samame, Delio Siticonatzi Camaiteri, Benoît Sagot, Esaú Zumaeta Rojas, Didier López Francis, Arturo Oncevay, Juan López Bautista, Gema Celeste Silva Villegas, Lucas Torroba Hennigen, Adam Ek, David Guriel, Peter Dirix, Jean-Philippe Bernardy, Andrey Scherbakov, Aziyana Bayyr-ool, Antonios Anastasopoulos, Roberto Zariquiey, Karina Sheifer, Sofya Ganieva, Hilaria Cruz, Ritván Karahó□a, Stella Markantonatou, George Pavlidis, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Candy Angulo, Jatayu Baxi, Andrew Krizhanovsky, Natalia Krizhanovskaya, Elizabeth Salesky, Clara Vania, Sardana Ivanova, Jennifer White, Rowan Hall Maudslay, Josef Valvoda, Ran Zmigrod, Paula Czarnowska, Irene Nikkarinen, Aelita Salchak, Brijesh Bhatt, Christopher Straughn, Zoey Liu, Jonathan North Washington, Yuval Pinter, Duygu Ataman, Marcin Wolinski, Totok Suhardijanto, Anna Yablonskaya, Niklas Stoehr, Hossep Dolatian, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Aryaman Arora, Richard J. Hatcher, Ritesh Kumar, Jeremiah Young, Daria Rodionova, Anastasia Yemelina, Taras Andrushko, Igor Marchenko, Polina Mashkovtseva, Alexandra Serova, Emily Prud'hommeaux, Maria Nepomniashchaya, Fausto Giunchiglia, Eleanor Chodroff, Mans Hulden, Miikka Silfverberg, Arya D. McCarthy, David Yarowsky, Ryan Cotterell, Reut Tsarfaty, and Ekaterina Vylomova. 2022. UniMorph 4.0: Universal morphology. In *Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC'22)*. 6.2

Steven Bedrick, Russell Beckley, Brian Roark, and Richard Sproat. 2012. Robust kaomoji detection in Twitter. In *Proceedings of the Second Workshop on Language in Social Media*, pages 56–64, Montréal, Canada. Association for Computational Linguistics. 3.7

Sarah Beemer, Zak Boston, April Bukoski, Daniel Chen, Princess Dickens, Andrew Gerlach, Torin Hopkins, Parth Anand Jawale, Chris Koski, Akanksha Malhotra, Piyush Mishra, Saliha Muradoglu, Lan Sang, Tyler Short, Sagarika Shreevastava, Elizabeth Spaulding, Testumichi Umada, Beilei Xiang, Changbing Yang, and Mans Hulden. 2020. Linguist vs. machine:

Rapid development of finite-state morphological grammars. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 162–170, Online. Association for Computational Linguistics. 6.2

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery. 3.3.3

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California. Association for Computational Linguistics. 2.4.2

Taylor Berg-Kirkpatrick, Greg Durrett, and Dan Klein. 2013. Unsupervised transcription of historical documents. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 207–217, Sofia, Bulgaria. Association for Computational Linguistics. 2.4.1

Toms Bergmanis, Katharina Kann, Hinrich Schütze, and Sharon Goldwater. 2017. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39, Vancouver. Association for Computational Linguistics. 6.2

Jean Berko. 1958. The child's learning of English morphology. *Word*, 14(2-3):150–177. 6.2

Ann Bies, Zhiyi Song, Mohamed Maamouri, Stephen Grimes, Haejoong Lee, Jonathan Wright, Stephanie Strassel, Nizar Habash, Ramy Eskander, and Owen Rambow. 2014. Transliteration of Arabizi into Arabic orthography: Developing a parallel annotated Arabizi-Arabic script SMS/chat corpus. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 93–103, Doha, Qatar. Association for Computational Linguistics. 3.4.1, 3.4.2

Peter W. M. Blayney. 1991. *The First Folio of Shakespeare: In Conjunction with the Exhibition at the Folger Shakespeare Library, Washington, DC, April 1, 1991-September 21, 1991*. Folger Shakespeare Library. 2.1, 2.2, 7

Peter W. M. Blayney, editor. 1996. *The First Folio of Shakespeare: The Norton Facsimile*. Norton. 1, 2.4.1

