Question Processing: Formulation & Expansion

Ling573
NLP Systems and Applications
May 8, 2014

Roadmap

- Query processing
  - Query reformulation
  - Query expansion
    - WordNet-based expansion
    - Stemming vs morphological expansion
    - Machine translation & paraphrasing for expansion

Deeper Processing for Query Formulation

- MULDER (Kwok, Etzioni, & Weld)
- Converts question to multiple search queries
  - Forms which match target
  - Vary specificity of query
    - Most general bag of keywords
    - Most specific partial/full phrases
  - Generates 4 query forms on average
- Employs full parsing augmented with morphology

Question Parsing

- Creates full syntactic analysis of question
  - Maximum Entropy Inspired (MEI) parser
    - Trained on WSJ
- Challenge: Unknown words
  - Parser has limited vocabulary
    - Uses guessing strategy
      - Bad: "tungsten" → number
- Solution:
  - Augment with morphological analysis: PC-Kimmo
  - If PC-KIMMO fails? Guess Noun

Syntax for Query Formulation

- Parse-based transformations:
  - Applies transformational grammar rules to questions
  - Example rules:
    - Subject-auxiliary movement:
      - Q: Who was the first American in space?
      - Alt: was the first American…; the first American in space was
    - Subject-verb movement:
      - Who shot JFK? => shot JFK
    - Etc

More General Query Processing

- WordNet Query Expansion
  - Many lexical alternations: ‘How tall’ → ‘The height is’
  - Replace adjectives with corresponding ‘attribute noun’
- Verb conversion:
  - Morphological processing
    - DO-AUX .... V-INF → V+inflection
    - Generation via PC-KIMMO
- Phrasing:
  - Some noun phrases should treated as units, e.g.:
    - Proper nouns: “White House”; phrases: “question answering”
- Query formulation contributes significantly to effectiveness

Query Expansion

Query Expansion

- Basic idea:
  - Improve matching by adding words with similar meaning/similar topic to query
- Alternative strategies:
  - Use fixed lexical resource
    - E.g. WordNet
  - Use information from document collection
    - Pseudo-relevance feedback

WordNet Based Expansion

- In Information Retrieval settings, mixed history
  - Helped, hurt, or no effect
  - With long queries & long documents, no/bad effect
- Some recent positive results on short queries
    - E.g. Fang 2008
  - Contrasts different WordNet, Thesaurus similarity
  - Add semantically similar terms to query
    - Additional weight factor based on similarity score

Similarity Measures

- Definition similarity: $S_{def}(t_1, t_2)$
  - Word overlap between glosses of all synsets
    - Divided by total numbers of words in all synsets glosses
- Relation similarity:
  - Get value if terms are:
    - Synonyms, hypernyms, hyponyms, holonyms, or meronyms
- Term similarity score from Lin's thesaurus

Results

- Definition similarity yields significant improvements
  - Allows matching across POS
  - More fine-grained weighting than binary relations

- Evaluated on IR task with MAP

	BL	Def	Syn	Hype	Hypo	Mer	Hol	Lin	Com
MAP	0.19	0.22	0.19	0.19	0.19	0.19	0.19	0.19	0.21
Imp		16%	4.3%	0	0	0.5%	3%	4%	15%

Managing Morphological Variants

- Bilotti et al. 2004
- “What Works Better for Question Answering: Stemming or Morphological Query Expansion?”
- Goal:
  - Recall-oriented document retrieval for QA
    - Can’t answer questions without relevant docs
- Approach:
  - Assess alternate strategies for morphological variation

Question

- Comparison
  - Index time stemming
    - Stem document collection at index time
    - Perform comparable processing of query
    - Common approach
      - Widely available stemmer implementations: Porter, Krovetz
  - Query time morphological expansion
    - No morphological processing of documents at index time
    - Add additional morphological variants at query time
      - Less common, requires morphological generation

Prior Findings

- Mostly focused on stemming
- Mixed results (in spite of common use)
  - Harman found little effect in ad-hoc retrieval: Why?
    - Morphological variants in long documents
    - Helps some, hurts others: How?
      - Stemming captures unrelated senses: e.g. AIDS → aid
  - Others:
    - Large, obvious benefits on morphologically rich langs.
    - Improvements even on English

Overall Approach

- Head-to-head comparison
- AQUAINT documents
  - Enhanced relevance judgments
- Retrieval based on Lucene
  - Boolean retrieval with tf-idf weighting
- Compare retrieval varying stemming and expansion
- Assess results

