Content Realization

Ling573
Systems and Applications
May 12, 2015

Roadmap

- Content realization
  - Goals
  - Broad approaches
  - Implementation exemplars

Goals of Content Realization

- Abstractive summaries:
  - Content selection works over concepts
  - Need to produce important concepts in fluent NL
- Extractive summaries:
  - Already working with NL sentences
  - Extreme compression: e.g 60 byte summaries: headlines
  - Increase information:
    - Remove verbose, unnecessary content
    - More space left for new information
  - Increase readability, fluency
    - Present content from multiple docs, non-adjacent sents
  - Improve content scoring
    - Remove distractors, boost scores: i.e. % signature terms in doc

Broad Approaches

- Abstractive summaries:
  - Complex Q-A: template-based methods
  - More generally: full NLG: concept-to-text
- Extractive summaries:
  - Sentence compression:
    - Remove “unnecessary” phrases:
      - Information? Readability?
  - Sentence reformulation:
    - Reference handling
      - Information? Readability?
  - Sentence fusion: Merge content from multiple sents

Sentence Compression

- Main strategies:
  - Heuristic approaches
    - Deep vs Shallow processing
    - Information- vs readability- oriented
  - Machine-learning approaches
    - Sequence models
      - HMM, CRF
    - Deep vs Shallow information
  - Integration with selection
    - Pre/post-processing; Candidate selection: heuristic/learned

Form	CLASSY	ISCI	UMd	SumBasic+	Cornell
Initial Adverbials	Y	M	Y	Y	Y
Initial Conj	Y		Y	Y	
Gerund Phr.	Y	M	M	Y	M
Rel clause appos	Y		M	Y	Y
Other adv	Y		N		
Numeric: ages,	Y		N		
Junk (byline, edit)	Y		N		Y
Attributives	Y	Y		Y	Y
Manner modifiers	M	Y	M		Y
Temporal modifiers	M	Y	Y		Y
POS: det, that, MD			Y		
XP over XP			Y		
PPs (w/, w/o constraint)			Y		
Preposed Adjuncts			Y		
SBARs			Y		M
Conjuncts			Y		
Content in parentheses		Y			Y

Shallow, Heuristic

- CLASSY 2006
  - Pre-processing! Improved ROUGE
    - Previously used automatic POS tag patterns: error-prone
- Lexical & punctuation surface-form patterns
  - “function” word lists: Prep, conj, det; adv, gerund; punct
- Removes:
  - Junk: bylines, editorial
  - Sentence-initial adv, conj phrase (up to comma)
  - Sentence medial adv (“also”), ages
  - Gerund (-ing) phrases
  - Rel. clause attributives, attributions w/o quotes
- Conservative: < 3% error (vs 25% w/POS)

Deep, Minimal, Heuristic

- ICSI/UTD:
  - Use an Integer Linear Programming approach to solve
- Trimming:
  - Goal: Readability (not info sqeezing)
  - Removes temporal expressions, manner modifiers, “said”
    - Why?: “next Thursday”
  - Methodology: Automatic SRL labeling over dependencies
    - SRL not perfect: How can we handle?
    - Restrict to high-confidence labels
    - Improved ROUGE on (some) training data

Example

A ban against bistros providing plastic bags free of charge will be lifted at the beginning of March.

Example

A ban against bistros providing plastic bags free of charge will be lifted at the beginning of March.

A ban against bistros providing plastic bags free of charge will be lifted.

Deep, Extensive, Heuristic

- Both UMD & SumBasic+
  - Based on output of phrase structure parse
  - UMD: Originally designed for headline generation
  - Goal: Information squeezing, compress to add content
- Approach: (UMd)
  - Ordered cascade of increasingly aggressive rules
    - Subsumes many earlier compressions
    - Adds headline oriented rules (e.g. removing MD, DT)
    - Adds rules to drop large portions of structure
      - E.g. halves of AND/OR, wholescale SBAR/PP deletion

Integrating Compression & Selection

- Simplest strategy: (Classy, SumBasic+)
  - Deterministic, compressed sentence replaces original
- Multi-candidate approaches: (most others)
  - Generate sentences at multiple levels of compression
    - Possibly constrained by: compression ratio, minimum len
      - E.g. exclude: < 50% original, < 5 words (ICSI)
  - Add to original candidate sentences list
  - Select based on overall content selection procedure
    - Possibly include source sentence information
    - E.g. only include single candidate per original sentence

Multi-Candidate Selection

- (UMd, Zajic et al. 2007, etc)

- Sentences selected by tuned weighted sum of feats
  - Static:
    - Position of sentence in document
    - Relevance of sentence/document to query
    - Centrality of sentence/document to topic cluster
      - Computed as: IDF overlap or (average) Lucene similarity
    - # of compression rules applied
  - Dynamic:
    - Redundancy: $S = \Pi_{w_i \text{ in } S} \lambda P(w|D) + (1 - \lambda) P(w|C)$
    - # of sentences already taken from same document

- Significantly better on ROUGE-1 than uncompressed
  - Grammaticality lousy (tuned on headlinese)

Learning Compression

- Cornell (Wang et al, 2013)
- Contrasted three main compression strategies
  - Rule-based
  - Sequence-based learning
  - Tree-based, learned models
- Resulting sentences selected by SVR model

