0.1 General goals

In my graduate career, I would like to explore the possibilities of hybridization of stochastic and rule-based approaches in NLP. The particular application that I am targeting is building working grammar fragments of multiple languages automatically, possibly from only loosely annotated data. Having a language expert involved at later stages of grammar building is expensive, and richly and consistently annotated data is not always available. Furthermore, data is often poorly structured, so it is not trivial to infer typological and morphological information even from rich sources such as IGT (Interlinear Glossed Text). On the one hand, effective stochastic approaches will allow to compensate for imperfections of the data; on the other hand, giving field linguists feedback in form of working grammar fragments will encourage collaboration between field and computational linguists and will promote data standards. More **broadly**, a working grammar fragment that is easily developed with minimal specific formal knowledge can serve as an attractive interface between linguistic formalism and linguistic data, while automatic grammar creation from poorly structured data will lead to rapid software localization into low-resource languages and open market opportunities to endangered languages' communities.

0.2 First year plan and non-linear approaches to word order classification

Currently, customizing a grammar via a system such as LinGO Grammar Matrix (Bender et al., 2002) typically results in a very small working fragment, and enhancing it requires detailed knowledge of HPSG (Pollard & Sag, 1994) and LKB (Copestake, 2002). Bender et al., 2010, along with Wax, 2014, Song, 2014, Goodman, 2013 describe various enhancements to the system with the goal of outputting more grammar information in machine-readable form, given only human-readable input, independent of any particular formalism. As mentioned in my personal statement, as an undergrad, I worked on a word order classifier on this project (given feature vectors obtained automatically from IGTs, output the language's word order label). Applying k-means clustering led to a 10% accuracy increase on one of the training sets; however, on more training data the performance was not impressive. My hypothesis is that the feature vectors obtained from the IGTs represent data that is not linearly separable, yet typological information can still be learned.

In linguistic terms, simply the counts of digrams representing SV vs. VS, OV vs. VO sequences, etc., do not do a good job accounting for variations within language and thus a model trained on vectors obtained linearly from such counts often makes wrong predictions. For example, a language can be basically SVO but allow other orders in certain contexts. Furthermore, while data may contain roughly the same amount of VO and OV examples, the language that the data represents may still prefer one to the other, while a "naive" approach will declare such word order "free". My more narrow hypothesis is that a step function which allows for thresholds of values such as VO and OV digram counts will more accurately map the counts to a basic word order.

To test this idea, I will train models using different step sizes for thresholds, i.e., consider Euclidean space for word orders as in Bender et al., 2013, but, consider also a threshold λ which represents how much variation we are ready to discard. If the number of SV digrams in the data is smaller than λ, we will not count them. If at the same time the number of VS digrams is roughly the same but still greater than $lambda$, we will count them, thus predicting that VS is more likely order for the dataset. Using labeled training data, it is possible to test multiple values for λ and determine the best one.

0.3 Subsequent years

The example above taken in isolation is a fairly simple machine learning problem; however, my objective is to learn the ways of customizing such approaches to better suite linguistic data. Consider the step function described above to be some non-linear function ϕ, also a variable, with appropriate parameters as $\vec{\lambda}$. My hypothesis is that, for language data such as IGT, some ϕ will yield better results. My plan is to first familiarize myself with relevant work regarding using SVM, kernels, decision trees in NLP. Then I will try some of these algorithms on the same feature vectors that I used for k-means clustering, with the goal of then generalizing the findings and applying them to other typological features of languages. As feature selection and parameter tuning are fundamental for machine learning, studying and documenting what approaches work best in grammar engineering could contribute to a breakthrough in language technology.

0.4 Resources

The AGGREGATION project already has enough resources for me to complete my first year objectives. Generally, I plan to use the ODIN data (Lewis & Xia, 2010), a large corpus of IGTs, noisy enough to represent the data quality issues which I set out to resolve. The CLMS grammar engineering course also provides relevant data that I typically have permission to use. In the future, as collaboration with field linguists increases, new IGT data will always be available.

A valuable resource is also the University of Washingon's CSE department which boasts excellent Machine Learning and NLP teams and with which UW Linguistics maintains a strong working relationship. A UW CSE alumna, I look forward to collaborating with them in the future.

0.5 Impact

Finding effective combinations of rule-based and stochastic approaches in grammar engineering has practical applications in language software which uses semantic information and is therefore more precise and reliable (e.g. machine translation). On the one hand, bringing language technology to endangered languages will give them modern currency which can help preserve the language and the culture. On the other hand, being able to learn grammars of low-resource languages from data that is neither well-structured nor well-annotated means better software solutions in healthcare and education will be available to every community.

References

Bender, E. M., Drellishak, S., Fokkens, A., Poulson, L., & Saleem, S. (2010). Grammar customization. *Research on Language & Computation*, 1-50. (10.1007/s11168-010-9070-1)

Bender, E. M., Flickinger, D., & Oepen, S. (2002). The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In J. Carroll, N. Oostdijk, & R. Sutcliffe (Eds.), *Proceedings of the workshop on grammar engineering and evaluation at the 19th international conference on computational linguistics* (pp. 8–14). Taipei, Taiwan.

Copestake, A. (2002). *Implementing typed feature structure grammars.* Stanford, CA: CSLI Publications.

Goodman, M. W. (2013). Generation of machine-readable morphological rules with human readable input. *UW Working Papers in Linguistics*, *30*.

Lewis, W., & Xia, F. (2010). Developing odin: A multilingual repository of annotated language data for hundreds of the world's languages. *Literary and Linguistic Computing*, *25*.

Pollard, C., & Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar.* Chicago, IL and Stanford, CA: The University of Chicago Press and CSLI Publications.

Song, S. (2014). *A grammar library for information structure.* Unpublished doctoral dissertation, University of Washington.

Wax, D. (2014). *Automated grammar engineering for verbal morphology.* University of Washington.