Dependency parsing is an integral part of Natural Language Processing (NLP) research for many languages. Research in dependency parsing has mainly dealt with improving accuracy for a limited number of languages. Current dependency parsing algorithms have developed mainly for languages with an ample amount of training data. Most of this data has been collected for shared tasks at conferences and are available mainly for European and resource-rich languages. New researchers into the area may not know which algorithm and techniques work best with a new, untested, language.

To address this issue, we will look at ensemble approaches to dependency parsing. More specifically, we look at three methods. First, stacking parsers' outputs into a weighted graph and extracting a tree structure using simple voting. Second, analyzing each parsers' errors distribution and using that as an input into the weighted graph through fuzzy clustering methods. Third, using a meta-classifier to choose the best parser for each and every word in our input. The parsers in each situation may come from a variety of techniques such as graph-based, transition-based, and constituent conversion. Using a variety of parsers allows us to study the errors associated with the parsers and choose the best combination or individual parser for each situation.

Even though many tools exist for these European and resource-rich languages, dependency parsing techniques are most commonly only tested using accuracy scores, both unlabeled and labeled. If a new technique is developed for a high accuracy such as English or Japanese, the results are often equivalent to existing techniques or sometimes worse. Due to this, research is often only concerned with a very specific linguistic construction, domain, or localized feature. This often leads to a scenario, where one size does not fit all, particularly for under-resourced languages.

To make sure our techniques are useful for most languages, we analyzed them on large and small language data sets from a variety of language families. We want to give special attention to under-resourced languages, so we additionally show techniques on semi-supervised training via self-training. For under-resourced languages, self-training can be an important tool both for parser accuracy and for creating new annotated data. When using ensemble parsers, a fundamental self-training question arises on whether the individual parsers should be retrained on their own data or on ensemble data. Whether under-resourced or resource-rich, we feel that limiting the analysis to accuracy scores does not fully determine whether a technique is useful or not. To test our techniques down a typical NLP pipeline, we turn to machine translation.

Machine translation is often the first task people want solved for their language but often the last step in the process. Many components go into a successful system. These systems come in a variety of forms, whether rule-based or statistically based. One concern for machine translation is whether the early components of the pipeline are accurate. A 2% error in part-of-speech tagging may lead to a much higher percentage of parsing errors which in turn ends up in a double figure error rate in the final translation. Reducing the errors in early pipeline components is a prime concern so that researchers in machine translation can focus on the actual translation and not generalize earlier errors.

To examine the effects of dependency parsing down the NLP pipeline. Our dependency models will be evaluated using the Treex system and TectoMT translation system. This system, as opposed to other popular machine translation systems, makes direct use of the dependency structure during the conversion from source to target languages via a tectogrammatical tree translation approach. We will compare UAS accuracy to corresponding NIST and BLEU scores from the start to finish of the machine translation pipeline.

Unfortunately any current approach to test dependency parsing's effect on machine translation is going to run into one major road block. There is no gold data for English dependency trees that has a corresponding gold standard translation. For the vast majority of English dependency parsers, the status quo is to train with data automatically converted from constituent trees. This leads to a final parse with at least an 8% error rate in UAS. This is too high of a rate to truly test the dependency's effect on the final output of the NLP pipeline. To address this issue we have hand annotated dependency trees for the WMT 2012 data set, commonly used to judge machine translation systems. Additionally, to improve future parser training and constituent conversions, we have hand corrected the dependency trees in one section of the Penn Treebank.

Within this dissertation, we aim to show both improvements to dependency parsing using ensemble methods for a variety of languages including under-resourced and resource-rich and show how these new dependency parsers effect the overall result in a machine translation pipeline. In addition to these results, we have developed new gold standard dependency trees for the purpose of machine translation. We have also determined an improved standard for constituent conversions through empirical means discovered from manual annotation of a part of the Penn Treebank.