Example

- Q: What is the name of the volcano that destroyed the ancient city of Pompeii?" A: Vesuvius
- New search query: "Pompeii" and "Vesuvius"
- Relevant: In A.D. 79, long-dormant Mount Vesuvius erupted, burying the Roman cities of Pompeii and Herculaneum in volcanic ash."
- Unsupported: Pompeii was pagan in A.D. 79, when Vesuvius erupted.
- Irrelevant: Vineyards near Pompeii grow in volcanic soil at the foot of Mt. Vesuvius

Stemming & Expansion

- Base query form: Conjunct of disjuncts
  - Disjunction over morphological term expansions
  - Rank terms by IDF
  - Successive relaxation by dropping lowest IDF term
- Contrasting conditions:
  - Baseline: No nothing (except stopword removal)
  - Stemming: Porter stemmer applied to query, index
  - Unweighted inflectional expansion:
    - POS-based variants generated for non-stop query terms
  - Weighted inflectional expansion: prev. + weights

Example

- Q: What lays blue eggs?
- Baseline: blue AND eggs AND lays
- Stemming: blue AND egg AND lai
- UIE: blue AND (eggs OR egg) AND (lays OR laying OR lay OR laid)
- WIE: blue AND (eggs OR egg^{w}) AND (lays OR $laying^{w}$ OR lay^{w} OR $laid^{w}$)

Evaluation Metrics

- Recall-oriented: why?
  - All later processing filters
- Recall @ n:
  - Fraction of relevant docs retrieved at some cutoff
- Total document reciprocal rank (TDRR):
  - Compute reciprocal rank for rel. retrieved documents
  - Sum overall documents
  - Form of weighted recall, based on rank

Results

Limit	Experiment	Recall relevant	Δ	Recall both	Δ	TDRR relevant	Δ	TDRR both	Δ
100	unstemmed	0.2720		0.2595		0.6403		0.6673	
	stemmed	0.2589	−4.82%	0.2460	−5.20%	0.5869	−8.33%	0.5987	−10.28%
	expanded	0.2748	+1.03%	0.2612	+0.66%	0.5752	−10.16%	0.5968	−10.56%
	w. expanded	0.2944	+8.24%	0.2798	+7.82%	0.6094	−4.82%	0.6305	−5.52%
250	unstemmed	0.3738		0.3584		0.6509		0.6790	
	stemmed	0.3626	−3.00%	0.3474	−3.07%	0.5995	−7.90%	0.6122	−9.84%
	expanded	0.3682	−1.50%	0.3533	−1.42%	0.5863	−9.93%	0.6090	−10.31%
	w. expanded	0.3776	+1.02%	0.3618	+0.95%	0.6185	−4.98%	0.6406	−5.67%
500	unstemmed	0.5393		0.5123		0.6596		0.6879	
	stemmed	0.5364	−0.54%	0.5097	−0.51%	0.6086	−7.74%	0.6216	−9.65%
	expanded	0.5467	+1.37%	0.5182	+1.15%	0.5957	−9.69%	0.6186	−10.08%
	w. expanded	0.5551	+2.93%	0.5258	+2.64%	0.6279	−4.81%	0.6501	−5.50%
750	unstemmed	0.5981		0.5689		0.6614		0.6899	
	stemmed	0.5934	−0.79%	0.5638	−0.90%	0.6103	−7.72%	0.6234	−9.63%
	expanded	0.6093	+1.87%	0.5799	+1.93%	0.5976	−9.65%	0.6207	−10.03%
	w. expanded	0.6112	+2.19%	0.5816	+2.23%	0.6296	−4.81%	0.6520	−5.49%
1000	unstemmed	0.6196		0.5917		0.6618		0.6904	
	stemmed	0.6131	−1.05%	0.5824	−1.57%	0.6111	−7.67%	0.6238	−9.64%
	expanded	0.6290	+1.52%	0.5993	+1.28%	0.5980	−9.65%	0.6211	−10.03%
	w. expanded	0.6290	+1.52%	0.5993	+1.28%	0.5980	−9.65%	0.6211	−10.03%

Overall Findings

- Recall:
  - Porter stemming performs WORSE than baseline
    - At all levels
  - Expansion performs BETTER than baseline
    - Tuned weighting improves over uniform
  - Most notable at lower cutoffs
- TDRR:
  - Everything's worse than baseline
  - Irrelevant docs promoted more

Observations

- Why is stemming so bad?
  - Porter stemming linguistically naïve, over-conflates
    - police = policy; organization = organ; European != Europe
  - Expansion better motivated, constrained
- Why does TDRR drop when recall rises?
  - TDRR – and RR in general – very sensitive to swaps at higher ranks
    - Some erroneous docs added higher
- Expansion approach provides flexible weighting

Local Context and SMT for Question Expansion

- "Statistical Machine Translation for Query Expansion in Answer Retrieval", Riezler et al, 2007

- Investigates data-driven approaches to query exp.
  - Local context analysis (pseudo-rel. feedback)
  - Contrasts: Collection global measures
    - Terms identified by statistical machine translation
    - Terms identified by automatic paraphrasing

    - Now, huge paraphrase corpus: wikianswers
      - /corpora/UWCSE/wikianswers-paraphrases-1.0.