Franz Boas. 1911. *Introduction [to Handbook of American Indian Languages]*. 677. US Government Printing Office. 4.1

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Lin-*

guistics, 5:135–146. 6.3

Claire Bowern. 2019. Semantic change and semantic stability: Variation is key. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 48–55, Florence, Italy. Association for Computational Linguistics. 5.1

Michel Bréal. 1904. *Essai de sémantique:(science des significations)*. Hachette. 4.1, 4.2, 5.2

Ian R. Burrows. 2013. "The peryod of my blisse": Commas, ends and utterance in Solyman and Perseda. *Textual Cultures: Texts, Contexts, Interpretation*, 8(2):95–120. 1

Lyle Campbell. 2013. *Historical Linguistics: an Introduction*. MIT Press, Cambridge, MA. 4.1, 4.2

Marc Canby, Aidana Karipbayeva, Bryan Lunt, Sahand Mozaffari, Charlotte Yoder, and Julia Hockenmaier. 2020. University of Illinois submission to the SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 137–145, Online. Association for Computational Linguistics. 6.2

A. Chalamandaris, P. Tsiakoulis, S. Raptis, G. Giannopoulos, and G. Carayannis. 2004. Bypassing Greeklish! In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA). 3.2

Aimilios Chalamandaris, Athanassios Protopapas, Pirros Tsiakoulis, and Spyros Raptis. 2006. All Greek to me! An automatic Greeklish to Greek transliteration system. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA). 3.1

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised POS induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 575–584, Cambridge, MA. Association for Computational Linguistics. 2.4.1

Paul Cook. 2012. Using social media to find English lexical blends. In *Proceedings of the 15th EURALEX International Congress (EURALEX 2012)*, pages 846–854, Oslo, Norway. 4.2

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? Encoder-decoder neural networks as cognitive models of English past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877, Florence, Italy. Association for Computational Linguistics. 6.2

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfver-

berg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels. Association for Computational Linguistics. 6.2

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics. 6.2

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics. 6.2

Ewa Dąbrowska and Marcin Szczerbiński. 2006. Polish children's productivity with case marking: the role of regularity, type frequency, and phonological diversity. *Journal of child language*, 33(3):559–597. 6.2

Kareem Darwish. 2014. Arabizi detection and conversion to Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 217–224, Doha, Qatar. Association for Computational Linguistics. 3.1, 3.2, 3.4.2

Kareem Darwish, Walid Magdy, and Ahmed Mourad. 2012. Language processing for Arabic microblog retrieval. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 2427–2430, New York, NY, USA. Association for Computing Machinery. 3.4.2

Frank E. Daulton. 2012. Lexical borrowing. In *The Encyclopedia of Applied Linguistics*. American Cancer Society. 4.2

Mark Davies. 2002. *The Corpus of Historical American English (COHA): 400 million words, 1810-2009*. Brigham Young University. 4.4.1, 5.4.2

Mark Davies. 2008. *The corpus of contemporary American English*. BYE, Brigham Young University. 4.4.1, 5.4.2

Marco Del Tredici and Raquel Fernández. 2018. The road to success: Assessing the fate of linguistic innovations in online communities. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1591–1603, Santa Fe, New Mexico, USA. Association

for Computational Linguistics. 4.2

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*. 4.4.4

Matthew S. Dryer. 2013. Prefixing vs. suffixing in inflectional morphology. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig. 6.3

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-out: Temporal referencing for robust modeling of lexical semantic change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470, Florence, Italy. Association for Computational Linguistics. 4.2

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2017. Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1136–1145, Copenhagen, Denmark. Association for Computational Linguistics. 4.2

Jacob Eisenstein. 2017. Identifying regional dialects in on-line social media. *The Handbook of Dialectology*, pages 368–383. 4.2

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2012. Mapping the geographical diffusion of new words. In *NIPS Workshop on Social Network and Social Media Analysis*. 4.2

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2014. Diffusion of lexical change in social media. *PloS one*, 9(11):e113114. 4.2, 5.1

Jason Eisner. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 1–8, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics. 3.3.3

