Multilingual Natural Language Processing via Generative Probabilistic Modeling

Wouldn't it be awesome if human language technology worked on all of the world's languages? Despite increased attention to a broader selection of languages in recent years, current practice generally focuses on processing speech and text written in English (and a handful of other widely spoken languages). This focus has biased NLP methods to perform better on languages with a certain typological profile. As a computer scientist interested in bringing natural language processing (NLP) to a diverse set of speakers, I view it as my job to remedy this situation. Specifically, given text in an *understudied* language or dialect, I want computer programs to be able to induce the language's underlying structure and use that structure to apply analytics and provide services to its speakers.

In terms of methodology, I work within machine learning. I develop generative probabilistic models—often with neural-network subcomponents—whose parameters can be estimated with little to no annotated training data. This often involves the development of novel approximate inference algorithms to enable efficient learning. In terms of my application area, NLP, much of my work has concentrated on modeling morphology—the manner in which individual words decompose into and are formed from meaning-bearing, subword units. When treating understudied languages with NLP models, it is requisite to move beyond words as the atomic units and to share parameters among related words.

As a data-driven field, machine learning, as applied to natural language processing, requires text or speech in the target language to learn. Unfortunately, the distribution of annotated data in the field disproportionately favors English and a handful of other widely spoken languages. To aid machine-learning-based methods for a diversity of languages, I am passionate about the creation of linguistic resources for lower-resource languages. To this end, I manage the UniMorph project (`https://unimorph.github.io/`), which has released open-source morphological lexicons in over 100 languages. Releasing data, however, is not enough. To promote community involvement with these languages, I have spearheaded the running of three shared tasks on various aspects of generating morphology.

The road map for the remainder of this document is as follows. First, I outline my excitement for generative modeling of language—specifically morphology—and, then, overview some of my published work in a series of case studies, highlighting relevant papers under preparation. Then, I discuss the details of the UniMorph project for the release of open-source morphologically annotated data and its potential to impact low-resource NLP in general. Afterwards, I describe my larger goal of promoting interdisciplinary research between NLP and linguistics. To conclude, I discuss my research agenda going forward.

Why Morphology? "Because sometimes we need characters and morphemes, too..."

Over the past few years I have pushed for modeling techniques that focus on the *subword* level—both morphemes and characters—rather than just the word level. Improvements in this area are necessary since, unfortunately, NLP has traditionally focused on languages with relatively impoverished morphological processes. For example, there is no overt marking on a noun in English or Chinese to indicate whether it is the object or the subject of a clause; rather, English and Chinese rely on word order to determine the proper syntactic relations. Compare *the bear is eating the fish* with *the fish is eating the bear*—the position of *the bear* determines whether it is the subject or the object. However, most languages in the world make a distinction in the word form itself (Comrie et al., 2013), making the pure reliance on word order insufficient. In Czech, for instance, one may either translate *the bear is eating the fish* as *medvěd jí rybu* or *rybu jí medvěd*; a Czech speaker simply expresses the concept of *fish* as *rybu* (rather than *ryba*) to indicate that it is the object of the sentence. When there is a plethora of annotated data and a paucity of inflectional morphology, it is possible to ignore the presence of inflection, i.e., treat the English words *fish* and *fishes* as unrelated. This scheme breaks down when we are working with a language like Czech, which has 12 individual forms for *fish*, due to the ensuing data sparsity.

Why Generative Models? "Because we really should explain *all* of the data..."

Human children learn language at a remarkable pace and without direct supervision—they are innately endowed with the ability to rapidly discover the "hidden" linguistic structure of the languages around them and generalize to produce novel utterances describing the events and concepts they encounter in the world. Why can't computers? In many respects, computers have shocking advantages: they have more computational power and scientists often furnish them with much more data—often annotated! Nevertheless, the children still come out on top in mastery. One reason for this is that NLP often approaches language at a very different tack: to create an NLP system for a given task, the reigning paradigm is to design a custom model or architecture, annotate data and train the system on those data. For most core NLP tasks, e.g., part-of-speech tagging and dependency parsing, conditional models yield the best performance (Goldberg, 2017). Human children, on the other hand, do not have annotated labeled data and, instead, try to find a model that explains the utterances they hear and produce utterances that make them understood (Chomsky, 1965). In machine-learning parlance, this corresponds to estimating a generative model. Generative models have many advantages: First, they may require fewer annotated training examples (Ng and Jordan, 2001; Yogatama et al., 2017) to effectively estimate the model's parameters. Second, generative models can benefit from unannotated data (allowing semi-supervised training).

Most important, the modeler may specify the stochastic process by which they believe the data were created. This direct specification of the generative process admits for the incorporation of scientific knowledge, e.g., linguistic facts and intuitions, that further help the model generalize from limited training data. Also, it allows the testing of scientific hypothesis about language in a unified framework.

Three Chapters

I now present three case studies of successful applications of (deep) generative modeling to various parts of human language; each corresponds to a published paper. In all three cases, I show that with an appropriate generative model, one can recover string forms and vector meaning representations for unseen words, and even generate facets of an entirely new language.