Motivation

- Fundamental challenge in QA (and IR)
  - Bridging the "lexical chasm"
    - Divide between user's info need, author's lexical choice
    - Result of linguistic ambiguity
- Many approaches:
  - QA
    - Question reformulation, syntactic rewriting
    - Ontology-based expansion
    - MT-based reranking
  - IR: query expansion with pseudo-relevance feedback

Task & Approach

- Goal:
  - Answer retrieval from FAQ pages
    - IR problem: matching queries to docs of Q-A pairs
    - QA problem: finding answers in restricted document set
- Approach:
  - Bridge lexical gap with statistical machine translation
  - Perform query expansion
    - Expansion terms identified via phrase-based MT

Creating the FAQ Corpus

- Prior FAQ collections limited in scope, quality
  - Web search and scraping 'FAQ' in title/url
  - Search in proprietary collections
  - 1-2.8M Q-A pairs
    - Inspection shows poor quality
- Extracted from 4B page corpus (they're Google)
  - Precision-oriented extraction
    - Search for 'faq', Train FAQ page classifier ➔ ~800K pages
    - Q-A pairs: trained labeler: features?
      - punctuation, HTML tags (<p>,..), markers (Q:), lexical (what,how)
      - ➔ 10M pairs (98% precision)

Machine Translation Model

- SMT query expansion:
  - Builds on alignments from SMT models
- Basic noisy channel machine translation model:
  - e: English; f: French $\arg\max_e p(e \mid f) = \arg\max_e p(f \mid e) p(e)$
  - p(e): 'language model'; p(f|e): translation model
    - Calculated from relative frequencies of phrases
      - Phrases: larger blocks of aligned words
  - Sequence of phrases:

$$p(f_1^I \mid e_1^I) = \prod_{i=1}^{I} p(f_i \mid e_i)$$

Question-Answer Translation

- View Q-A pairs from FAQ as translation pairs
  - Q as translation of A (and vice versa)
- Goal:
  - Learn alignments b/t question words & synonymous answer words
    - Not interested in fluency, ignore that part of MT model
- Issues: Differences from typical MT
  - Length differences ➔ Modify null alignment weights
  - Less important words ➔ Use intersection of bidirectional alignments

Example

- Q: “How to live with cat allergies”
- Add expansion terms
  - Translations not seen in original query

(how, how) (to, to) (live, live) (with, with) (cat, **pet**) (allergies, allergies)
(how, how) (to, to) (live, live) (with, with) (cat, cat) (allergies, **allergy**)
(how, how) (to, to) (live, live) (with, with) (cat, cat) (allergies, **food**)
(how, how) (to, to) (live, live) (with, with) (cat, **cats**) (allergies, allergies)

SMT-based Paraphrasing

- Key approach intuition:
  - Identify paraphrases by translating to and from a 'pivot' language
  - Paraphrase rewrites yield phrasal 'synonyms'
    - E.g. translate E -> C -> E: find E phrases aligned to C
- Given paraphrase pair (trg, syn): pick best pivot
- $$p(syn \mid trg) = \max_{src} p(src \mid trg)\, p(syn \mid src)$$
  $$p(trg \mid syn) = \max_{src} p(src \mid syn)\, p(trg \mid src)$$

SMT-based Paraphrasing

- Features employed:
  - Phrase translation probabilities, lexical translation probabilities, reordering score, # words, # phrases, LM
- Trained on NIST multiple Chinese-English translations
- $$p(syn_1^I \mid trg_1^I) = \left(\prod_{i=1}^{I} p_\phi(syn_i \mid trg_i)^{\lambda_\phi}\right.$$
  $$\times p_{\phi'}(trg_i \mid syn_i)^{\lambda_{\phi'}} \times p_w(syn_i \mid trg_i)^{\lambda_w}$$
  $$\left.\times p_{w'}(trg_i \mid syn_i)^{\lambda_{w'}} \times p_d(syn_i, trg_i)^{\lambda_d}\right)$$
  $$\times l_w(syn_1^I)^{\lambda_l} \times c_\phi(syn_1^I)^{\lambda_c} \times p_{LM}(syn_1^I)^{\lambda_{LM}}$$