Micha Elsner and Sara Court. 2022. OSU at SigMorphon 2022: Analogical inflection with rule features. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 220–225, Seattle, Washington. Association for Computational Linguistics. 6.2, 6.2

Ramy Eskander, Mohamed Al-Badrashiny, Nizar Habash, and Owen Rambow. 2014. Foreign words and the automatic processing of Arabic social media text written in Roman script. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 1–12, Doha, Qatar. Association for Computational Linguistics. 3.2

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Work-*

shop on Evaluating Vector-Space Representations for NLP, pages 30–35, Berlin, Germany. Association for Computational Linguistics. 4.4.4, 4.7

David Francis, Ella Rabinovich, Farhan Samir, David Mortensen, and Suzanne Stevenson. 2021. Quantifying cognitive factors in lexical decline. *Transactions of the Association for Computational Linguistics*, 9:1529–1545. 4.7

Evgeniy Gabrilovich and Alex Gontmakher. 2002. The homograph attack. *Commun. ACM*, 45(2):128. 3.3.2

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644. 4.2

Philip Gaskell. 2007. *A New Introduction to Bibliography*. Oak Knoll Press. 1

Omer Goldman, David Guriel, and Reut Tsarfaty. 2022. (Un)solving morphological inflection: Lemma overlap artificially inflates models' performance. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 864–870, Dublin, Ireland. Association for Computational Linguistics. 6.1, 6.2, 6.4.1, 6.4.2, 4, 6.5, 6.5.1

Omer Goldman and Reut Tsarfaty. 2021. Minimal supervision for morphological inflection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2078–2088, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 6.3

Kyle Gorman. 2016. Pynini: A Python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, pages 75–80, Berlin, Germany. Association for Computational Linguistics. 3.5.2

Kyle Gorman, Arya D. McCarthy, Ryan Cotterell, Ekaterina Vylomova, Miikka Silfverberg, and Magdalena Markowska. 2019. Weird inflects but OK: Making sense of morphological generation errors. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 140–151, Hong Kong, China. Association for Computational Linguistics. 1, 2, 6.5.1

Kyle Gorman and Richard Sproat. 2021. Morphological analysis and generation. In *Finite-State Text Processing*, pages 77–92. Springer. 6.2

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 6.3

Nizar Habash, Mona Diab, and Owen Rambow. 2012. Conventional orthography for dialectal Arabic. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 711–718, Istanbul, Turkey. European Language Resources Association (ELRA). 3.4.1

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics. 4.2, 4.4, 5.1

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*. 3.2, 3.5.1, A.1

Lars Hellsten, Brian Roark, Prasoon Goyal, Cyril Allauzen, Françoise Beaufays, Tom Ouyang, Michael Riley, and David Rybach. 2017. Transliterated mobile keyboard input via weighted finite-state transducers. In *Proceedings of the 13th International Conference on Finite State Methods and Natural Language Processing (FSMNLP 2017)*, pages 10–19, Umeå, Sweden. Association for Computational Linguistics. 3.2

Charlton Hinman. 1963. *The printing and proof-reading of the First Folio of Shakespeare*, volume 1. Oxford: Clarendon Press. 2.1, 2.2, 2.4.1

G. E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800. 3.5.1

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*, pages 7273–7283, Online. Association for Computational Linguistics. 4.2

David I. Holmes. 1994. Authorship attribution. *Computers and the Humanities*, 28(2):87–106. 2.1

Jonathan Hope. 1994. *The authorship of Shakespeare's plays: A socio-linguistic study*. Cambridge University Press. 2.1

Trevor H. Howard-Hill. 1973. The compositors of Shakespeare's Folio Comedies. *Studies in Bibliography*, 26:61–106. 2.2, 2.4.1, 7

Trevor H. Howard-Hill. 1976. *Compositors B and E in the Shakespeare First Folio and Some Recent Studies*. Self-published. 2.4.1

Trevor H. Howard-Hill. 1980. New light on compositor E of the Shakespeare First Folio. *The Library*, 6(2):156–178. 2.4.1

Alexandra Jaffe and Shana Walton. 2000. The voices people read: Orthography and the representation of non-standard speech. *Journal of Sociolinguistics*, 4(4):561–587. 1

