Protokoll zur Sitzung vom 03.07.2017

Vortrag von Anton Serjogin

Thema: Learning String Edit Distance

Anton did not write thesis this semester, so he introduced the topic Learning String Edit Distance in this meeting. String distance is a way of quantifying a dissimilarity of two strings, which is applied in natural language processing and bioinformatics. The most common metric in edit distance, the Levenshtein distance, consists of three operations, which are deletion (i.e. remote -> emote), insertion (i.e. lad -> glad) and substitution (i.e. carrot -> parrot). A process of converting from one form into the other is called transduction, which generates a substitutional, a deletion or an insertion pair.

There are two types of distances: the Viterbi edit distance and the stochastic edit distance. The first one, Viterbi, is a negative logarithm of the probability of the most likely edit sequence for the string pair. The second one, stochastic, is a negative logarithm of the probability of the string pair. The second distance differs from the first in considering the contribution of all ways to simultaneously generate the two strings. If a given string pair has many likely generation paths, then the stochastic distance can be less than the Viterbi distance. There are three variants of the stochastic transduction: parameter tying, finite mixtures and stochastic transducer with memory. Parameter tying is the most widely used edit distance, has only four distinct costs (insertion, deletion, identity and substitution) and has the advantage of requiring less training data to accurately learn the edit costs. Finite mixtures transducer is a linear combination of memoryless transducers defined on the same alphabets. Stochastic transducer with memory is the most powerful model, because of dependency of consecutive operations, so it would condition the probability.

Anton mentioned that the presented algorithm cannot be directly applied to solve string classification problems. To learn a string classifier, they were presented with a corpus of labeled strings, not pairs of similar strings. In this part a stochastic solution to string classification problem allows to automatically and efficiently learn a powerful string classifier from a corpus of labeled strings.

Then Anton said a few words about the using of these techniques, which are applied to the problem of learning the pronunciation of words. The motivation for it is that a given word of a natural language may be pronounced in several different ways, depending on such factors as the dialect, the speaker or the linguistic environment. So they formalize Pronunciation Recognition (PR), which has a six-tuple input, consisting of a set W of syntactic words, an alphabet A of phonological segment, an alphabet B of phonetic segments, a pronouncing lexicon L, a training corpus C of labeled phonetic strings and corpus C' of unlabeled phonetic strings. The output is a set of labels for the testing corpus C'. They conducted five experiments using seven models, consisting of Levenshtein distance as well as six variants resulting from two interpretations (the Viterbi and the stochastic edit distances) of three models. For each interpretation, they built a tied model with only four parameters, an untied model, and a mixture model consisting of a uniform mixture of the tied and untied models. Corpus that they use for this task contains over three million words of spontaneous telephone speech conversations. The first experiment E1 uses the full pronouncing lexicon for all 66 284 words; the second one E2 uses the subset of the pronouncing lexicon for the 9 015 words in the corpus; the third one E3 uses the training corpus only to construct the pronouncing lexicon; E4 uses the entire corpus - both training and testing portions and E5 merges the E1 and E3 lexicons. According to Anton, the principal difference among these five experiments is how much information the training corpus provides about the test corpus. In E1 the pronouncing lexicon provides

weak knowledge of the set of syntactic words that appear in the test corpus. In E2 the pruned pronouncing lexicon provides stronger knowledge of the set of syntactic words that actually appear in the test corpus. In E4 the pronouncing lexicon provides complete knowledge of the set of syntactic words paired with their actual phonetic forms in the test corpus. In experiment E4 the lexicon contains an entry for every sample in the test corpus. Anton presented results of every experiment, showing the word error rate. Both interpretations, the stochastic and the Viterbi distances, have the word error rate of only 9% in E4, which is the best performance. But Levenshtein distance shows very low results. Anton explained it by the fact that the mapping from phonetic forms to syntactic words is many-to-many in E4. Moreover, a pronouncing lexicon that is constructed directly from actual pronunciations offers the possibility of better performance than one constructed in traditional ways.