Sequence-based Compression

- View as sequence labeling problem
  - Decision for each word in sentence: keep vs delete
  - Model: linear-chain CRF
    - Labels: B-retain, I-retain, O (token to be removed)
  - Features:
    - “Basic” features: word-based
    - Rule-based features: if fire, force to O
    - Dependency tree features: Relations, depth
    - Syntactic tree features: POS, labels, head, chunk
    - Semantic features: predicate, SRL
      - Include features for neighbors

Compression Corpus

- (Clark & Lapata, 2008)
- Manually created corpus:
  - Written: 82 newswire articles (BNC, ANT)
  - Spoken: 50 stories from HUB-5 broadcast news
- Annotators created compression sentence by sentence
  - Could mark as not compressable
- http://jamesclarke.net/research/resources/

Feature Set

- Detail:

Basic Features	**Syntactic Tree Features**
first 1/3/5 tokens (toks)?	POS tag
last 1/3/5 toks?	parent/grandparent label
first letter/all letters capitalized?	leftmost child of parent?
is negation?	second leftmost child of parent?
is stopword?	is headword?
Dependency Tree Features	in NP/VP/ADVP/ADJP chunk?
dependency relation (dep rel)	**Semantic Features**
parent/grandparent dep rel	is a predicate?
is the root?	semantic role label
has a depth larger than 3/5?	

Rule-Based Features

For each rule in Table 2 , we construct a corresponding feature to indicate whether the token is identified by the rule.

Tree-based Compression

- Given a phrase-structure parse tree,
  - Determine if each node is: removed, retained, or partial
- Issues:
  - # possible compressions exponential
  - Need some local way of scoring a node
  - Need some way of ensuring consistency
    - I.e. can't have retain over remove
  - Need to ensure grammaticality

Tree-based Compression

- Given a phrase-structure parse tree,
  - Determine if each node is: removed, retained, or partial
- Issues & Solutions:
  - # possible compressions exponential
    - Order parse tree nodes (here post-order)
    - Do beam search over candidate labelings
  - Need some local way of scoring a node
    - Use MaxEnt to compute probability of label
  - Need some way of ensuring consistency
    - Restrict candidate labels based on context
  - Need to ensure grammaticality
    - Rerank resulting sentences using n-gram LM

Features

- Basic features:
  - Analogous to those for sequence labeling
- Enhancements:
  - Context features: decisions about child, sibling nodes
  - Head-driven search:
    - Reorder so head nodes at each level checked first
      - Why? If head is dropped, shouldn't keep rest
      - Revise context features

Summarization Features

- (aka MULTI in paper)
- Calculated based on current decoded word sequence W
- Linear combination of:
  - Score under MaxEnt
  - Query relevance:
    - Proportion of overlapping words with query
  - Importance: Average sumbasic score over W
  - Language model probability
  - Redundancy: 1 --- proportion of words overlapping summ

Summarization Results

	DUC 2006			DUC 2007		
System	**C Rate**	**R-2**	**R-SU4**	**C Rate**	**R-2**	**R-SU4**
Best DUC system	–	9.56	15.53	–	12.62	17.90
Davis et al. (2012)	–	10.2	15.2	–	12.8	17.5
SVR	100%	7.78	13.02	100%	9.53	14.69
LambdaMART	100%	9.84	14.63	100%	12.34	15.62
Rule-based	78.99%	10.62 ∗†	15.73 †	78.11%	13.18†	18.15†
Sequence	76.34%	10.49 †	15.60 †	77.20%	13.25†	18.23†
Tree (BASIC + $Score_{Basic}$)	70.48%	10.49 †	15.86 †	69.27%	13.00†	18.29†
Tree (CONTEXT + $Score_{Basic}$)	65.21%	10.55 ∗†	16.10 †	63.44%	12.75	18.07†
Tree (HEAD + $Score_{Basic}$)	66.70%	10.66 ∗†	16.18 †	65.05%	12.93	18.15†
Tree (HEAD + MULTI)	70.20%	**11.02** ∗†	**16.25** †	73.40%	**13.49**†	**18.46**†

Compression Results

System	C Rate	Uni-Prec	Uni-Rec	Uni-F1	Rel-F1
HedgeTrimmer	57.64%	0.72	0.65	0.64	0.50
McDonald (2006)	70.95%	0.77	0.78	*0.77*	0.55
Martins and Smith (2009)	71.35%	0.77	0.78	*0.77*	0.56
Rule-based	87.65%	0.74	0.91	0.80	0.63
Sequence	70.79%	0.77	0.80	*0.76*	0.58
Tree (BASIC)	69.65%	0.77	0.79	0.75	0.56
Tree (CONTEXT)	67.01%	**0.79**	0.78	*0.76*	0.57
Tree (HEAD)	68.06%	**0.79**	0.80	*0.77*	**0.59**

Discussion

- Best system incorporates:
  - Tree structure
  - Machine learning
  - Summarization features
- Rule-based approach surprisingly competitive
  - Though less aggressive in terms of compression
- Learning based approaches enabled by sentence compression corpus

General Discussion

- Broad range of approaches:
  - Informed by similar linguistic constraints
  - Implemented in different ways:
    - Heuristic vs Learned
    - Surface patterns vs parse trees vs SRL
  - Even with linguistic constraints
    - Often negatively impact linguistic quality
    - Key issue: errors in linguistic analysis
      - POS taggers → Parsers → SRL, etc

General Discussion

- Compression has range of uses:
  - Removing irrelevant information for selection
  - Improving readability
  - Allowing inclusion of more information
- Slightly different strategies for each