Sequential Monte Carlo Text Normalization

Çağıl Uluşahin Sönmez and Mine Öğretir

Abstract

In our project we implemented an unsupervised text-normalisation algorithm that uses sequential importance sampling, which is a member of sequential Monte Carlo family.[*] Local context and surface similarity between the source strings and the target strings are the main sources of information to be used.

Using supervised learning techniques is not practical because of the changing non-standard tokens and difficulty obtaining labeled data and impossibility of using standard supervised techniques because of large label set.

Model

Given a set of source language sentences:

$$S = \{s_1, s_2, s_3, \ldots\}$$

Transduce them into target-language sentences:

$$T = \{t_1, t_2, t_3, \ldots\}$$

Target language model: P(t)

Vocabularies: v_S, v_T

A log-linear model to score the strings:

$$P(\mathbf{s}|\mathbf{t};\theta) \propto \exp\left(\theta^{\mathsf{T}}\mathbf{f}(\mathbf{s},\mathbf{t})\right)$$

We assume that we can write the target language as an N-gram language model. With this assumption the target language model becomes the prior distribution and the following defines the likelihood function:

$$P(\mathbf{s}|\mathbf{t};\theta) = \prod_n^N \frac{\exp\left(\theta^{\mathsf{T}}\mathbf{f}(s_n,t_n)\right)}{Z(t_n)}$$

$$Z(t_n) = \sum_s \exp\left(\theta^{\mathsf{T}}\mathbf{f}(s,t_n)\right). \qquad \omega_n^k \propto \frac{P(\mathbf{t}_{1:n}^k|\mathbf{s}_{1:n})}{Q(\mathbf{t}_{1:n}^k|\mathbf{s}_{1:n})}$$

Then the posterior probability P(t|s) can be calculated using sequential importance sampling.
Proposal Distribution and unnormalized weight :

$$Q(t_n^k|s_n,t_{n-1}^k) \stackrel{def}{=} \frac{P(s_n|t_n^k)Z(t_n^k)P(t_n^k|t_{n-1}^k)}{\sum_{t'}P(s_n|t')Z(t')P(t'|t_{n-1}^k)} = \frac{\exp\left(\theta^{\mathsf{T}}\mathbf{f}(s_n,t_n)\right)P(t_n^k|t_{n-1}^k)}{\sum_{t'}\exp\left(\theta^{\mathsf{T}}\mathbf{f}(s_n,t')\right)P(t'|t_{n-1}^k)}$$

$$\tilde{\omega}_n^k = \omega_{n-1}^k \frac{\sum_{t'}\exp\left(\theta^{\mathsf{T}}\mathbf{f}(s_n,t')\right)P(t'|t_{n-1}^k)}{Z(t_n^k)}$$

At each step, and for each hypothesis k, a new target word is sampled from a proposal distribution, and the weight of the hypothesis is updated.
O(#| vT |+#| vS |)

Example Tweet

Source Tweet: “smh jaz what was *u doin*? RT @redliteroxanne: my body hurt like crazy smh”

Target Tweet: “smh jaz what was *you doing*? RT @redliteroxanne: my body hurt like crazy smh”

Motivation

- Unsupervised
- Low-resource
- Featurized
- Context-driven
- Holistic

A randomised algorithm, which approximates the necessary feature expectation through weighted samples.

Algorithm

1) Find all possible candidates, given t_{n-1}^k and s_n and choose top 10 according $Q(t_n^k \mid s_n, t_{n-1}^k)$ as given in the model, where f(s,t) is feature function and θ is weighted sum of features.

$$\theta \cdot f(s, t)$$

f(s,t) = [pair-score(<s,t>),similarity(<s,t>)]

2) Calculate w_n^k for each k

3) Select top 1 w_N^k

Feature Name	Description
word-word pair	A set of binary features for each source/target word pair <s,t>
string similarity	A set of binary features indicating whether s is one of the top N strings similar non-standard words of t, for N {5,10,25,50,100,250,500,1000}

Feature Examples: is_first_letter_equal, is_last_letter_equal, common_letter_ratio etc.

Conclusion

Sequential Monte Carlo algorithms permit efficient computation and they are parallelizable and the number of samples provides a fine tuning between accuracy and speed. The model creates word-to-word relationships through features. Sequential Monte Carlo allows to overcome the challenge of the large label space.

[*] Yang, Yi, and Jacob Eisenstein. "A Log-Linear Model for Unsupervised Text Normalization." *EMNLP*. 2013.