Chapter 1: Generating Unattested Word Forms. Native English speakers regularly construct utterances with novel words. Consider the following sentence: *The young politician was prone to overgenuflection—subordinating himself to the senior congressmen, often to his disadvantage*. Speakers may never have used the noun *overgenuflection*, but they are nevertheless confident how to generate its denominalized verb: *overgenuflect*. Indeed, it is clear that native competence in a language involves the ability to produce and analyze novel forms of all sorts. Luckily, languages are systematic—a speaker will have seen transformations of the type *overgenuflect*$\mapsto$*overgenuflection* before, e.g., *reflect*$\mapsto$*reflection* and *detect*$\mapsto$*detection*. In Cotterell et al. (2015), we formalize this notion as a generative graphical model over string-valued random variables. Using probabilistic finite-state machines, we directly model the string-to-string mapping that transforms a sequence of (underlying) morphemes to the actual word form. Thus, through the construction of a graph over the entire lexicon of a language, we allow the inference of the forms of unattested words. This line of research has turned out to be quite fruitful, leading to the invention of several new algorithms (Cotterell et al., 2014; Cotterell and Eisner, 2015; Peng et al., 2015) and modeling extensions (Shapiro et al., 2018). Following up, in Cotterell et al. (2017), we present a novel method of combining LSTM-based sequence-to-sequence models (Bahdanau et al., 2015) together into a graphical model over strings, improving performance over the finite-state techniques; this work received outstanding paper at EACL 2017.

Chapter 2: Generating Unattested Word Embeddings. NLP has undergone a paradigm shift over the past few years. Where past practitioners typically endowed their systems with hand-designed features of words, it is now more common to *learn* those features from the data themselves, representing words as high-dimensional real vectors. This approach is problematic, however, in that there is no parameter sharing among related words, i.e., the words *eat*

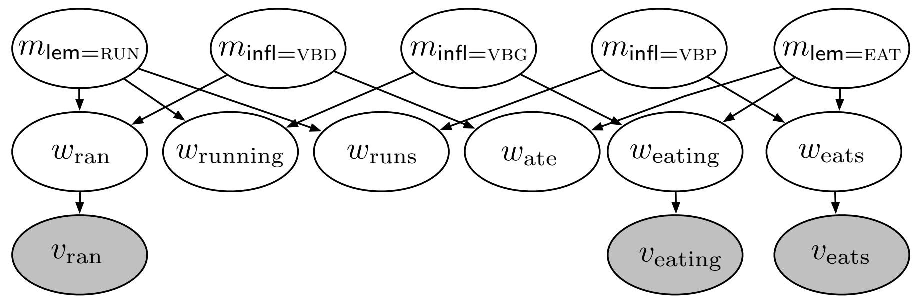

Figure 1: Gaussian graphical model for the generation of unattested word embeddings using morphologically related words.

and *eating* have completely divorced representations. Moreover, in most architectures, there is no principled manner to guess the meaning for unseen words. In Cotterell et al. (2016), we fix both of these problems with a generative model of word embeddings. Given a set of pretrained word embeddings and a morphological analyzer for the language, we construct a Gaussian graphical model that explains the attested word embeddings through a transformation of latent morpheme embeddings. We show a picture of the model in 1. Much like the word generation scenario above, we may now generate embeddings for novel words, as well as smoothing embeddings for rare words, based on their more frequent morphological relatives. We derive a coordinate descent procedure for fast inference and show improvements under several standard word embedding evaluation metrics.

Chapter 3: Generating New Languages. The previous two case studies have focused on generative modeling of individual pieces of human language: the first generated the forms of the words and the second generated the meaning of the words. However, we endeavor to go one step further and generate entire languages—we term this enterprise probabilistic typology. Concretely, we hope to construct a universal prior distribution, from which entire linguistic systems are drawn, whose parameters we will estimate using entire languages as training data. Generating an entirely new language is difficult, so we started with the construction of a distribution over sets of phonemes. Our first publication on the topic was awarded Best Long Paper at ACL 2017 (Cotterell and Eisner, 2017) and several extensions are in preparation (Cotterell and Eisner, 2018b,a). In the next years, we plan to extend our efforts beyond phonetics and phonology, to the generation of morphological and syntactic systems as well as complete lexicons. In addition to the engineering motivation of low-resource NLP, we also believe such models have a scientific motivation. We contend that a good generative model of language should get at the Chomskyan notion of universal grammar and help reveal the principles that undergird human language itself. For instance, a model that assigns high probability to held-out languages must have internalized certain notions of linguistic fitness. Such generative models should eventually yield insights into the science of linguistic typology, as they enable the quantitative testing of competing hypotheses.

Interdisciplinary Motivation

Motivation for my research comes from two disciplines in parallel: I am equally fascinated by the scientific questions behind human language and the engineering applications involving language that our society needs. I describe these two directions of computational research on language, and discuss bridging the gap between them.