Kokil Jaidka, Niyati Chhaya, and Lyle Ungar. 2018. Diachronic degradation of language models: Insights from social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 195–200, Melbourne, Australia. Association for Computational Linguistics. 4.2

Matthew L. Jockers and Daniela M. Witten. 2010. A comparative study of machine learning methods for authorship attribution. *Literary and Linguistic Computing*, 25(2):215–223. 2.1

Cibu Johny, Lawrence Wolf-Sonkin, Alexander Gutkin, and Brian Roark. 2021. Finite-state script normalization and processing utilities: The Nisaba Brahmic library. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 14–23, Online. Association for Computational Linguistics. A.2

Patrick Juola. 2006. Authorship attribution. *Foundations and Trends in Information Retrieval*, 1(3):233–334. 2.1

Katharina Kann and Hinrich Schütze. 2016a. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70, Berlin, Germany. Association for Computational Linguistics. 6.2

Katharina Kann and Hinrich Schütze. 2016b. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. Association for Computational Linguistics. 6.2

Andres Karjus. 2021. *Competition, selection and communicative need in language change: an investigation using corpora, computational modelling and experimentation*. Ph.D. thesis, The University of Edinburgh. 4.1

Andres Karjus, Richard A Blythe, Simon Kirby, and Kenny Smith. 2020. Quantifying the dynamics of topical fluctuations in language. *Language Dynamics and Change*, 1:1–40. 4.2

Michael H Kelly. 1992. Using sound to solve syntactic problems: the role of phonology in grammatical category assignments. *Psychological review*, 99(2):349. 6.3

Charles Kemp, Yang Xu, and Terry Regier. 2018. Semantic typology and efficient communication. *Annual Review of Linguistics*, 4(1):109–128. 4.1

Daniel Kershaw, Matthew Rowe, and Patrick Stacey. 2016. Towards modelling language innovation acceptance in online social networks. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*,

pages 553–562. ACM. 4.2

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. In *International Conference on Learning Representations*. 6.2

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*. 6.2

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sabrina J. Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal Morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 6.2

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612. 3.2

Jordan Kodner, Salam Khalifa, Khuyagbaatar Batsuren, Hossep Dolatian, Ryan Cotterell, Faruk Akkus, Antonios Anastasopoulos, Taras Andrushko, Aryaman Arora, Nona Atanalov, Gábor Bella, Elena Budianskaya, Yustinus Ghanggo Ate, Omer Goldman, David Guriel, Simon Guriel, Silvia Guriel-Agiashvili, Witold Kieraś, Andrew Krizhanovsky, Natalia Krizhanovsky, Igor Marchenko, Magdalena Markowska, Polina Mashkovtseva, Maria Nepomniashchaya, Daria Rodionova, Karina Scheifer, Alexandra Sorova, Anastasia Yemelina, Jeremiah Young, and Ekaterina Vylomova. 2022. SIGMORPHON–UniMorph 2022 shared task 0: Generalization and typologically diverse morphological inflection. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 176–203, Seattle, Washington. Association for Computational Linguistics. 6.2

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics. 3.5.3

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, 60(1):9–26. 2.1

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant

detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 625–635. ACM. 4.2

Vivek Kulkarni, Yingtao Tian, Parth Dandiwala, and Steve Skiena. 2018. Simple neologism based domain independent models to predict year of authorship. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 202–212, Santa Fe, New Mexico, USA. Association for Computational Linguistics. 5.3.2

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics. 4.2

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*. 3.2, 3.5.1, A.1

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*. 3.2, A.1

Jackson L. Lee, Lucas F.E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. 2020. Massively multilingual pronunciation modeling with WikiPron. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4223–4228, Marseille, France. European Language Resources Association. 3.7

Alessandro Lenci. 2018. Distributional models of word meaning. *Annual review of Linguistics*, 4:151–171. 4.1, 3, 5.2

Percy Liang and Dan Klein. 2009. Online EM for unsupervised models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 611–619, Boulder, Colorado. Association for Computational Linguistics. 3.3.3, 3.3.3

Rochelle Lieber. 2017. Derivational morphology. In *Oxford Research Encyclopedia of Linguistics*. Oxford University Press. 4.2

