Thesis Meeting 8

kense, for the thesis

Previously, on the thesis...

1. Cost LTL
2. Longer dependency sequences for frequency-based analysis

LTL for cost functions

- Extending LTL expression with notions of cost functions (counting occurrences, length of intervals, etc)
- Operator is defined $U^{\leq N}$ which means a formula $x\ U^{\leq N}\ y$ implies x holds until y, except at most N times (where x is allowed to not hold before y).
- The idea is to allow some mistakes, but bound global mistakes

Definition

The defined grammar is as follows:

$\boldsymbol{a}$, implies the current symbol is a.

$\wedge$, $\vee$, implies conjunction and disjunction.

$\boldsymbol{X\varphi}$, implies the next symbol is $\boldsymbol{\varphi}$.

$\boldsymbol{\varphi U \omega}$, implies $\boldsymbol{\varphi}$ holds until $\boldsymbol{\omega}$.

$\boldsymbol{\varphi U^{\leq N} \omega}$ implies $\boldsymbol{\varphi}$ holds until $\boldsymbol{\omega}$, with $\boldsymbol{N}$ exceptions.

$\boldsymbol{\Omega}$, implies the end of the sequence.

Until Implementation

- For simplicity in design, we operate on a sequence, and find all instances where $\varphi U \omega$ are True.

Input(s): trace, labels, labels (where the trace is a list of labels, and the labels correspond to φ and ω respectively, where each is a list of labels to be checked).

Output: list of tuples (indicating the starting index and ending index of the sub-sequence that matches $\varphi U \omega$)

Until$^{\leqq N}$ Implementation

- We use the same logic as the Until implementation, with the addition that we run a counter N to reset tracking if we violate the bounds

Input(s): trace, labels, labels, N (where N is the number of times $\boldsymbol{\varphi}$ is allowed to not hold until $\boldsymbol{\omega}$).

Output: list of tuples (*s,e,c,l*) (where *s* and *e* mark the sequence satisfying the LTL-N expression, *c* is the number of labels not belonging to $\boldsymbol{\varphi}$, and *l* is the length of the sequence).

Defining Sub-conversations

- We can now define sub-conversations using LTL expressions on labels.
- Our first two preliminary sub-conversations
  - Give.Monologue
  - Recall.From.Memory
- We first define from intuition, then we can run different combinations.

Give.Monologue

A Sub-Conversation is defined as follows:

$$\beta = \varphi \, U^{\leqq N} \, \omega$$

Where φ, ω are sets, and $\varphi, \omega \in \ell$, our set of labels, $U^{\leqq N}$ is the defined Until-N function.

Give.Monologue is defined as follows:

Give.Monologue = (give.statement $\vee$ give.opinion) $U^{\leqq N}$ (respond.agree $\vee$ respond.deny)

Recall.From.Memory

Recall.From.Memory is defined as follows:

Recall.From Memory = (recall) $U^{\leq N}$ (respond.agree $\vee$ respond.deny)

- We suspect there might be better/more accurate ways to describe these sub-conversations as reflected by the provided eventlog.
- Additional considerations are needed here to consider the definitions based on provided eventlog data (i.e: can we derive a group of sub-conversations based on properties bounded on N and sets φ, ω instead of manually?)

Generating Sub-Conversations

- Sub-conversation sequences shouldn’t be too short (length 2 or 3 is probably not a good indicator of long-term dependency).
- Sub-conversation sequences shouldn’t be too long.
- Finding the right value *N* will make frequency analysis more accurate (this is the assumption).
- Try different labels in sets $\boldsymbol{\varphi, \omega}$ for our definition.

Generating Sub-Conversations

- Starting with expressions in the form:

  $\varphi \, U^{\leqq N} \, \omega$

Where φ, ω are sets of length 1, spanning the set of labels ℓ

- These should intuitively yield less interesting sub-sequences, which should mostly consist of sub-sequences that are too short, essentially reflecting the $X\varphi$ function.

Generating Sub-Conversations

- Most likely want to expand the combinations of labels in the $\boldsymbol{\omega}$ set (second label set), as the $\boldsymbol{\omega}$ set in the definition for the until-N function is the set of labels that binds the sequences.
- Look for cases where N violations are < *X%* of the sequence length.
- Potential for using Until and Until-N expressions to soft-label different speakers (sequences that fit the properties established before might very often contain different speakers for labels of the sets $\boldsymbol{\varphi,\omega}$).

Goals

1. Experiment on best values (N, min/max length) for discovering sub-conversation trends.
2. Write functions to accomplish best values/discovery.
3. Establish some explanations for sub-conversation trends.
4. <<reserved>>