NLP for Social Media

Lecture 3: Normalization with the Noisy Channel

Monojit Choudhury

Microsoft Research Lab, monojitc@microsoft.com

What will we learn?

- Noisy-channel approach to normalization
- Language Modeling
- Channel Modeling

Spell-checking or edit-distance based approaches does not work well for orthographic normalization of SM data.
What model or approach might work?

The Translation Metaphor

$S: s_1\ s_2\ \ldots\ s_n$

Standard Language

NOISY CHANNEL

$T: t_1\ t_2\ \ldots\ t_m$

SM Language

$$S^* = \delta(T) = \underset{S}{\operatorname{argmax}}\ Pr(S|T)$$

$$= \underset{S}{\operatorname{argmax}}\ Pr(T|S)Pr(S)$$

Channel Model

Language Model

Intuition behind the NC model

T: *2day I lost my fne*

Use the channel model to find the set of possible Standard language words $\{S_1, S_2, S_3 \ldots\}$ such that $Pr(T|S) > p$

TL token (SL token)	Decoder output ($-\log(P(w|t))$)
2day (today)	today (3.02), stay (11.46), away (13.13), play (13.14), clay (13.14)
fne (phone)	fine (3.52), phone (5.13), funny (6.26), fined (6.51), fines (6.72)

Intuition behind the NC model

T: *2day I lost my fne*

Generate candidates:

- *S1: today I lost my fine*
- *S2: today I lost my phone*
- *S3: stay I lost my fine*
- *S4: stay I lost my phone*

Compute and rank by P(T|S)P(S)

TL token (SL token)	Decoder output ($-\log(P(w\|t))$)
2day (today)	today (3.02), stay (11.46), away (13.13), play (13.14), clay (13.14)
fne (phone)	fine (3.52), phone (5.13), funny (6.26), fined (6.51), fines (6.72)

Some commonly made assumptions

$$S^* = argmax\ P(T|S)P(S)$$
$$= argmax\ [\prod_{i=1}^{n} P(t_i|s_i)]P(S)$$
$$= argmax\ [\prod_{i=1}^{n} P(t_i|s_i)][\prod_{i=1}^{n} P(s_i\ |s_{i-1}s_{i-2}s_{i-3\dots})]$$
$$= argmax\ \prod_{i=1}^{n}[P(t_i|s_i)\mathrm{P}(s_i|s_{i-1}s_{i-2}s_{i-3\dots})]$$
$$= argmax\ \prod_{i=1}^{n}[P(t_i|s_i)\mathrm{P}(s_i|s_{i-1}s_{i-2})]$$

Word-by-word normalization

Word choice depends only on the history

Markov assumption: limited history

$S: s_1\ s_2\ \dots\ s_n$ $\Rightarrow$ NOISY CHANNEL $\Rightarrow$ $T: t_1\ t_2\ \dots\ t_m$

Standard Language

SM Language

Noisy Channel Model is extensively used in

- Speech Recognition
- Machine Translation
- Transliteration
- Paraphrasing
- Parts-of-speech Tagging

(and a variety of sequence labeling problems)

What will we learn?

- Noisy-channel approach to normalization
- Language Modeling
- Channel Modeling

N-gram Language Model

$$S^* = argmax \prod_{i=1}^{n} [P(t_i | s_i) \boxed{\mathrm{P}(s_i | s_{i-1} s_{i-2})]}$$

your ?? (bigram)

at your ?? (trigram)

task at your ?? (4-gram)

task at your ?? convenience

Please finish this task at your earliest convenience

How do you learn the n-gram probabilities?

Estimating N-gram Probabilities

$$S^* = argmax \prod_{i=1}^{n}[P(t_i|s_i)\boxed{\mathrm{P}(s_i|s_{i-1}s_{i-2})]}$$

- From text corpus (of *S*) that closely resembles the language usage of the application domain.
- Larger the N, better the LM, provided we have an appropriately large corpus to estimate the probabilities.
- Estimating probabilities for unseen N-grams:
  - Fallback to lower N-grams, till unigram.
  - Smoothing

Smoothing

$$S^{*} = argmax \prod_{i=1}^{n} [P(t_i | s_i) \mathrm{P}(s_i | s_{i-1} s_{i-2})]$$

- Zero probabilities are dangerous
- Smoothing is the process of estimating the probabilities of unseen events, based on the distribution of seen events. It eliminates zero probabilities.

$$P(s_i | s_{i-1} s_{i-2}) = \frac{count(s_{i-2} s_{i-1} s_i)}{count(s_{i-2} s_{i-1})}$$

d is the number of unique trigrams seen that begin with $s_{i\text{-}2} s_{i\text{-}1}$

This is called Add-one Smoothing (and more generally Additive Smoothing)

Evaluating Language Models

- **Task completion**: Good models yield good performance for the end-application
  - Speech Recognition
  - Machine Translation
  - Dialogue Modeling
  - Text Prediction, etc.