Ling Liu and Mans Hulden. 2020. Analogy models for neural word inflection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2861–2878, Barcelona, Spain (Online). International Committee on Computational Linguistics. 6.1, 6.2

Ling Liu and Mans Hulden. 2022. Can a Transformer pass the wug test? Tuning copying bias in neural morphological inflection models. In *Proceedings of the 60th Annual Meeting of the*

Association for Computational Linguistics (Volume 2: Short Papers), pages 739–749, Dublin, Ireland. Association for Computational Linguistics. 6.2, 6.4.1

Olga N Lyashevskaya and Sergey A Sharov. 2009. *Frequency dictionary of modern Russian based on the Russian National Corpus [Chastotnyy slovar' sovremennogo russkogo jazyka (na materiale Nacional'nogo korpusa russkogo jazyka)]*. Azbukovnik, Moscow. 3.4.2

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11):2579–2605. 4.1

Peter Makarov and Simon Clematide. 2018. UZH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 69–75, Brussels. Association for Computational Linguistics. 6.2

Peter Makarov, Tatiana Ruzsics, and Simon Clematide. 2017. Align and copy: UZH at SIGMORPHON 2017 shared task for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 49–57, Vancouver. Association for Computational Linguistics. 6.2

Gary F Marcus, Ursula Brinkmann, Harald Clahsen, Richard Wiese, and Steven Pinker. 1995. German inflection: The exception that proves the rule. *Cognitive psychology*, 29(3):189–256. 6.2

Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020. UniMorph 3.0: Universal Morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association. 6.2

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics. 6.2

Gretchen McCulloch. 2020. *Because Internet: Understanding the new rules of language*. Penguin. 5.1

Kate McCurdy, Sharon Goldwater, and Adam Lopez. 2020. Inflecting when there's no majority:

Limitations of encoder-decoder neural networks as cognitive models for German plurals. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1745–1756, Online. Association for Computational Linguistics. 6.2

Donald Francis McKenzie. 1969. Printers of the mind: Some notes on bibliographical theories and printing-house practices. *Studies in Bibliography*, 22:1–75. 1

Donald Francis McKenzie. 1984. Stretching a point: Or, the case of the spaced-out comps. *Studies in Bibliography*, 37:106–121. 1

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119. 4.4.3, 5.3.3

Mehryar Mohri and Michael Riley. 2002. An efficient algorithm for the n-best-strings problem. In *Seventh International Conference on Spoken Language Processing*. 3.5.1

Padraic Monaghan, Nick Chater, and Morten H Christiansen. 2005. The differential role of phonological and distributional cues in grammatical categorisation. *Cognition*, 96(2):143–182. 6.3

Steven Moran and Daniel McCloy, editors. 2019. *PHOIBLE 2.0*. Max Planck Institute for the Science of Human History, Jena. 3.7

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 3.7, 6.3

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484, Osaka, Japan. The COLING 2016 Organizing Committee. 3.7

Radford M Neal. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265. 2.4.3

Dong Nguyen, Laura Rosseel, and Jack Grieve. 2021. On learning and representing social meaning in NLP: a sociolinguistic perspective. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 603–612, Online. Association for Computational Linguistics. 1

John O'Connor. 1975. Compositors D and F of the Shakespeare First Folio. *Studies in Bibliography*, 28:81–117. 2.4.1

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics. 3.5.3

Allison Parrish. 2021. Desire (under)lines: Notes toward a queer phenomenology of spell check. Online. 1

Martin Paulsen. 2014. Translit: Computer-mediated digraphia on the Runet. *Digital Russia: The Language, Culture and Politics of New Media Communication*. 3.1