The Scientific Question: Computational Linguistics

The core questions in linguistics involve the nature of linguistic knowledge, and how that knowledge is acquired by children. Computational linguistics seeks to address these questions using computational means. In its modern instantiation, this usually refers to the development of cognitively motivated probabilistic modeling of language and quantitative testing of linguistic hypotheses. In this regard, the relation between computational linguistics and traditional linguistics is no different than any other computational analog of a scientific discipline, e.g., computational biology and traditional biology. Why does computational linguistics matter now? For the first time in human history we have access to petabytes of user-generated language. Unlike traditional experiments in psycholinguistics, which rely on analyses data from a handful of subjects in a lab, we have access to language as it is used by speakers in the wild.

The Engineering Question: Natural Language Processing

Natural language processing, on the other hand, revolves around solving engineering problems. Here, rather than trying to explain how humans process languages, the goal is create useful artifacts that can be employed to improve the quality of life. For example, you don't judge Google Translate on whether it explains how human translators do their job. You judge it on whether it produces reasonably accurate and fluent translations for people who need to translate certain things in practice. The machine translation community has ways of measuring this, and they focus strongly on improving those scores. NLP is mainly used to help people navigate and digest large quantities of information that already exist in text form. It is also used to produce interfaces that allow better communication.

Bridging the Gap

Computational linguistics and natural language processing, despite both focusing on the computational study of language, are not always in lockstep. Indeed, many of the large conferences in the area, e.g., ACL and EMNLP, mostly feature work that is engineering-oriented in nature. A larger goal of my research trajectory is to remedy this divide. Computers are the telescopes of our era and have allowed scientists to more easily investigate a number of formidable questions in biology and astrophysics. Why should language be any different? As NLP develops more and more technically sophisticated methods for processing language, it is important for scientists to analyze the extent, to which those methods can help shed light on the question of how *humans* process language. The converse is also true—NLP built without linguistic insight may be brittle. For instance, if practitioners focus on optimizing models for only a handful of languages, e.g., English, without regard to the panoply of linguistic structures that exist, we run the risk of not being able to effectively process large swathes of the linguistic landscape (Bender, 2009).

I strive for my research to help bridge the gap between these two subfields. On one hand, I have a deep interest in bringing the quantitative methods standard in NLP to linguistics. For instance, the latent-variable model for morphophonology in Cotterell et al. (2015) may be seen as a trainable, probabilistic instantiation of the classic work *The Sound Pattern of English* (Chomsky and Halle, 1968). Likewise, the typological work in Cotterell and Eisner (2017) is a reinterpretation of the simulation work of Liljencrants and Lindblom (1972) using machinery from modern artificial intelligence. On the other, my focus on creating morphologically inspired models and typologically diverse datasets for the community already brings linguistic insight into the NLP work. I hope my work will help bring these two traditions of modeling language closer together.

Time Line

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Emily M. Bender. 2009. Linguistically naïve != language independent: why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 26–32. Association for Computational Linguistics.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*, volume 11. MIT Press.

Noam Chomsky and Morris Halle. 1968. *Sound Pattern of English*. Harper and Row, New-York.

Bernard Comrie, Matthew S. Dryer, David Gil, and Martin Haspelmath. 2013. Introduction. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Ryan Cotterell and Jason Eisner. 2015. Penalized expectation propagation for graphical models over strings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 932–942, Denver, Colorado. Association for Computational Linguistics.

Ryan Cotterell and Jason Eisner. 2017. Probabilistic typology: Deep generative models of vowel inventories. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada. Association for Computational Linguistics. Best Paper Award.

Ryan Cotterell and Jason Eisner. 2018a. A deep generative of vowel formant inventories. In *Proceedings of the 2017 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics. In Preparation.

Ryan Cotterell and Jason Eisner. 2018b. Three generative models of phoneme inventories. In *Proceedings of the 2017 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics. In Preparation.

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

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics (TACL)*, 3:433–447.

Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1651–1660, Berlin, Germany. Association for Computational Linguistics.

Ryan Cotterell, John Sylak-Glassman, and Christo Kirov. 2017. Neural graphical models over strings for principal parts morphological paradigm completion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 759–765, Valencia, Spain. Association for Computational Linguistics. Outstanding Paper Award.

Yoav Goldberg. 2017. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.

Johan Liljencrants and Björn Lindblom. 1972. Numerical simulation of vowel quality systems: The role of perceptual contrast. *Language*, pages 839–862.

Andrew Y. Ng and Michael I. Jordan. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 841–848.

Nanyun Peng, Ryan Cotterell, and Jason Eisner. 2015. Dual decomposition inference for graphical models over strings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–927, Lisbon, Portugal. Association for Computational Linguistics.

Pamela Shapiro, Ryan Cotterell, and Jason Eisner. 2018. A generative model of non-concatenative morphology. *Transactions of the Association for Computational Linguistics (TACL)*. In Preparation.

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898*.