1 Episodic Markov Model

The Episodic Markov Model (EMM) is a version of Episodic Grammar based on NGrams of arbitrary length. Like a Ngram Model, EMM computes the word transition probability within a test sentence by finding Ngrams (of order k) that it has seen before in the train corpus, and that match the k words up to the current sentence position. However, in EMM the Ngrams seen in the training corpus are not stored in an external table, but locally in so-called Wordlet objects and in a distributed fashion. For every sentence seen in the train corpus, and for every Wordlet corresponding to a word of the sentence, a pointer (also called a trace) is stored inside the Wordlet that encodes information about the position of the word within the sentence. Given a trained EMM, consisting of a collection of Wordlets with traces inside, it is possible to reconstruct every train sentence by following the traces. Moreover, a probability model can be defined based on the stored episodes. P In EMM word transition probability are not estimated beforehand, but computed on the fly: Ngrams that overlap with the current sentence are reconstructed by concatenating words from the traces at derivation time, as one goes through the sentence.

Because NGrams are not represented explicitly in a table but reconstructed only if needed there is no problem of data sparsity, and no probability estimation has to be done beforehand.

More formally, an Episodic Markov Model is a collection of Wordlets, $\{\mathcal{W}_1, \ldots, \mathcal{W}_N\}$, where N is the total number of word types in the corpus. A Wordlet $\mathcal{W}$ refers to a word type, w, and may contain zero or more traces e.

1.0.1 Training

1. create a Wordlet for every unique word type in the train corpus.

2. loop through sentences in the train corpus
   for word w_k in a sentence i, find Wordlet $\mathcal{W}_{w_k}$ corresponding to the word and add a trace to the Wordlet that is encoded $i - k$

1.0.2 Testing

Given test sentence $w_1, \ldots, w_n$ Loop through Wordlets corresponding to words For every trace t in current Wordlet w$_k$*updatecommonhistoryofthetrace* : $if thereisapredecessortrace t_x xinWor$
$1) then CH(t) = CH(t_(k-1))+1 otherwise CH(t) Computetheactivationofthetrace, dependingonthechosena$

$$A(e_{x_i}) = \lambda_0^{CH(e_{x_i}, d)} \tag{1}$$

where λ_0 is a parameter of the model.

Compute transition probability:

The probability of moving from Wordlet $\mathcal{W}_q$ to $\mathcal{W}_{q'}$ in the next step of the derivation is simply the sum of activations of traces that point to $\mathcal{W}_{q'}$, divided by the sum of all activations. Let $E_{\mathcal{W}_q}^{\mathcal{W}_{q'}}$ be the set of traces in Wordlet $\mathcal{W}_q$ that point to Wordlet $\mathcal{W}_{q'}$, and $E_{\mathcal{W}_q}$ the full set of traces in Wordlet $\mathcal{W}_q$. Then, the probability of moving the derivation to Wordlet $\mathcal{W}_{q'}$ is

$$P_{episodic}(t_{q'}|t_q) = \frac{\sum_{e_i \in E_{t_q}^{t_{q'}}} A(e_i)}{\sum_{e_j \in E_{t_q}} A(e_j)} \tag{2}$$

1.1 Training the episodic grammar

To evaluate the concept of episodic grammar quantitatively a probabilistic version is implemented that is trained on a corpus of realistic language. Probabilistic grammars assign probabilities to different parses of a sentence and select the most probable one, hence can be evaluated on their ability to disambiguate between parses. As explained in section **??**, one estimates the parameters of the probabilistic episodic grammar from a treebank, which is a corpus consisting of natural language sentences manually annotated with phrase structure trees.

After deciding on a derivation strategy (i.e., top-down or left-corner), the training proceeds by distributing a trace $e = \langle s, k \rangle$ in every visited treelet t_k of derivation $x = \langle t_0, \ldots, t_k, \ldots, t_n \rangle$ of sentence number s in the treebank. Specifically, given a treebank, then

Create an empty treelet for every unique context free production extracted from the treebank. In case of a left corner derivation one must also create separate treelets for distinct visits to the same production (i.e., after an attach), that is one must distinguish register positions of a treelet. Further, in case of a left corner derivation, create special shift treelets (as described in section **??**) corresponding to the *shift* moves (to terminals) of the left corner parser.

For every treebank parse determine the sequential order of (register-indexed) treelets according to the chosen derivation strategy.

For every step k in the derivation of sentence number s, leave a (register-indexed) trace in the visited treelet, encoded as $\langle s, k \rangle$.

At every derivation step the probability of moving to the next treelet in the derivation can be computed based on the traces in the current treelet and their activations, according to Equation 2.

1.2 Statistical parsing with the episodic grammar

After training the grammar one can use the model to assign probabilities to candidate parses of a new sentence. Given an ongoing derivation d of a sentence, that has arrived at a certain treelet $t_{q,\,r}$, in register position r, one defines the probability of continuing the derivation to any other treelet $t_{q',\,s}$ in register position s based on the activation values of the episodic traces of earlier derivations stored in treelet $t_{q,\,r}$. The activation $A(e_{x_i})$ of the trace e_{x_i} (in $t_{q,\,r}$) of earlier derivation x is a function of the common history $CH(e_{x_i}, d)$ of derivation x (of which e_{x_i} is the i^{th} trace) with the ongoing derivation d. The CH is simply given by the number of derivation steps (i.e., treelets) that the stored derivation x and the pending derivation d have shared the same path before arriving at $t_{q,\,r}$. Episodic traces that share a long common history should contribute relatively much to the parser decision.

This probability can be computed dynamically, while simultaneously updating the common histories (and activations) of all traces at every step of the derivation. Let t_q and $t_{q'}$ be two successive treelets in the pending derivation d, and let $e' = \langle s, j \rangle$ be a trace stored in $t_{q'}$. Then its CH is updated according to

$$CH(e', d_{q'}) = CH(e, d_q) + 1 \tag{3}$$

if there exists a trace $e = \langle s, j-1 \rangle$ in t_q (i.e., a predecessor of e'). Otherwise, $CH(e', d_{q'}) = 0$.