Tiago Pimentel, Maria Ryskina, Sabrina J. Mielke, Shijie Wu, Eleanor Chodroff, Brian Leonard, Garrett Nicolai, Yustinus Ghanggo Ate, Salam Khalifa, Nizar Habash, Charbel El-Khaissi, Omer Goldman, Michael Gasser, William Lane, Matt Coler, Arturo Oncevay, Jaime Rafael Montoya Samame, Gema Celeste Silva Villegas, Adam Ek, Jean-Philippe Bernardy, Andrey Shcherbakov, Aziyana Bayyr-ool, Karina Sheifer, Sofya Ganieva, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Andrew Krizhanovsky, Natalia Krizhanovsky, Clara Vania, Sardana Ivanova, Aelita Salchak, Christopher Straughn, Zoey Liu, Jonathan North Washington, Duygu Ataman, Witold Kieraś, Marcin Woliński, Totok Suhardijanto, Niklas Stoehr, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Richard J. Hatcher, Emily Prud'hommeaux, Ritesh Kumar, Mans Hulden, Botond Barta, Dorina Lakatos, Gábor Szolnok, Judit Ács, Mohit Raj, David Yarowsky, Ryan Cotterell, Ben Ambridge, and Ekaterina Vylomova. 2021. SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–259, Online. Association for Computational Linguistics. 1, 6.2, 6.2, 6.4.2

Steven Pinker and Alan Prince. 1988. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28(1-2):73–193. 2

Yuval Pinter, Cassandra L. Jacobs, and Max Bittker. 2020. NYTWIT: A dataset of novel words in the New York Times. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6509–6515, Barcelona, Spain (Online). International Committee on Computational Linguistics. 1

Nima Pourdamghani and Kevin Knight. 2017. Deciphering related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2513–2518, Copenhagen, Denmark. Association for Computational Linguistics. 3.2

Michael Proffitt, editor. 2018. *OED Online*. Oxford University Press. `http://www.oed.com/`.

4.6

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531. 4.4.4

Sujith Ravi and Kevin Knight. 2009. Learning phoneme mappings for transliteration without parallel data. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 37–45, Boulder, Colorado. Association for Computational Linguistics. 3.2

Samuel Reese, Gemma Boleda, Montse Cuadros, Lluís Padró, and German Rigau. 2010. Wikicorpus: A word-sense disambiguated multilingual Wikipedia corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA). 4.4.2

Vincent Renner, François Maniez, and Pierre Arnaud, editors. 2012. *Cross-disciplinary perspectives on lexical blending*. De Gruyter Mouton, Berlin. 4.2

Pervez Rizvi. 2016. Use of spellings for compositor attribution in the First Folio. *The Papers of the Bibliographical Society of America*, 110:1–53. 1

Brian Roark, Richard Sproat, Cyril Allauzen, Michael Riley, Jeffrey Sorensen, and Terry Tai. 2012. The OpenGrm open-source finite-state grammar software libraries. In *Proceedings of the ACL 2012 System Demonstrations*, pages 61–66, Jeju Island, Korea. Association for Computational Linguistics. 3.5.2

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing South Asian languages written in the Latin script: the Dakshina dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2413–2423, Marseille, France. European Language Resources Association. 3.4.3

David Rumelhart and Jay McClelland. 1986. On learning the past tenses of English verbs. In J. McClelland, D. Rumelhart, and the PDP Research Group, editors, *Parallel distributed processing: explorations into the microstructure of cognition. Vol. 2: Psychological and biological models*, pages 216–271. Bradford Books, Cambridge. 2

Maria Ryskina, Hannah Alpert-Abrams, Dan Garrette, and Taylor Berg-Kirkpatrick. 2017. Automatic compositor attribution in the First Folio of Shakespeare. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 411–416, Vancouver, Canada. Association for Computational Linguistics. 1

Maria Ryskina, Matthew R. Gormley, and Taylor Berg-Kirkpatrick. 2020a. Phonetic and visual priors for decipherment of informal romanization. In *Proceedings of the 58th Annual Meeting*

of the Association for Computational Linguistics, pages 8308–8319, Online. Association for Computational Linguistics. 1

Maria Ryskina, Eduard Hovy, Taylor Berg-Kirkpatrick, and Matthew R. Gormley. 2021. Comparative error analysis in neural and finite-state models for unsupervised character-level transduction. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 198–211, Online. Association for Computational Linguistics. 1

Maria Ryskina, Ella Rabinovich, Taylor Berg-Kirkpatrick, David Mortensen, and Yulia Tsvetkov. 2020b. Where new words are born: Distributional semantic analysis of neologisms and their semantic neighborhoods. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 367–376, New York, New York. Association for Computational Linguistics. 1, 7, 5.1, 5.2, 5.2, 5.3, 5.3.2, 5.6