Example

- Q: “How to live with cat allergies”
- Expansion approach:
  - Add new terms from n-best paraphrases

(how, how) (to live, to live) (with cat, with cat) (allergies, **allergy**)
(how, **ways**) (to live, to live) (with cat, with cat) (allergies, allergies)
(how, how) (to live with, to live with) (cat, **feline**) (allergies, allergies)
(how to, how to) (live, **living**) (with cat, with cat) (allergies, allergies)
(how to, how to) (live, **life**) (with cat, with cat) (allergies, allergies)
(how, **way**) (to live, to live) (with cat, with cat) (allergies, allergies)
(how, how) (to live, to live) (with cat, with cat) (allergies, **allergens**)
(how, how) (to live, to live) (with cat, with cat) (allergies, **allergen**)

Retrieval Model

- Weighted linear combination of vector similarity vals
  - Computed between query and fields of Q-A pair
- 8 Q-A pair fields:
  - 1) Full FAQ text; 2) Question text; 3) answer text;
  - 4) title text; 5-8) 1-4 without stopwords
  - Highest weights: Raw Q text;
    - Then stopped full text, stopped Q text
    - Then stopped A text, stopped title text
  - No phrase matching or stemming

Query Expansion

- SMT Term selection:
  - New terms from 50-best paraphrases
    - 7.8 terms added
  - New terms from 20-best translations
    - 3.1 terms added
    - Why? - paraphrasing more constrained, less noisy
- Weighting: Paraphrase: same; Trans: higher A text
- Local expansion (Xu and Croft)
  - top 20 docs, terms weighted by tfidf of answers
    - Use answer preference weighting for retrieval
    - 9.25 terms added

Experiments

- Test queries from MetaCrawler query logs
  - 60 well-formed NL questions
- Issue: Systems fail on 1/3 of questions
  - No relevant answers retrieved
    - E.g. “how do you make a cornhusk doll?”, “what does 8x certification mean”, etc
  - Serious recall problem in QA DB
- Retrieve 20 results:
  - Compute evaluation measures @10, 20

Evaluation

- Manually label top 20 answers by 2 judges
- Quality rating: 3 point scale
  - adequate (2): Includes the answer
  - material (1): Some relevant information, no exact ans
  - unsatisfactory (0): No relevant info
- Compute 'Success$_{type}$ @ n'
  - Type: 2,1,0 above
  - n: # of documents returned
- Why not MRR? - Reduce sensitivity to high rank
  - Reward recall improvement
  - MRR rewards systems with answers in top 1, but poorly on everything else

Results

	S_2@10	S_2@20	$S_{1,2}$@10	$S_{1,2}$@20
baseline *tfidf*	27	35	58	65
local expansion	30 (+ 11.1)	40 (+ 14.2)	57 (- 1)	63 (- 3)
SMT-based expansion	38 (+ 40.7)	43 (+ 22.8)	58	65

Example Expansions

	how to live with cat allergies
-	allergens allergic infections filter plasmacluster rhinitis introduction effective replacement
+	allergy cats pet food
+	way allergens life allergy feline ways living allergen
	how to design model rockets
-	models represented orientation drawings analysis element environment different structure
+	models rocket
+	missiles missile rocket grenades arrow designing prototype models ways paradigm
	what is dna hybridization
-	instructions individual blueprint characteristics chromosomes deoxyribonucleic information
	genetic molecule
+	slides clone cdna sitting sequences
+	hibridization hybrids hybridation anything hibridacion hybridising adn hybridisation nothing
	how to enhance competitiveness of indian industries
+	resources production quality processing established investment development facilities institut
+	increase industry
+	promote raise improve increase industry strengthen
	how to induce labour
-	experience induction practice imagination concentration information consciousness different
-	relaxation
-	birth industrial induced induces
	way workers inducing employment ways labor working child work job action unions

Observations

- Expansion improves for rigorous criteria
  - Better for SMT than local RF

- Why?
  - Both can introduce some good terms
  - Local RF introduces more irrelevant terms
  - SMT more constrained
  - Challenge: Balance introducing info vs noise

Machine Learning Approaches

- Diverse approaches:
  - Assume annotated query logs, annotated question sets, matched query/snippet pairs
- Learn question paraphrases (MSRA)
  - Improve QA by setting question sites
  - Improve search by generating alternate question forms