- **Perplexity**: A measure of how well a LM learnt from the training corpus C1 is able to predict the distribution of the words (or n-grams) observed in the test corpus C2.

Perplexity of Language Model

- Entropy of a probability distribution:

$$H(p) = \sum_{\substack{all\ events \\ in\ \boldsymbol{p}}} -p_i \log_2(p_i)$$

- Perplexity of a distribution:

$$Perp(p) = 2^{H(p)}$$

- Perplexity of LM (or probability model):

$$Perp(p, q) = 2^{-\sum_{all\ i} p_i \log(q_i)}$$

Estimated from Test Data

Estimated from Training data (LM)

Higher the entropy or perplexity, higher the unpredictability

Procuring Corpora for LM

$$S^{*} = argmax \prod_{i=1}^{n} [P(t_i | s_i) \boxed{\mathrm{P}(s_i | s_{i-1} s_{i-2})]}$$

- LM should be estimated from text corpus (of *S*) that closely resembles the language usage of the application domain.

How to get Standard Language Corpora corresponding to the Social Media Domain?

- Create by Manual normalization: requires time and effort
- Scrape from Social Media: no human annotation, but non-trivial
- Domain Adaptation: Interpolate between a domain specific and standard language domain.

What will we learn?

- Noisy-channel approach to normalization
- Language Modeling
- Channel Modeling

The Channel Model

$$S^* = argmax \prod_{i=1}^{n} [\boxed{P(t_i \,|\, s_i)} \mathrm{P}(s_i \,|\, s_{i-1} s_{i-2})]$$

$$P(t|s) = \prod_{j=1}^{k} P(\tau_j|\sigma_j),$$

where t = $\tau_1\ \tau_2 \ldots \tau_{k,}$ s =$\sigma_1\sigma_2 \ldots \sigma_k$

- Problem: The characters of *t* and *s* may not have one-to-one correspondence.
- Solution: τ and σ could be defined as group of letters, rather than single letters
- How do you discover and split the words into meaningful segments of letters (aka character n-grams)?

Let's work out *tomorrow*

t	$\tau_1\ \tau_2 \dots \tau_k$	$\sigma_1\sigma_2 \dots \sigma_k$	t	$\tau_1\ \tau_2 \dots \tau_k$	$\sigma_1\sigma_2 \dots \sigma_k$
2moro			tomm		
tomoz			tomo		
tomoro			tomorow		
tomrw			2mro		
tom			morrow		
tomra			tomor		
tomorrow			tmorro		
tomora			moro		

Let's work out *tomorrow*

t	$\tau_1\ \tau_2 \ldots \tau_k$	$\sigma_1\sigma_2 \ldots \sigma_k$	t	$\tau_1\ \tau_2 \ldots \tau_k$	$\sigma_1\sigma_2 \ldots \sigma_k$
2moro	2|m|o|r|o	to|m|o|rr|ow	tomm	t|o|mm|$	t|o|m|orrow
tomoz	t|o|m|o|z	t|o|m|o|rrow	tomo	t|o|m|o|$	t|o|m|o|rrow
tomoro	t|o|m|o|r|o	t|o|m|o|rr|ow	tomorow	tomorow	tomorrow
tomrw	t|o|m|r|w	t|o|mo|rro|w	2mro	2|m|r|o	to|mo|rr|ow
tom	t|o|m|$	t|o|m|orrow	morrow	$|m|o|r|r|o|w	to|m|o|r|r|o|w
tomra	t|o|m|r|a	t|o|mo|rr|ow	tomor	t|o|m|o|r	t|o|m|o|rrow
tomorrow	tomorrow	tomorrow	tmorro	t|m|o|r|r|o	to|m|o|r|r|ow
tomora	t|o|m|o|r|a	t|o|m|o|rr|ow	moro	$|m|o|r|o	to|m|o|rr|ow

Compute $P(\tau_j = "o" | \sigma_j = "ow")$

t	$\tau_1\ \tau_2 \dots \tau_k$	$\sigma_1\sigma_2 \dots \sigma_k$	t	$\tau_1\ \tau_2 \dots \tau_k$	$\sigma_1\sigma_2 \dots \sigma_k$
2moro	2|m|o|r|o	to|m|o|rr|ow	tomm	t|o|mm|$	t|o|m|orrow
tomoz	t|o|m|o|z	t|o|m|o|rrow	tomo	t|o|m|o|$	t|o|m|o|rrow
tomoro	t|o|m|o|r|o	t|o|m|o|rr|ow	tomorow	tomorow	tomorrow
tomrw	t|o|m|r|w	t|o|mo|rro|w	2mro	2|m|r|o	to|mo|rr|ow
tom	t|o|m|$	t|o|m|orrow	morrow	$|m|o|r|r|o|w	to|m|o|r|r|o|w
tomra	t|o|m|r|a	t|o|mo|rr|ow	tomor	t|o|m|o|r	t|o|m|o|rrow
tomorrow	tomorrow	tomorrow	tmorro	t|m|o|r|r|o	to|m|o|r|r|ow
tomora	t|o|m|o|r|a	t|o|m|o|rr|ow	moro	$|m|o|r|o	to|m|o|rr|ow