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal. Association for Computational Linguistics. 4.4.4

Tatiana Shavrina and Olga Shapovalova. 2017. To the methodology of corpus construction for machine learning: Taiga syntax tree corpus and parser. In *Proc. CORPORA 2017 International Conference*, pages 78–84, St. Petersburg. 3.4.2

Zhiyi Song, Stephanie Strassel, Haejoong Lee, Kevin Walker, Jonathan Wright, Jennifer Garland, Dana Fore, Brian Gainor, Preston Cabe, Thomas Thomas, Brendan Callahan, and Ann Sawyer. 2014. Collecting natural SMS and chat conversations in multiple languages: The BOLT phase 2 corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1699–1704, Reykjavik, Iceland. European Language Resources Association (ELRA). 3.4.1

Ian Stewart and Jacob Eisenstein. 2018. Making "fetch" happen: The influence of social and linguistic context on nonstandard word growth and decline. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4370, Brussels, Belgium. Association for Computational Linguistics. 4.2, 4.4.4

Andreas Stolcke. 1998. Entropy-based pruning of backoff language models. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pages 270—274. 3.5.2

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2021. Survey of computational approaches to lexical semantic change detection. In Nina Tahmasebi, Lars Borin, Adam Jatowt, Yang Xu, and Simon Hengchen, editors, *Computational approaches to semantic change*. Language Science Press. 4.2

Gary Taylor. 1981. The shrinking compositor A of the Shakespeare First Folio. *Studies in Bibliography*, 34:96–117. 2.1, 2.2, 2.4.1, 7, 2.3, 2.5.1

John R Taylor and Anthony P. Grant. 2014. *Lexical Borrowing*. Oxford University Press, Oxford. 4.2

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics. 6.2, 6.2, 6.4.1, 6.4.2

Uriel Weinreich. 2010. *Languages in contact: Findings and problems*. Walter de Gruyter, The Hague. 4.6

Paul Werstine. 1982. Cases and compositors in the Shakespeare First Folio Comedies. *Studies in Bibliography*, 35:206–234. 2.4.1

Adina Williams, Tiago Pimentel, Hagen Blix, Arya D. McCarthy, Eleanor Chodroff, and Ryan Cotterell. 2020. Predicting declension class from form and meaning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6682–6695, Online. Association for Computational Linguistics. 6.3

Ian H Witten and Timothy C Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE transactions on information theory*, 37(4):1085–1094. 3.5.2

Lawrence Wolf-Sonkin, Vlad Schogol, Brian Roark, and Michael Riley. 2019. Latin script keyboards for South Asian languages with finite-state normalization. In *Proceedings of the 14th International Conference on Finite-State Methods and Natural Language Processing*, pages 108–117, Dresden, Germany. Association for Computational Linguistics. 3.2

Shijie Wu and Ryan Cotterell. 2019. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537, Florence, Italy. Association for Computational Linguistics. 3.7, 6.2

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Associa-*

tion for Computational Linguistics: Main Volume, pages 1901–1907, Online. Association for Computational Linguistics. 6.2, 6.4.2, 3

Shijie Wu, Pamela Shapiro, and Ryan Cotterell. 2018. Hard non-monotonic attention for character-level transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4425–4438, Brussels, Belgium. Association for Computational Linguistics. 3.7, 6.2

Quirin Würschinger. 2021. Social networks of lexical innovation. Investigating the social dynamics of diffusion of neologisms on Twitter. *Frontiers in Artificial Intelligence*, page 106. 5.3.2

Yang Xu and Charles Kemp. 2015. A computational evaluation of two laws of semantic change. In *Proceedings of the 37th Annual Conference of the Cognitive Science Society*. 4.1, 4.2

Changbing Yang, Ruixin (Ray) Yang, Garrett Nicolai, and Miikka Silfverberg. 2022. Generalizing morphological inflection systems to unseen lemmas. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 226–235, Seattle, Washington. Association for Computational Linguistics. 6.2, 6.4.1, 6.5