Compute $P(\tau_j = "o" | \sigma_j = "ow")$

5/7

t	$\tau_1\ \tau_2 \ldots \tau_k$	$\sigma_1\sigma_2 \ldots \sigma_k$	t	$\tau_1\ \tau_2 \ldots \tau_k$	$\sigma_1\sigma_2 \ldots \sigma_k$
2moro	2|m|o|r|o	to|m|o|rr|ow	tomm	t|o|mm|$	t|o|m|orrow
tomoz	t|o|m|o|z	t|o|m|o|rrow	tomo	t|o|m|o|$	t|o|m|o|rrow
tomoro	t|o|m|o|r|o	t|o|m|o|rr|ow	tomorow	tomorow	tomorrow
tomrw	t|o|m|r|w	t|o|mo|rro|w	2mro	2|m|r|o	to|mo|rr|ow
tom	t|o|m|$	t|o|m|orrow	morrow	$|m|o|r|r|o|w	to|m|o|r|r|o|w
tomra	t|o|m|r|a	t|o|mo|rr|ow	tomor	t|o|m|o|r	t|o|m|o|rrow
tomorrow	tomorrow	tomorrow	tmorro	t|m|o|r|r|o	to|m|o|r|r|ow
tomora	t|o|m|o|r|a	t|o|m|o|rr|ow	moro	$|m|o|r|o	to|m|o|rr|ow

Estimating the channel model

$$S^* = argmax \prod_{i=1}^{n} [\boxed{P(t_i \mid s_i)} \mathrm{P}(s_i \mid s_{i-1} s_{i-2})]$$

Bayesian Approach:

$$P(t|s) = \sum_{\substack{all\ possible \\ segmentations}} \prod_{j=1}^{k} P(\tau_j|\sigma_j)$$

Maximum Likelihood/frequentist Approach:

$$P(t|s) = \max_{\substack{all\ possible \\ segmentations}} \prod_{j=1}^{k} P(\tau_j|\sigma_j)$$

Depending on which approach one chooses, one can accordingly estimate the probabilities from the data.

Data for Learning the Channel Model

- A corpus of Social Media text, and the corresponding standard language forms (normalized forms)
  - It is sufficient to have only word pairs (with frequency of occurrence).
- Corpus creation
  - Is usually a manual effort
  - Either controlled (more expensive but accurate) or crowdsourcing (cheap and fast, but noisy)
  - Can you scrape <*t*,*s*> pairs from social media text?
- Can you estimate the channel model without any word-pair data?

Different Avatars of the Noisy Channel Model

- Hidden-Markov Models to combine linguistic information with the NC model [Choudhury et al., 2007]
- Conditional Random Fields [Liu et al., 2012]

Handling data scarcity:

- Semi-supervised and unsupervised techniques
- Automatic extraction of training data [Hassan & Menzes, 2014]

Summary

- The translation metaphor, modeled as the Noisy Channel, provides a useful and potent approach for normalization.

$$S^{*} = argmax \prod_{i=1}^{n} [P(t_i | s_i)\mathrm{P}(s_i | s_{i-1} s_{i-2})]$$

- Language Model predicts the probability of a sequence of words (or other linguistic units) and can be estimated and interpolated from appropriate datasets.
- Word n-gram language models are simple, yet very useful.
- There are several ways to model and learn the channel characteristics: character n-gram based, HMMs, CRFs, etc.

Suggested Readings & References

For Language Modeling:

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 2009 http://www.cs.colorado.edu/~martin/slp.html (*Ch 4: N-grams)*

For Noisy Channel Model based Normalization:

AiTi Aw , Min Zhang , Juan Xiao and Jian Su. “A phrase-based statistical model for SMS text normalization”, *COLING/ACL* 2006

Choudhury, Monojit, et al. "Investigation and modeling of the structure of texting language." *International Journal of Document Analysis and Recognition (IJDAR)* 10.3-4 (2007): 157-174.

Kaufmann, Max, and Jugal Kalita. "Syntactic normalization of twitter messages." *ICON,* 2010.

Liu, Fei, Fuliang Weng, and Xiao Jiang. "A broad-coverage normalization system for social media language." *ACL* 2012.